Mars Science Laboratory (MSL)
Software Interface Specification
SAM Experiment Data Record (EDR)

Initial Release
Version 1.4
JPL D-38114
SIS-SCI011-MSL

DATE: December 31, 2013

Prepared by:
Costin Radulescu, OPGS (JPL)

JPL Document Review CL#14-0025

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

SIGNATURES

Prepared by:

Author/Custodian:	Costin Radulescu, JPL	Date

Approval:

PDS Geosciences Node Manager	Raymond Arvidson, Washington University	Date
PDS Program Manager	Thomas H. Morgan, Washington University	Date
OPGS CDE:	Maher Hanna, JPL	Date

Concurrence:

MSL GDS:	Tadas Sesplaukis, JPL, MSL GDS SE	Date
Principal Investigator:	Paul Mahaffy, GSFC	Date

CHANGE LOG

DATE	SECTIONS CHANGED	REASON FOR CHANGE	REVISION
11/28/2010	Appendix A	Introduced the definition of SAM_EXPERIMENT_ID	Ver 1.2
12/13/2010	Section 3	List of EDRs generated.	Ver 1.2
2/25/2011	All	Incorporated changes and feedback from Susan Slavney (PDS Geosciences Node)	Ver 1.3
12/30/2013	All	Document updated to fill in missing information and make corrections throughout.	Ver 1.4

CONTENTS

ACRONYMS AND ABBREVIATIONS

ARA	Areocentric Right Ascension Angle
ASCII	American Standard Code for Information Interchange
CODMAC	Committee on Data Management and Computation
CPSL	Chemical Separation and Processing Laboratory
CS	Coordinate System
DAWG	Data Archive Working Group
EDR	Experiment Data Record
FEI	File Exchange Interface
GC	Gas Chromatograph
GCMS	Gas Chromatograph Mass Spectrometer
GDS	Ground Data System
HGA	High Gain Antenna
IC	Inlet Cover (covers CheMin and SAM solid-sample openings)
JMS	Java Message Server
JPL	Jet Propulsion Laboratory
MIPL	Multi-mission image Processing Laboratory
MPCS	Mission data Processing and Control Subsystem
MSL	Mars Science Laboratory
NAIF	Navigation and Ancillary Information Facility
NASA	National Aeronautics and Space Administration
ODL	Object Description Language
ODS	MSL's Operations Data Storage
OPGS	Operations Products Generation Sub-system
PDS	Planetary Data System
QMS	Quadrupole Mass Spectrometer
RDR	Reduced Data Record
RSM	Remote Sensing Mast
RTO	Real Time Operations (MSL terminology)
SAM	Sample Analysis at Mars
SIS	Software Interface Specification
SMS	Sample Manipulation System
TBD	To Be Determined
TDS	Telemetry Data System
TLS	Tunable Laser Spectrometer

1. INTRODUCTION

1.1 Purpose and Scope

This Software Interface Specification is intended for use by those who wish to understand the format and content of the Mars Science Lander's (MSL) Sample Analysis at Mars (SAM) instrument Experiment Data Record (EDR) archive products kept by the NASA's Planetary Data System (PDS). These SAM EDRs were produced for mission operations and used by the SAM team to create their Reduced Data Records (RDRs), which are covered by a separate SAM RDR SIS provided by the SAM team (JPL D-38123, MSL SIS-SCI020-MSL). Their Level 0 RDRs are the lowest-level products which are likely to be useful to the science community. Because of this, the Project chose to only provide the EDRs as a 'safed' data set, in the form in which they were used in tactical operations, rather than as a fully PDS-compliant archive. The SAM RDR volumes archived in the PDS by the SAM Team constitute a complete archive that should meet all the needs of the science community. The EDRs provide a backup of the raw data which were generated in operations, in case it is ever needed in the future.

1.2 Contents

This SIS describes a very high level description of the SAM instrument. It also describes, at high level, how the SAM data products are acquired by the instrument, and how they are processed, formatted, labeled, and uniquely identified on the ground.

The document discusses standards used in generating the products 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 Exploration Program Data Management Plan, R.E. Arvidson and S. Slavney, Rev. 4, June 15, 2011.
2. Mars Science Laboratory Project Archive Generation, Validation and Transfer Plan, J. Crisp, JPL D-35281, November 13, 2006.
3. Planetary Data System Standards Reference, Version 3.8, JPL D-7669 Part 2, February 27, 2009.
4. Planetary Archive Preparation Guide, Version 1.4, JPL D-31224, http://pds.nasa.gov/tools/archiving.shtml , April 1, 2010.
5. SAM Functional Description Document, Revision A, JPL D-34225, MSL 375-1235, September 24, 2010.

6. SAM Science Team and PDS Geosciences Node ICD, Version 3.0, April 5, 2011.

In addition, this SIS makes reference to the following publication for technical background information about the SAM instrument:

7. Mahaffy, P.M., C.R. Webster, M. Cabane, P.C. Conrad, P. Coll, S.K. Atreya, R. Arvey, M. Barciniak, M. Benna, L. Bleacher, W.B. Brinckerhoff, J.L. Eigenbrode, D. Carignan, M. Cascia, R.A. Chalmers, J.P. Dworkin, T. Errigo, P. Everson, H. Franz, R. Farley, S. Feng, G. Frazier, C. Freissinet, D.P. Glavin, D.N. Harpold, D. Hawk, V. Holmes, C.S. Johnson, A. Jones, P. Jordan, J. Kellogg, J. Lewis, E. Lyness, C.A. Malespin, D.K. Martin, J. Mauren, A.C. McAdam, D. McLennan, T.J. Nolan, M. Noriega, A.A. Pavlov, B. Prats, E. Raaen, O. Sheinman, D. Sheppard, J. Smith, J.C. Stern, F. Tan, M. Trainer D.W. Ming, R.V. Morris, J. Jones, C. Gundersen, A. Steele, J. Wray, O. Botta, L.A. Leshin, T. Owen, S. Battel, B.M. Jakosky, H. Manning, S. Squyres, R. Navarro-González, C.P. McKay, F. Raulin, R. Sternberg, A. Buch, P. Sorensen, R. Kline-Schoder, D. Coscia, C. Szopa, S. Teinturier, C. Baffes, J. Feldman, G. Flesch, S. Forouhar, R. Garcia, D. Keymeulen, S. Woodward, B.P. Block, K. Arnett, R. Miller, C. Edmonson, S. Gorevan, and E. Mumm, The Sample Analysis at Mars investigation and instrument suite, Space Science Reviews, 170:401-478, doi:10.1007/s11214-012-9879-z, 2012.

