

# Mars Exploration Rover (MER)

## Software Interface Specification

Interface Title: **Mini-TES Experiment Data Record (EDR), Reduced Data Record (RDR) and Higher-level Data Products**

Mission: MER Date: February 1, 2007

Module ID: SIS-SCI007-MER

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

Reference Module ID: N/A

Date: N/A

### Signatures

#### ***GDS Generating Elements:***

Ops Product Generation Subsystem (OPGS)  
Justin Maki

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

#### ***Concurrence:***

Mini-TES Payload Element Lead  
Phil Christensen

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

MER Mini-TES Investigation Scientist  
Diana Blaney

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

MER Science Manager  
John Callas

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

PDS Central Node Data Engineer  
Betty Sword

\_\_\_\_\_  
Data Engineer Date

MER MIPL Cognizant Engineer  
Douglas Alexander

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

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

Mini-TES Experiment Data Record (EDR), Reduced  
Data Record (RDR) and Higher-level Data Products  
Software Interface Specification (SIS)

## **Version 2.2**

Prepared by:

Saadat Anwar, MER MINI-TES Software Engineer  
Noel Gorelick, MER MINI-TES Software Engineer  
Kimberly Murray, MER MINI-TES SIS Custodian

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

D-22847  
February 1, 2007



Jet Propulsion Laboratory  
California Institute of Technology

## CHANGE LOG

DATE	SECTIONS CHANGED	REASON FOR CHANGE	REVISION
05/21/02	All	First draft	Draft
08/15/02	All	Significant changes to data format and completion of missing sections	Draft
11/07/02	3.3, Appendix A	Updates to data and header format	Draft
01/22/03	Appendix A	Updates to header format	Draft
02/06/03	All	Update	Draft
02/24/03	All	Update	Draft
03/13/03	All	Update	Draft
04/16/03	All	Updates to PDS label and source column in label keyword definitions.	Draft v1.2
05/27/03	2.4.4  Appendices A-B, D-E	Added an example of file name with missing sequence information  Updates to the PDS header structure  Updates to the HISTORY object.  Updates to the PDS keyword definitions	Draft v1.3
07/25/03	All	Updates to PDS header structure as per fields required by Mark Powell.  Revisions as per comments by Betty Sword on Draft SIS v1.2.  Expansion of several sections to describe all Mini-TES data products.	Draft v1.4
10/10/03	All	Expanded description of Mini-TES design parameters and data acquisition  Updated descriptions of data products and header contents	Draft v1.5 (Not released)
11/12/03	All	Update of fields	v1.6
12/29/03	Appendices A-C, & E	Header corrections in response to PDS data product review	v1.7
05/15/03	Sections 2 & 3, Appendix A, and other minor edits throughout	Interferogram EDR data products described in detail  Changes to keywords	v1.8

DATE	SECTIONS CHANGED	REASON FOR CHANGE	REVISION
06/15/04	Sections 2, 3, & 4  Appendix A-E	Modification to format of high level products  Change LOCAL_SOLAR_TIME to LOCAL_TRUE_SOLAR_TIME  PDS header revisions and modifications throughout	v2.0
07/15/04	Sections 1 & 2	Changed references to parties responsible for product generation	v2.0
09/15/04	Appendix A-E	PDS keyword value modifications; RELEASE_ID added to all headers	v2.1
02/01/07	Section 2.4.4	Updated rover motion counter site/position encoding; replaced pound (#) with underscore (_)	v2.2