1.4 Relationships with Other Interfaces

Changes to this SAM SIS document 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
SAM EDRs	P	OPGS/MIPL
MSLEdrGen (telemproc)	S	MIPL
MIPL database schema	P	MIPL
Other SAM Programs/Products/Documents	P/S/D	SAM Science Team

2. DATA PRODUCT CHARACTERISTICS AND ENVIRONMENT

2.1 Instrument Overview

The Sample Analysis at Mars (SAM) Suite Investigation in the MSL Analytical Laboratory is designed to address the present and past habitability of Mars by exploring molecular and elemental chemistry relevant to life and the building blocks for life. SAM addresses carbon chemistry through a search for organic compounds, the chemical state of light elements other than carbon, and isotopic tracers of planetary change.

SAM is a suite of three instruments, a Quadrupole Mass Spectrometer (QMS), Gas Chromatograph (GC), and Tunable Laser Spectrometer (TLS). The QMS and GC can operate together in a GCMS mode for separation (GC) and definitive identification (QMS) of organic compounds. The TLS obtains precise isotope ratios for C and O in carbon dioxide, H and O in water, and also measures trace levels of methane and its carbon isotope ratio.

Three questions about the ability of Mars to support past, present, or future life are addressed by SAM's five science goals as stated in the table below.

Science and Measurement Goal	Habitability Question
1) Survey carbon compound sources and evaluate their possible mechanisms of formation and destruction 2) Search for organic compounds of biotic and prebiotic importance, including methane	What does the inventory or lack of carbon compounds near the surface of Mars tell us about its potential habitability?
3) Reveal the chemical and isotopic state of elements (i.e. N, H, O, S and others) that are important for life as we know it. 4) Determine atmospheric composition including trace species that are evidence of interactions between the atmosphere and soil.	What are the chemical and isotopic states of the lighter elements in the solids and in the atmosphere of Mars and what do they tell us about its potential habitability?
5) Better constrain models of atmospheric and climatic evolution through measurements of noble gas and light element isotopes.	Were past habitability conditions different from today's?

The SAM instrument suite is supported by a sample manipulation system (SMS) and a Chemical Separation and Processing Laboratory (CSPL) that includes high conductance and micro valves, gas manifolds with heaters and temperature monitors, chemical and mechanical pumps, carrier gas reservoirs and regulators, pressure monitors, pyrolysis ovens, and chemical scrubbers and getters. The Mars atmosphere is sampled by CSPL valve and pump manipulations that introduce an appropriate amount of gas through an inlet tube to the SAM instruments. The solid phase materials are sampled by transporting finely sieved materials to one of 74 SMS sample cups that can then be inserted into a SAM oven and thermally processed for release of volatiles.

The SAM mechanical configuration and a top-level schematic of its sample flow configuration are illustrated below. A more detailed description of the instrument can be found in Mahaffy et al., 2012 [7].

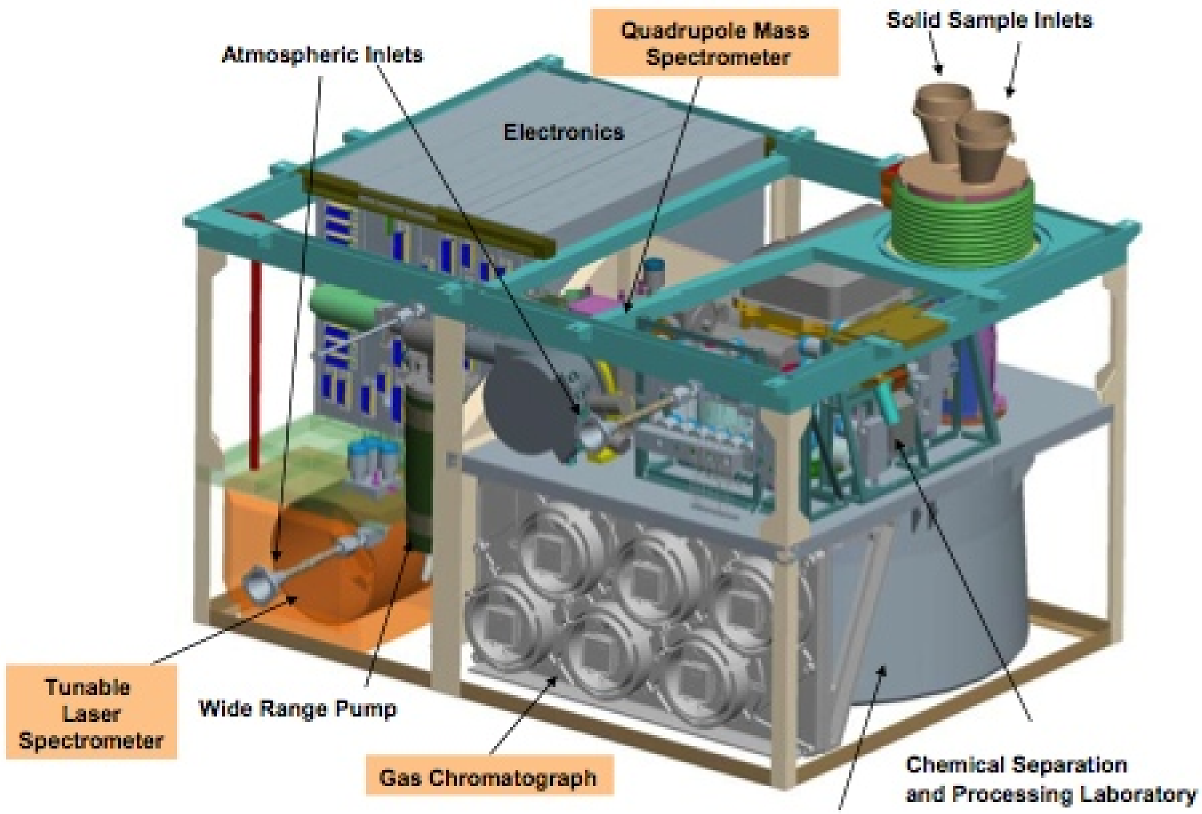

Figure 1: The illustration of the mechanical configuration of SAM shows the three instruments and several elements of the Chemical Separation and Processing Laboratory.

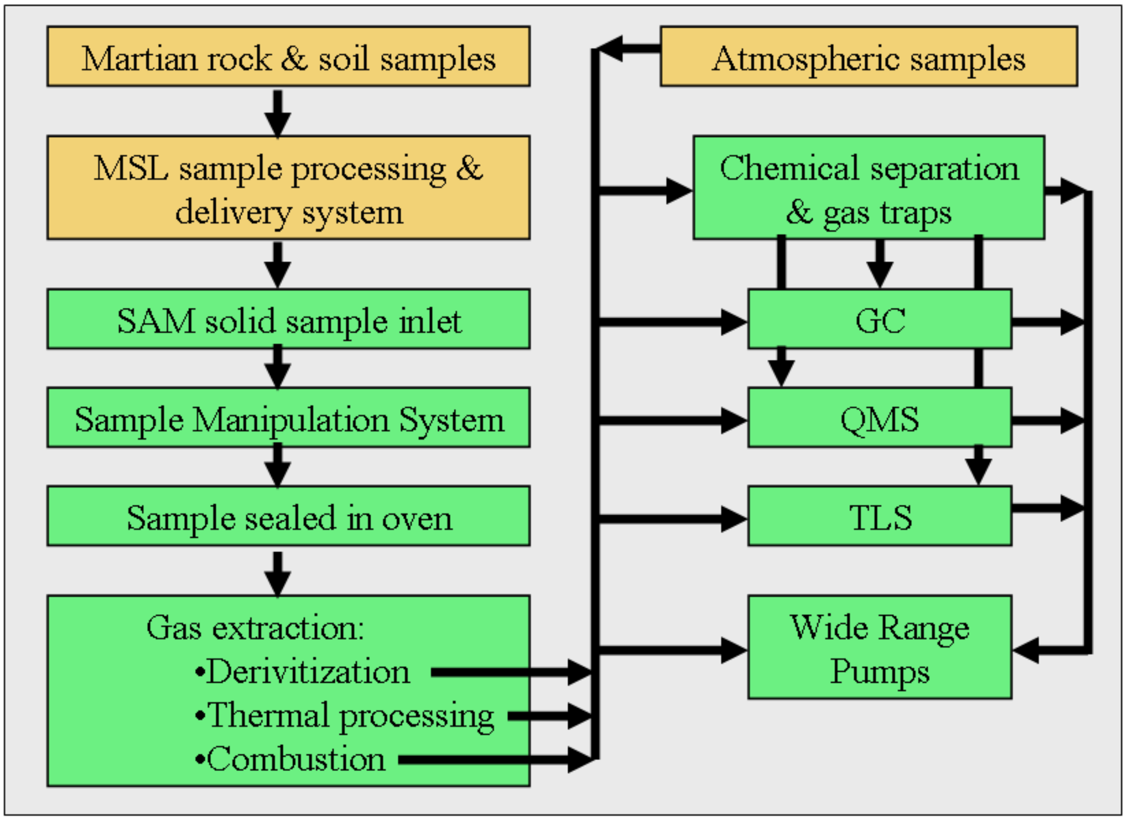

Figure 2: The path of solid and gas samples delivered by MSL subsystems to the SAM instruments is shown. Arrows designate the direction of gas and solid transport.

2.2 Data Product Overview

SAM's NASA Level 0 EDRs are generated from the reconstructed data products that are produced by the Mission data Processing and Control (MPCS) subsystem, which is a component of MSL Real Time Operations (RTO), which itself is part of the MSL Ground Data System (GDS).

SAM EDRs consist of clusters of uncalibrated data records, in time-order, which are grouped together. Each EDR has a unique PDS detached label file, which provides all the meta and ancillary data for that EDR. The pair of files, namely the SAM binary data file and its PDS label, are 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".

2.3 Data Processing

2.3.1 Data Processing Level

SAM EDRs use the "Committee on Data Management and Computation" (CODMAC) data level numbering system. See the table below for more details. The scope of this SIS is limited to data products in CODMAC Level 2 (equivalent to 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).
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 SAM EDR data products are generated by the Multi-mission Image Processing Laboratory (MIPL) at JPL, under the Operations Products Generation Subsystem (OPGS) using the telemetry processing software called MSLEdrGen. The EDR data products are raw uncalibrated data reconstructed from telemetry data products and formatted according to this EDR SIS. With the exception of NAIF-provided ancillary data, all meta data stored in the label are acquired from the telemetry data stream.

The SAM EDR data products generated by OPGS during operations are created collectively from:

a) MPCS data products

b) SPICE kernels (from NAIF)

c) a meta-data database.

The EDR processing begins with the reconstruction of packetized telemetry data resident on the Telemetry Data System (TDS) by the Mission data Processing and Control Subsystem (MPCS) into a binary ".dat" data product and associated ".emd" Earth meta-data file. The data product and meta-data are written by MPCS to the Operations Data Store (ODS) and messages are generated on a Java Message Server (JMS) bus, where they are ingested by MIPL's EDR generator "MSLEdrGen" and processed with SPICE kernels provided by NAIF. The EDR is generated within 60 seconds after the JMS message describing the ODS location of the respective the binary data product and associated Earth meta-data file has been received by the OPGS pipeline system.

In all EDR cases, missing packets will be identified and reported for retransmission to the ground as "partial datasets." Prior to retransmission, the missing EDR data are filled with zeros. The EDR data are reprocessed only after all "partial datasets" are retransmitted and received on the ground. In these cases, the EDR version is incremented to avoid overwriting any previous EDR versions. The EDR data product is then placed into FEI for distribution and to facilitate the SAM team's archiving process.

The size of a SAM EDR data file varies depending on the type of SAM experiment that was run.