## CONTENTS

CHANGE LOG .....	III
CONTENTS .....	V
ACRONYMS AND ABBREVIATIONS .....	VII
GLOSSARY .....	IX
<b>1. INTRODUCTION.....</b>	<b>1</b>
1.1 Purpose and Scope .....	1
1.2 Contents .....	1
1.3 Applicable Documents and Constraints .....	1
1.4 Relationships with Other Interfaces .....	2
<b>2. DATA PRODUCT CHARACTERISTICS AND ENVIRONMENT .....</b>	<b>2</b>
2.1 Instrument Overview .....	2
2.2 Data Product Overview .....	4
2.3 Data Acquisition .....	5
2.3.1 Dwell and Co-Adding of Multiple Observations .....	6
2.3.2 Data Dropouts and Extraneous Data .....	8
2.3.3 Center of the Cube .....	9
2.4 Data Processing .....	10
2.4.1 Data Processing Level .....	10
2.4.2 Data Product Generation .....	11
2.4.3 Data Flow .....	12
2.4.4 File Naming .....	12
2.5 Standards Used in Generating Data Products .....	15
2.5.1 PDS Standards .....	15
2.5.2 Time Standards .....	15
2.5.3 Coordinate Systems .....	16
2.5.4 Data Storage Conventions .....	16
2.6 Data Validation .....	16
<b>3. DETAILED DATA PRODUCT SPECIFICATIONS.....</b>	<b>17</b>
3.1 Data Product Structure and Organization .....	17
3.2 Label and Header Description .....	18
3.2.1 PDS Label .....	18
3.2.2 History Object.....	18
3.2.3 PDS TABLE Object .....	18
3.2.4 SPECTRAL QUBE Object.....	19
3.3 Data Record Format Description .....	20
3.3.1 Interferogram Experimental Data Records (EDR) .....	20
3.3.2 Radiance Experimental Data Records (EDR) .....	21
3.3.3 Telemetry Records.....	22
3.3.4 Higher Level Data Records (RDR, BTR, and EMR) .....	23

APPENDIX A: EXAMPLE OF A MINI-TES EDR LABEL ..... 24

APPENDIX B: EXAMPLE OF MINI-TES RDR LABEL ..... 35

APPENDIX C: EXAMPLE OF MINI-TES BTR AND EMR LABELS ..... 39

APPENDIX D: HISTORY OBJECT..... 41

APPENDIX E: KEYWORD DEFINITIONS..... 43

## ACRONYMS AND ABBREVIATIONS

ASCII	American Standard Code for information Interchange
ASU	Arizona State university
BTR	Brightness Temperature Record
EDR	Experiment Data Record
EMR	Emissivity Record
FEI	File Exchange Interface
FFT	Fast Fourier Transform
ICK	Instrument Counter Keeper
IDD	Instrument Deployment Device
IDPH	Instrument Data Product Header
IFGM	Interferogram
IFOV	Instantaneous Field of View
IR	Infrared
ISO	International Standards Organization
JPL	Jet Propulsion Laboratory
K	Kelvin
KB	Kilobytes
MB	Megabytes
MER	Mars Exploration Rover
MIPL	Multimission Image Processing Laboratory
MSB	Most Significant Byte
mTES	Mini-TES or Mini Thermal Emission Spectrometer
NASA	National Aeronautics and Space Administration
ODL	Object Description Language
OPGS	Operations Product Generation Subsystem
OSS	Operations Storage Server
PDS	Planetary Data System
PEL	Payload Element Lead
PMA	Pancam (or Panoramic Camera) Mast Assembly
RAT	Rock Abrasion Tool
RDR	Reduced Data Record
SCM	Spacecraft Maneuver
SFDU	Standard Formatted Data Unit
SIS	Software Interface Specification
SOAS	Science Operations Analysis Software

SPICE	Spacecraft, Planet, Instrument, C-matrix, Events
SSW	System Software
TBD	To Be Determined
TDS	Telemetry Distribution Subsystem
USGS	United States Geological Survey
V	Volts
ZPD	Zero Path Difference

## GLOSSARY

<b>TERM</b>	<b>DEFINITION</b>
Brightness temperature	Temperature of a black body required to emit the same amount of radiance at a given wave number as the observed radiance at that wave number.
Co-add	Average of multiple spectra collected at one azimuth and elevation point. Onboard application of this algorithm enhances the signal to noise ratio of the downlinked data. A co-add of one is usually referred to as no co-add. Co-add is linked to dwell such that dwell is typically a multiple of co-add.
Dwell	Collecting multiple observations at one azimuth and elevation point before moving the mirror. Dwell is linked to dwell such that dwell is always a multiple of co-add.
Emissivity	Ratio of radiant flux from a body to that from a blackbody at the same kinetic temperature.
MER-A, MER-2, Spirit	Identifies Mars Exploration Rover A, also known as MER-2, or Spirit, or the rover carrying MTES-1; this rover landed in the Gusev crater.
MER-B, MER-1, Opportunity	Identifies Mars Exploration Rover B, also known as MER-1, or Opportunity, or the rover carrying MTES-2; this rover landed in Meridiani Planum.
Sol	One Martian Day (24.6 earth hours)
Target temperature	Temperature of the target surface derived from the best fit of the radiance spectrum to a blackbody curve of known temperature.