2.3.3 Data Flow

The EDRs are stored and staged on the MSL Project's ODS (Operations Data Storage) at JPL, and are then published into MIPL's File Exchange Interface (FEI) in real-time for secured electronic delivery and distribution to all subscribed users/sites for operations. The PDS also uses FEI to collect the SAM EDR data products for safekeeping.

The SAM team converts the EDRs to a more easy-to-use format before delivery to the PDS for archiving as the lowest-level RDR product, at NASA level 0. Higher-level RDR products generated by the SAM team represent the results of further data manipulation and interpretation.

2.3.4 Labeling and Identification

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

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

<INSTR><CONFIG><SCLK><PROD><SOL><SITE><DRIVE><EXPID><VENUE/WHO><VER>.<EXT>

where,

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

Valid values for Instrument ID's are:

Valid values for:

"**SM**" - SAM

Config/ spec = (2 alphanumeric) Instrument Configuration, an operational attribute of the Instrument that assists in characterizing the data.

Not used for SAM EDRs. Filled in with "__"

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 were acquired. For SAM EDRs, the SCLK provided in the emd meta data file is 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 <u>no</u> processing applied,
The next two characters identify the type of EDR product.

Valid values for Product identifiers are listed below for EDRs:

Product Type Description	Value
TelemetryHiPri	"**ETH**"
Telemetry	"**ETE**"
Directory	"**EDI**"

Housekeeping	“**EHK**”
MessageLog	“**EML**”
MemoryDump	“**EMD**”
Parms	“**EPA**”

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

site = (3 alphanumeric) Site location count, from the Rover Motion Counter IRMC).

This field has the following rules-of-thumb:

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

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

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

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

This field has the following rules-of-thumb:

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

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

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

Range 41200 thru 41299 - “**CA00**”, “**CA01**”, **…** “**CA99**”

•

•

•

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

expID = (5 alphanumeric) Experiment ID. The last two characters, if not used, are filled in with “__”

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

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

This field has the following rules-of-thumb:

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

b) Producer - If value is “**P**” (for Flight) or “**Y**” (for Engineering), the provider of the product is the Principal Investigator. Except for MIPL as the provider (“**M**” for Flight or “**Z**” for Engineering).

See the following table of valid values:

Venue		**by Producer**
Flight Model	Eng. Model	
“**M**”	“**Z**”	**M**IPL (OPGS at JPL)
“**P**”	“**Y**”	**P**rincipal Investigator of Instrument … Instrument — Principal Investigator Affiliation SAM — GSFC (Goddard, MD)
“**A**” - “**L**”	“**Q**” - “**X**”	Co-Investigators (to be identified by P.I. per instrument). Not used for SAM.

ver = (1 alphanumeric) Version identifier.

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

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

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

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

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

Valid values for nominal operations non-camera data products:

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

Example #1: SM__413235771ETH0177006000025058__M2.DAT

SM__413235771ETH0177006000025058__M2.LBL

where,

instr	=	"**SM**"	=	APXS
config	=	"__"	=	None
sclk	=	"**413235771**"	=	Spacecraft Clock Start Count of 413235771 secs
prod	=	"**ETH**"	=	"E"-EDR, "TH"-TelemetryHiPri
sol	=	"**0177**"	=	Sol 177
site	=	"**006**"	=	Site 6
drive	=	"**0000**"	=	Drive (position within site) 0
expID	=	"**25058**"	=	SAM Experiment ID
__	=	"__"	=	None
venue/who	=	"**M**"	=	Flight Model data / produced at JPL)
ver	=	"**1**"	=	Version 1
ext		"**DAT**"		Data product with PDS label

2.4 Standards Used in Generating Data Products

2.4.1 PDS Standards

The SAM EDRs comply with Planetary Data System standards for file formats and labels, as specified in the PDS Standards Reference [3].

2.4.2 Time Standards

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

Time Format	*Definition*
SCET	Spacecraft event time. This is the time when an event occurred on-board the spacecraft, in UTC. It is usually derived from SCLK.
SCLK	Spacecraft Clock. This is an on-board 64-bit counter, in units of nano-seconds and increments once every 100 milliseconds. Time zero corresponds to midnight on 1-Jan-1980.
ERT	Earth Received Time. This is the time when the first bit of the packet containing the current data was received at the Deep Space (DSN) station. Recorded in 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 therefore is SOL zero.

2.4.3 Coordinate Frame Standards

The following coordinate frame systems are used within the project to refer to the position of the rover and its instruments.

Frame Name (Label Keyword Value)	*SHORT NAME (SAPP FDD)*	*REFERENCE FRAME (USED TO DEFINE)*	*Coordinate Frame*	
			Origin	*Orientation*
ROVER_NAV_FRAME	RNAV	Enclosing SITE_FRAME	Attached to rover	Aligned with rover
ROVER_MECH_FRAME	RMECH	Enclosing SITE_FRAME	Attached to rover	Aligned with rover
LOCAL_LEVEL_FRAME	LL	Enclosing SITE_FRAME	Attached to rover (coincident with Rover Nav Frame)	North/East/Nadir
SITE_FRAME	SITE(n)	Previous SITE_FRAME	Attached to surface	North/East/Nadir
RSM_HEAD_FRAME	RSM_HEAD	ROVER_NAV_ FRAME	Attached to mast head	Aligned with pointing of mast head. This corresponds to RSM_HEAD in the Frame Manager

Frame Name (Label Keyword Value)	SHORT NAME (SAPP FDD)	REFERENCE FRAME (USED TO DEFINE)	Coordinate Frame	
			Origin	Orientation
Arm Frames: ARM_TURRET_FRAME ARM_DRILL_FRAME ARM_DRT_FRAME ARM_MAHLI_FRAME ARM_APXS_FRAME ARM_PORTION_FRAME ARM_SCOOP_TIP_FRAME ARM_SCOOP_TCP_FRAME	Arm Frames: TURRET DRILL DRT MAHLI APXS PORTION SCOOP_TIP SCOOP_TCP	ROVER_NAV_FRAME	Attached to the tool; see PPPCS for the specific tool frame.	Aligned with tool in some way; see PPPCS [Ref 1] for the specific tool Frame.

2.4.4 Data Storage Conventions

SAM EDR products are stored as binary files. For every SAM EDR, there is a corresponding detached PDS label file. The detached PDS labels for SAM EDRs are stored as ASCII text, with each keyword definition terminated by ASCII carriage-return and line-feed characters. The EDR products are described and defined as PDS table objects. The PDS label keywords are described in Appendix-A.

All SAM EDRs contain fixed-length records, although the size of the records in each file may differ depending on the context of the data. Label keywords provide necessary information to determine the size and organization of the binary records.

2.5 Data Validation

There is no plan to subject the SAM EDRs to PDS peer review and verification, because the SAM Level 0 RDRs should meet the needs of the science community as the lowest level of data necessary for scientific investigations.

3. DETAILED DATA PRODUCT SPECIFICATIONS

The SAM instrument generates different data packets depending on the state of the instrument. The instrument sends its data out, in the form of packets to the RCE, which in turn builds them into discreet data products.

To facilitate ground processing, each of the data products is saved as a different EDR. Each are defined in the following sections. See Section 2.3.4 for a detailed description of the SAM EDR file-naming convention, which uniquely identifies each data product.

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

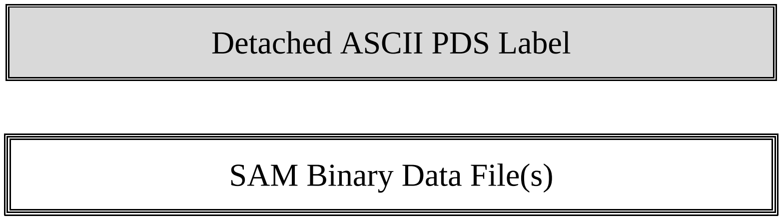

Figure 3.1: The SAM EDR consists of two files.

OPGS creates the following SAM EDR types:

SAM Data Product (DP)	EDR
SamTelemetryHiPri	Telemetry High Priority EDR (“**ETH**”) – Science
SamTelemetry	Telemetry EDR (“**ETE**”) - Science
SamDirectory	Directory EDR (“**EDI**”)
SamHousekeeping	Housekeeping EDR (“**EHK**”)
SamMessageLog	Message Log EDR (“**EML**”)
SamMemoryDump	Memory Dump EDR (“**EMD**”)
SamParms	Parms EDR (“**EPA**”)

3.1 Data Format Descriptions

Appendix – B includes the table for binary data columns definition to the byte level.

3.2 Label and Header Descriptions

3.2.1 PDS Label

SAM EDR products have detached PDS labels stored as ASCII. 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. Detached labels are paired with the science data, having the same file name, but with file extension of “*.LBL”.

PDS labels are written in Object Description Language (ODL) [3]. PDS label statements have the form of "keyword = value". Each label statement is terminated with a carriage return character (ASCII 13) and a line feed character (ASCII 10) sequence to allow the label to be read by many operating systems. Pointer statements with the following format are used to indicate the location of data objects in the file:

^Object = location

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

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

Per PDS rules, “N/A” is used when a keyword exists, but it does not apply to a particular data product, 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