# 1. INTRODUCTION

## 1.1 Purpose and Scope

The purpose of this SIS is to provide users of the Mini-TES data products with a detailed description of the products and a description of how they were generated, including data sources and destinations. The Mini-TES products are spectral image cubes with band-suffix back-planes. This SIS is intended to provide enough information to enable users to read and understand the data products. This SIS is intended for the scientists who will analyze the data, including those associated with the MER Project and those in the general planetary science community.

## 1.2 Contents

This Data Product SIS describes how the Mini-TES data products are acquired by the instrument, and how they are processed, formatted, labeled, and uniquely identified. The document discusses standards used in generating the product and software that may be used to access the products. The data products structure and organization are described in sufficient detail to enable a user to read the products. Finally, examples of the product labels are provided.

## 1.3 Applicable Documents and Constraints

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

1. Mars Exploration Program Data Management Plan, R. E. Arvidson and S. Slavney, Rev. 2, Nov. 2, 2000.
2. Mars Exploration Rover Project Archive Generation, Validation and Transfer Plan, R. E. Arvidson and S. Slavney, JPL D-19658, March 22, 2002.
3. MER Flight-Mission Systems ICD (FMICD), Vol. 4 Command Dictionary, MER 420-3-15.04, JPL D-20616.
4. MER Flight-Mission Systems ICD (FMICD), Vol. 7 Telemetry Dictionary, MER 420-3-15.047 JPL D-20617.
5. Pointing, Positioning, Phasing & Coordinate Systems Master (PPPCS), S.R. Doudrick, JPL D-19720, June 28, 2001.

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

6. Planetary Data System Data Preparation Workbook, February 1, 1995, Version 3.1, JPL D-7669, Part 1.
7. Planetary Data System Data Standards Reference, October 30, 2002, Version 3.4, JPL D-7669, Part 2.
8. Planetary Data System Data Dictionary, August 28, 2002, JPL D-7116.  
(Note: The Data Dictionary is being updated to include several QUBE specific changes.)

The user is referred to the following Mini-TES documents for additional information:

9. The Miniature Thermal Emission Spectrometer for the Mars Exploration Rovers, P.R. Christensen, et. Al., submitted to *J. Geophys. Res.*, 2003.
10. The Mini-TES Data Processing Guide; P.R. Christensen, Arizona State University, Tempe, AZ.
11. The website of the Davinci hyper-spectral processing system:  
<http://davinci.asu.edu/>

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

#### 1.4 Relationships with Other Interfaces

Changes to this Mini-TES data products SIS document affect the following products, software, and/or documents.

**Table 1.1: Product and Software Interfaces to this SIS**

Name	Type P=product S=software D=document	Owner
MIPL database schema	P	MIPL (JPL)
MER Mini-TES EDRs	P	Mini-TES Team, ASU
MER Mini-TES RDRs	P	Mini-TES Team, ASU
MER Mini-TES BTRs	P	Mini-TES Team, ASU
MER Mini-TES EMRs	P	Mini-TES Team, ASU
Mini-TES EDR Software	S	Mini-TES Team, ASU
Mini-TES RDR, BTR, EMR Software	S	Mini-TES Team, ASU

## 2. DATA PRODUCT CHARACTERISTICS AND ENVIRONMENT

### 2.1 Instrument Overview

The Mini-TES instrument is a Michelson moving-mirror point interferometer with a single IR detector. A pointing mirror and movable camera head work together to aim the instrument at various targets. Interferograms are collected every two seconds as the mirror scans across a target area. Each interferogram consists of approximately 1110 points, which may be converted by the in-flight FFT software into a frequency spectrum. The downlinked data record includes the raw data and various instrument telemetry

parameters. On the ground, all records from a single target sequence are assembled into a single hyperspectral image.