This does not apply to SAM 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. EXAMPLE OF A SAM PDS LABEL

```
PDS_VERSION_ID                  = PDS3

/* FILE DATA ELEMENTS */

RECORD_TYPE                     = FIXED_LENGTH
RECORD_BYTES                    = 2
FILE_RECORDS                    = 1

/* POINTERS TO DATA OBJECTS */

^SCIENCE_TABLE                  =
("SM__413235771ETH0177006000025058__M1.DAT",1)

/* IDENTIFICATION DATA ELEMENTS */

DATA_SET_ID                     = "MSL-M-SAM-2-EDR-V1.0"
DATA_SET_NAME                   = "MSL MARS SAMPLE ANALYSIS
                                  AT MARS 2 EDR V1.0"
COMMAND_SEQUENCE_NUMBER         = 5
INSTRUMENT_HOST_ID              = MSL
INSTRUMENT_HOST_NAME            = "MARS SCIENCE LABORATORY"
INSTRUMENT_ID                   = SAM
```

```
INSTRUMENT_NAME                   = "SAMPLE ANALYSIS AT MARS"
INSTRUMENT_TYPE                   = SPECTROMETER
MSL:SAM_EXPERIMENT_ID             = 25058
MSL:LOCAL_MEAN_SOLAR_TIME         = "Sol-00177M20:36:30:091"
LOCAL_TRUE_SOLAR_TIME             = "20:36:39"
MISSION_NAME                      = "MARS SCIENCE LABORATORY"
MISSION_PHASE_NAME                = "PRIMARY SURFACE MISSION"
OBSERVATION_ID                    = UNK
PLANET_DAY_NUMBER                 = 177
PRODUCER_INSTITUTION_NAME         = "MULTIMISSION IMAGE PROCESSING LABORATORY,
                                    JET PROPULSION LAB"
PRODUCT_CREATION_TIME             = 2013-02-05T15:50:57.000
PRODUCT_ID                        = "SM__413235771ETH0177006000025058__M1"
PRODUCT_VERSION_ID                = "V1.0 D-38114"
PRODUCT_TYPE                      = SAM_EDR
RELEASE_ID                        = "0001"
MSL:REQUEST_ID                    = "0"
ROVER_MOTION_COUNTER              = (6,0,2,824,38,72,1910,1754,0,0)
ROVER_MOTION_COUNTER_NAME         = (SITE, DRIVE, POSE, ARM, CHIMRA,
                                    DRILL, RSM, HGA, DRT, IC)
SEQUENCE_ID                       = "sam_01177"
SEQUENCE_VERSION_ID               = "1"
MSL:ACTIVE_FLIGHT_STRING_ID       = A
SOLAR_LONGITUDE                   = -102.021
SPACECRAFT_CLOCK_CNT_PARTITION    = 1
SPACECRAFT_CLOCK_START_COUNT      = "0413235771.007"
SPACECRAFT_CLOCK_STOP_COUNT       = "UNK"
START_TIME                        = 2013-02-04T07:47:27.706
STOP_TIME                         = "UNK"
TARGET_NAME                       = MARS
TARGET_TYPE                       = PLANET

/* TELEMETRY DATA ELEMENTS */

APPLICATION_PROCESS_ID            = 533
APPLICATION_PROCESS_NAME          = "SamTelemetryHipri"
MSL:VIRTUAL_CHANNEL_ID            = "0"
MSL:COMMUNICATION_SESSION_ID      = "31791"
DOWNLOAD_PRIORITY                 = 95
EARTH_RECEIVED_START_TIME         = 2013-02-05T16:11:12.379
MSL:EXPECTED_TRANSMISSION_PATH    = "3851"
MSL:AUTO_DELETE_FLAG              = "0"
MSL:TRANSMISSION_PATH             = "65535"
FLIGHT_SOFTWARE_VERSION_ID        = "141801503"
MSL:FLIGHT_SOFTWARE_MODE          = "8"
MSL:PRODUCT_COMPLETION_STATUS     = COMPLETE_CHECKSUM_PASS
MSL:PRODUCT_TAG                   = "0"
MSL:SEQUENCE_EXECUTION_COUNT      = 1
RECEIVED_PACKETS                  = 1
SPICE_FILE_NAME                   = "chronos.msl"
TELEMETRY_PROVIDER_ID             = MPCS_MSL_DP
MSL:TELEMETRY_SOURCE_CHECKSUM     = 323
MSL:TELEMETRY_SOURCE_SCLK_START   = "413235771.7559"
MSL:TELEMETRY_SOURCE_SIZE         = 2
MSL:TELEMETRY_SOURCE_START_TIME   = 2013-035T07:47:27.821
TELEMETRY_SOURCE_NAME             = "SamTelemetryHipri_0413235771-07559-1.dat"
```

```
MSL:TELEMETRY_SOURCE_TYPE         = "DATA PRODUCT"
MSL:TELEMETRY_SOURCE_HOST_NAME    = mslsmsampcs1
EXPECTED_PACKETS                  = 1

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

GROUP                             = ROVER_COORDINATE_SYSTEM_PARMS
 COORDINATE_SYSTEM_NAME           = ROVER_FRAME
 COORDINATE_SYSTEM_INDEX          = (6,0,2,824,38,72,1910,1754,0,0)
 COORDINATE_SYSTEM_INDEX_NAME     = (SITE, DRIVE, POSE, ARM, CHIMRA,
                                    DRILL, RSM, HGA, DRT, IC)
 ORIGIN_OFFSET_VECTOR             = (0,0,0)
 ORIGIN_ROTATION_QUATERNION       = (-0.0307528,-0.0331578,0.991347,0.123228)
 POSITIVE_AZIMUTH_DIRECTION       = CLOCKWISE
 POSITIVE_ELEVATION_DIRECTION     = UP
 REFERENCE_COORD_SYSTEM_NAME      = SITE_FRAME
 REFERENCE_COORD_SYSTEM_INDEX     = 6
END_GROUP                         = ROVER_COORDINATE_SYSTEM_PARMS

/* ARTICULATION DEVICE STATE: REMOTE SENSING MAST */

GROUP                             = RSM_ARTICULATION_STATE_PARMS
 ARTICULATION_DEVICE_ID           = RSM
 ARTICULATION_DEVICE_NAME         = "REMOTE SENSING MAST"
 ARTICULATION_DEVICE_ANGLE        = (3.15916<rad>, 0.750437<rad>)
 ARTICULATION_DEVICE_ANGLE_NAME   = (AZIMUTH, ELEVATION)
END_GROUP                         = RSM_ARTICULATION_STATE_PARMS

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

GROUP                             = ARM_ARTICULATION_STATE_PARMS
 ARTICULATION_DEVICE_ID           = ARM
 ARTICULATION_DEVICE_NAME         = ROBOTIC_ARM
 ARTICULATION_DEVICE_ANGLE        = (-1.49019<rad>, -1.69638<rad>,
                                     -1.77323<rad>, 1.93044<rad>,
5.46116<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_PARMS


/* DATA OBJECT DEFINITION DESCRIPTIONS */

/* SAM SCIENCE OBJECT */

OBJECT                            = SCIENCE_TABLE
   INTERCHANGE_FORMAT             = BINARY
   COLUMNS                        = 1
   ROWS                           = 1
   ROW_BYTES                      = 2

   OBJECT                         = COLUMN
      NAME                        = "SamExperimentId"
```

```
      MSL:PRODUCT_COMPLETION_STATUS = "COMPLETE_CHECKSUM_PASS"
      START_BYTE                    = 1
      BYTES                         = 2
      DATA_TYPE                     = MSB_UNSIGNED_INTEGER
      DESCRIPTION                   = "MSL SAM binary data."
      CHECKSUM                      = 323
   END_OBJECT                       = COLUMN

END_OBJECT                          = SCIENCE_TABLE

END
```