In addition to imaging the surface and the atmospheric targets (i.e. scenes), the Mini-TES also periodically images the internal and external calibration targets to provide calibration information. The internal calibration target is observed at least at the start and end of each scene, and usually several times throughout the complete scene acquisition. Calibration spectra thus collected are stored in a table within the header of the image cube file. Observation sequences of the external calibration target are commanded as necessary between other scenes. The external calibration observation sequence follows the same structure as a Mars scene observation sequence, including two or more internal calibration target observations; the entire external calibration sequence is stored as a single image file.

The user is referred to “The Mini-TES for MER” paper [9] for a complete description of the Mini-TES instrument; however, the following table of design parameters is provided here for convenience.

**Table 2.1: Mini-TES Design and Performance Parameters**

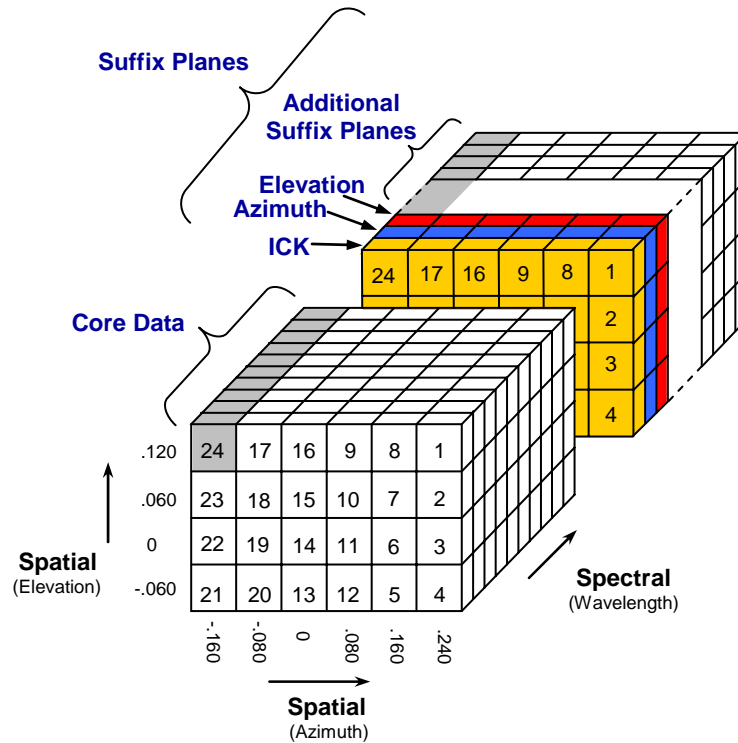
Attribute	Value
Spectral Sampling Interval (or Angular Resolution)	9.99 cm <sup>-1</sup>
Spectral Range	1997.06 - 339.5 cm <sup>-1</sup> (5.01 – 29.45 μm)
Boresight Height from Martian Surface	approx 1.4 m
Telescope Aperture	6.35 cm diameter Cassegrain
Detectors	Uncooled Alonine-doped Deuterated Triglycine Sulphate (AIDTGS) Pyroelectric Detector D* > 6 x 10 <sup>8</sup> at 20Hz
Michelson Mirror Travel	-0.25 – 0.25 mm
Mirror Velocity (physical travel)	0.0325 cm/sec
Laser Fringe Reference Wavelength	978 ± 2 nm
Interferometer Sample Rate	645 samples/sec
Azimuth Scan Range	360°
Elevation Scan Range	-50° - +30° (viewable) -180° (internal cal target)
Fields Of View	8 mrad and 20 mrad
Number of Scans to Achieve 400 SNR at 1000 cm <sup>-1</sup> with 270°K Scene Temperature and 0°C Instrument Temperature	80 (8 mrad), 2 (20 mrad)
Number of Samples per Interferogram	Typically 1110
Number of Bits per Sample of Interferogram	16
Cycle Time per Measurement (or ICK Duration)	2 seconds (1.8 seconds Michelson mirror forward scan, 0.2 seconds retrace)

Attribute	Value
Number of Samples per Spectrum	Typically 167 (but changeable via command)
Number of Bits per Sample of Spectrum	14
Primary Time of Operation	10:00AM to 3:00PM
Dimensions	9.25x6.4x6.1 in (23.5x16.3x15.5 cm)
Mass	2.40 kg
Power	5.6 watts (operating), 0.3 watts (daily average)
Operating Temperature Range – Survival and Operability	-45° to +50°C
Operating Temperature Range – Performance within Specs	-10° to +30°C
Raster start location	At the most positive (azimuth, elevation) location within the commanded region.