Appendix – A

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		"PDS<*version*>"	PDS
/* FILE DATA ELEMENTS */	Comment				
RECORD_TYPE	Specifies the record format of a file. Note: In the PDS, when RECORD_TYPE is used in a detached label file, it always describes its corresponding detached data file 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	Calculation
RECORD_BYTES	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		"0" to n	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		“0” to n	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)		"MSL-M-SAM-2-EDR-V1.0" (or some other version number)	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		“"MSL MARS SAMPLE ANALYSIS AT MARS 2 EDR V1.0"	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.	Integer		n/a	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		"SAM"	EMD: ProductName
INSTRUMENT_NAME	Name of the instrument, free format, enclosed in double quotes.	String		"SAMPLE ANALYSIS AT MARS"	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. One of two types of solar time used to express the time of day at a point on the surface of a planetary body. LTST is measured relative to the true position of the Sun as seen from a point on the planet's surface. The coordinate system used to define LTST has its origin at the center of the planet. Its Z-axis is the north pole vector (or spin axis) of the planet. The X-axis is chosen to point in the direction of the vernal equinox of the planet's orbit. (The vernal or autumnal equinox vectors are found by searching the planetary ephemeris for those times when the vector from the planet's center to the Sun is perpendicular to the planet's north pole vector. The vernal equinox is the time when the Sun appears to rise above the planet's equator.) Positions of points in this frame can be expressed as a radius and areocentric 'right ascension' and 'declination' angles. The areocentric right ascension angle, or ARA, is measured positive eastward in the equatorial plane from the vernal equinox vector to the intersection of the meridian containing the point with the equator. Similarly, the areocentric declination is the angle between the equatorial plane and the vector to the point. LTST is a function of the difference between the ARAs of the vectors to the Sun and to the point on the planet's surface. Specifically, $LTST = (a(P) - a(TS)) * (24 / 360) + 12$ where, LTST = the local true solar time in true solar hours, a(P) = ARA of the point on the planet's surface in deg, and a(TS) = ARA of the true sun in degrees. The conversion factor of 24/360 is applied to transform the angular measure in decimal degrees into hours-minutes-seconds of arc. This standard representation divides 360 degrees into 24 hours, each hour into 60 minutes, and each minute into 60 seconds of arc. The hours, minutes, and seconds of arc are called 'true solar' hours, minutes, and seconds when used to measure LTST. The constant offset of 12 hours	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
	is added to the difference in ARAs to place local noon (12:00:00 in hours, minutes, seconds) at the point where the Sun is directly overhead; at this time, the ARA of the true sun is the same as that of the surface point so that a(P) - a(TS) = 0. The use of 'true solar' time units can be extended to define a true solar day as 24 true solar hours. Due to the eccentricity of planetary orbits and the inclination of orbital planes to equatorial planes (obliquity), the Sun does not move at a uniform rate over the course of a planetary year. Consequently, the number of SI seconds in a true solar day, hour, minute or second is not constant. For MSL, the valid value is expressed in terms of a 24-hour clock, so the acceptable range is “00:00:00.000” to “23:59:59.999”. See also LOCAL_MEAN_SOLAR_TIME.				
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 negative.	Integer			Calculation - SCLK kernel

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
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”, “ENTERY, 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			DPO
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

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
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
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		n/a	User parameter input.

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
REQUEST_ID	Specifies the Request ID value associated with the Data Product generation command. Unsigned integer.	String		n/a	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 ten values. In order, they are “Site”, “Drive”, “Pose”, “Arm”, “CHIMRA”, “Drill”, “RSM”, “HGA”, “DRT”, and “IC”. 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, along with Pose, resets all later indices to 0 when they increment. The Arm, CHIMRA, DRILL, RSM, HGA, DRT and IC increment whenever the corresponding articulation device moves. These all increment independently of each other; they are reset to zero only when the SITE or DRIVE or POSE changes.	Integer array [10]		n/a	EMD: RoverMotionCounter (SiteIndex, DriveIndex, PoseIndex, ArmIndex, Chimralndex, DrillIndex, RsmIndex, HgaIndex, DrtIndex, IcIndex)

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
ROVER_MOTION_COUNTER_NAME	Specifies an array that provides the formal names identifying each integer in ROVER_MOTION_COUNTER.	String		"SITE", “DRIVE”, “POSE”, “ARM”, “CHIMRA”, “DRILL”, “RSM”, “HGA”, “DRT”, “IC”	Constant
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.	String		n/a	EMD: SequenceId
SEQUENCE_VERSION_ID	Specifies the version identifier for a particular observation sequence used during planning or data processing.	String		n/a	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. Specifies the value of the spacecraft clock at the beginning of a time period of interest. Format is “ssssssssss.mmm”, stored as a floating point number where, “ssssssssss” = seconds converted from clock’s coarse counter “mmm” = milliseconds converted from clock’s fine counter. For MSL, the time period of interest is the beginning of data acquisition. The fractional component “mmm” is computed as follows: [(shift right 12 bits) / 2**20] * 1000	string(30)		Format is ssssssssss.mmm, 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

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
TARGET_NAME	Specifies a target. The target may be a planet, satellite, ring, region, feature, asteroid or comet. See TARGET_TYPE.	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
APPLICATION_PROCESS_NAME	Provides the name associated with the source/process which created the data.	String(256)		n/a	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).	String		NULL	EMD: VcId
COMMUNICATION_SESSION_ID	Active Communication Session ID at time of MPDU creation.	String			EMD: CommSessionId

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
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.	Integer		“0” to “101”	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	String		n/a	EMD: TransmissionControlCriterion
SAM_EXPERIMENT_ID	The SAM_EXPERIMENT_ID is a number that uniquely identifies the experiment that produced the data product.	U16		00000 to 99999	"SamExperimentId" DPOs: - ID field The SamExperimentId DPO is the first DPO encapsulated in all the SAM data products (*.dat files).
AUTO_DELETE_FLAG	Indicates if the DP will be deleted upon transmission: 1 for delete, 0 for no delete.	String		0 = “FALSE” 1 = “TRUE”	EMD: DeleteOnSend
TRANSMISSION_PATH	Routing status at time of MPDU generation. Indicates the actual transmission paths (routes) of the Data Product.	String		n/a	EMD: TransmissionStatus
FLIGHT_SOFTWARE_VERSION_ID	Identifies the flight software version	String		n/a	EMD: FswVersion

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
FLIGHT_SOFTWARE_MODE	Active Flight Software version at Data Product creation	Unsigned integer		**Valid Values** Keyword Value FSW Field Value “0” UNKNOWN “1” TEST “2” PRELAUNCH “3” LAUNCH “4” ECLIPSE “5” CRUISE “6” EDL_APPROAH “7” EDL_MAIN “8” SURFACE_NOMINAL “9” SURFACE_STANDBY “10” NONPRIME_TEST “11” NONPRIME_PRELAUNCH “12” NONPRIME_LAUNCH “13” NONPRIME_ECLIPSE “14” NONPRIME_CRUISE “15” NONPRIME_EDL_APPROACH “16” NONPRIME_EDL_MAIN “17” NONPRIME_SURFACE_NOMINAL “18” NONPRIME_SURFACE_STANDBY	FswMode
PRODUCT_COMPLETION_STATUS	Identifies the completion level of the product	String		“COMPLETE” “PARTIAL”	EMD: GroundStatus

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
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.	Integer		n/a	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	String		"A", "B"	EMD: CreationStringId
TELEMETRY_PROVIDER_ID	Specifies the provider and version of the telemetry data used in the generation of the data.	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.	String		<sssssssss>.<mmm>	EMD: DvtCoarse/DvtFine
TELEMETRY_SOURCE_START_TIME	A SCET calculated by MPCS using the Data Validity time and the SCLK/SCET correlation file.	Time		<YYYY>-<DDD>T<hh>:<mm>:<ss>[.<fff>]	EMD: Scet
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.	Integer		n/a	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".	Integer		0 to n	EMD: PartList:TotalReceived

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
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		n/a	User parameter input
TELEMETRY_SOURCE_NAME	Identifies the telemetry originator	String		n/a	Data File Name
TELEMETRY_SOURCE_TYPE		String		“DATA PRODUCT”	Constant
TELEMETRY_SOURCE_HOST_NAME	Identifies the VENUE HOST name	String		n/a	EMD: Venue:Host
EXPECTED_PACKETS	Specifies the total number of telemetry packets which constitute a complete data product, i.e., a data product without missing data. NOTE: For MSL, telemetry data processing does not track “packets”, but instead data product “parts”.	Integer		n/a	EMD: TotalParts
/* COORDINATE SYSTEM STATE: ROVER */	Comment				

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
COORDINATE_SYSTEM_NAME	Specifies the full name of the coordinate system to which the state vectors are referenced. When in a COORDINATE_SYSTEM group, this keyword provides the full name of the coordinate system being defined by the group. The rest of the keywords in the group describe how this coordinate system is related to some other (the "reference"). Non-unique coordinate systems (such as "SITE" for rover or lander missions), which have multiple instances using the same name, also require COORDINATE_SYSTEM_INDEX to completely identify the coordinate system. NOTE: A CS is named by three things: 1) the CS name (e.g. site, rover), 2) the set of indices, and 3) the solution ID (see SOLUTION_ID). A set of index-names is sometimes included for documentation purposes only. The solution ID is often omitted to indicate the default, but is logically part of the name. Any time there's a location (XYZ) or an orientation (in quaternion or angle form), the label group containing that location/orientation should contain the name of the CS in which it is expressed. This means it should contain the three "REFERENCE_*" keywords (or two if the solution ID is omitted). This includes CS definition groups, which have to define in what frame the origin/rotation numbers are expressed (i.e. what is the reference frame).	String		ROVER_FRAME	Constant

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
COORDINATE_SYSTEM_INDEX	Specifies an integer array used to record and track the movement of a rover or lander during surface operations. When in a COORDINATE_SYSTEM_STATE group, this keyword identifies which instance of the coordinate frame, named by COORDINATE_SYSTEM_NAME, is being defined by the group. For MSL, these indices are based on the ROVER_MOTION_COUNTER and are in the same order as specified by ROVER_MOTION_COUNTER_NAME. • EDRs will always contain all 10 elements for this keyword. • For RDRs, the number of indices can be anything from 1 (used for SITE_FRAME) up to 10; however only 1, 2, 3, and 10 indices are common in RDRs.	String array[10]		Example: (6,0,2,824,38,72,1910,1754,0,0)	EMD: RoverMotionCounter (SiteIndex, DriveIndex, PoseIndex, ArmIndex, Chimralndex, DrillIndex, RsmIndex, HgaIndex)
COORDINATE_SYSTEM_INDEX_NAME	Specifies an array that provides the formal names identifying each integer in COORDINATE_SYSTEM_INDEX.	String array[10]		(SITE, DRIVE, POSE, ARM, CHIMRA, DRILL, RSM, HGA, DRT, IC)	RoverMotionCounter (SiteIndex, DriveIndex, PoseIndex, ArmIndex, Chimralndex, DrillIndex, RsmIndex, HgaIndex, DrtIndex, IcIndex)

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. For MSL, here is an example: In the case of the RSM_COORDINATE_SYSTEM group, ORIGIN_OFFSET_VECTOR describes the rotation of the RSM (camera head) boresight (about the ORIGIN_OFFSET_VECTOR) relative to the Rover frame.	Float Array[3]		(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 rotation occurs. Note that quaternions have different component order conventions between flight and ground software. They are received in the order “(v1, v2, v3, s)”. However, the ground order convention is “(s, v1, v2, v3)”, and all values are converted to the ground order before being stored in the label. 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, Chimralndex, 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, Chimralndex, DrillIndex, RsmIndex, HgaIndex)

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
/* ARTICULATION_DEVICE_STATE: REMOTE SENSING MAST */	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(30)		RSM_ARTICULATION_STATE	static
ARTICULATION_DEVICE_NAME	Specifies the formal name which identifies each of the values used in ARTICULATION_DEVICE_ANGLE	string		REMOTE_SENSING_MAST	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. For the RSM, the "MEASURED" values (see ARTICULATION_DEVICE_ANGLE _NAME) represent the value of the resolver (attached to the output side of the joint), while the "FINAL" values represent the encoder (attached to the motor). The resolver should be preferentially used if available, as it measures the angle after joint backlash. A value of 1e+30 indicates the angle is not available, in which case, the encoder should be used instead. Note that the "INITIAL" and "REQUESTED" values are also encoder measurements, and could be used to determine the joint's direction of motion for backlash determination.	float array[2]		(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
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)

Keyword Name	Definition	Type	Units	Valid Values	Location & Source
ARTICULATION_DEVICE_ANGLE_NAME	Specifies the formal name which identifies each of the values used in ARTICULATION_DEVICE_ANGLE.	Integer		**Valid values for the Robotic Arm:** ("JOINT 1 AZIMUTH-ENCODER", "JOINT 2 ELEVATION-ENCODER", "JOINT 3 ELBOWENCODER", "JOINT 4 WRIST-ENCODER", "JOINT 5 TURRET-ENCODER", "JOINT 1 AZIMUTHRESOLVER", "JOINT 2 ELEVATIONRESOLVER", "JOINT 3 ELBOWRESOLVER", "JOINT 4 WRISTRESOLVER", "JOINT 5 TURRETRESOLVER")	EMD: SampleArmOrientation (JointAngle1, JointAngle2, JointAngle3, JointAngle4, JointAngle5)