## 2.2 Data Product Overview

The basic unit of data from the Mini-TES instrument is a record consisting of either a single interferogram or a single radiance spectrum, and a number of telemetry words. On the ground, records from each image sequence are collected into raw EDR (Experiment Data Record) files. EDR files are processed into the higher level data products: RDR (Reduced Data Record) files, BTR (Brightness Temperature Record) files, and EMR (Emissivity Record) files.

Within the EDR, the records from a sequence are organized by azimuth and elevation into a data cube, with the data records occupying the core portion of the cube, and the per-record telemetry values stored in band-suffix back-planes. Raw internal calibration target records are separated from the scene target records and stored, in temporal order, as a binary table in the header of the cube file. A complete description of a Mini-TES data file is available in Section 3; the following figure shows only the data cube core and suffix back-planes:

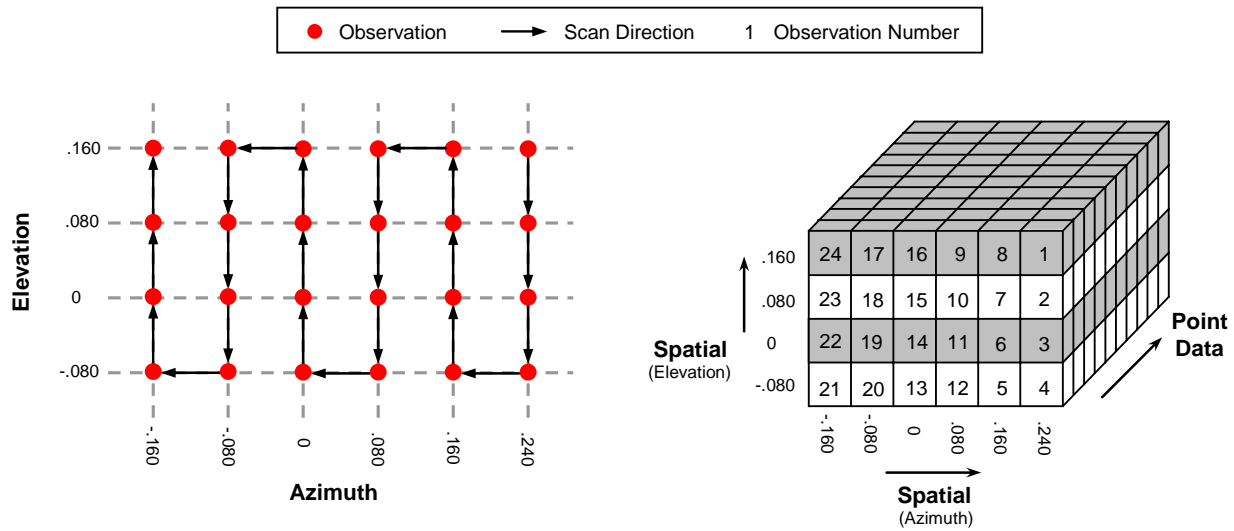


**Figure 2.1: A Mini-TES spectral radiance cube with suffix back-planes. The grayed region is the data related to pixel #24.**

The higher level data products are derived in succession from the previous level data product: RDR from EDR; BTR and EMR from RDR. In each product, the data occupies the core portion of the cube, maintaining the same azimuth and elevation organization as in the EDR. Limited telemetry values are stored in back-planes: ICK, AZIMUTH, ELEVATION, and LOCAL\_TRUE\_SOLAR\_TIME; additional standard back-planes are described in Section 3 as appropriate for each data product.

### 2.3 Data Acquisition

Each commanded image actually consists of many individual observation records, collected in a systematic order across the target area, and interspersed with internal calibration observations. Ignoring the calibration records, Figure 2.2 illustrates how the acquisition of the image sequence relates to the spatial dimensions of the assembled data cube.

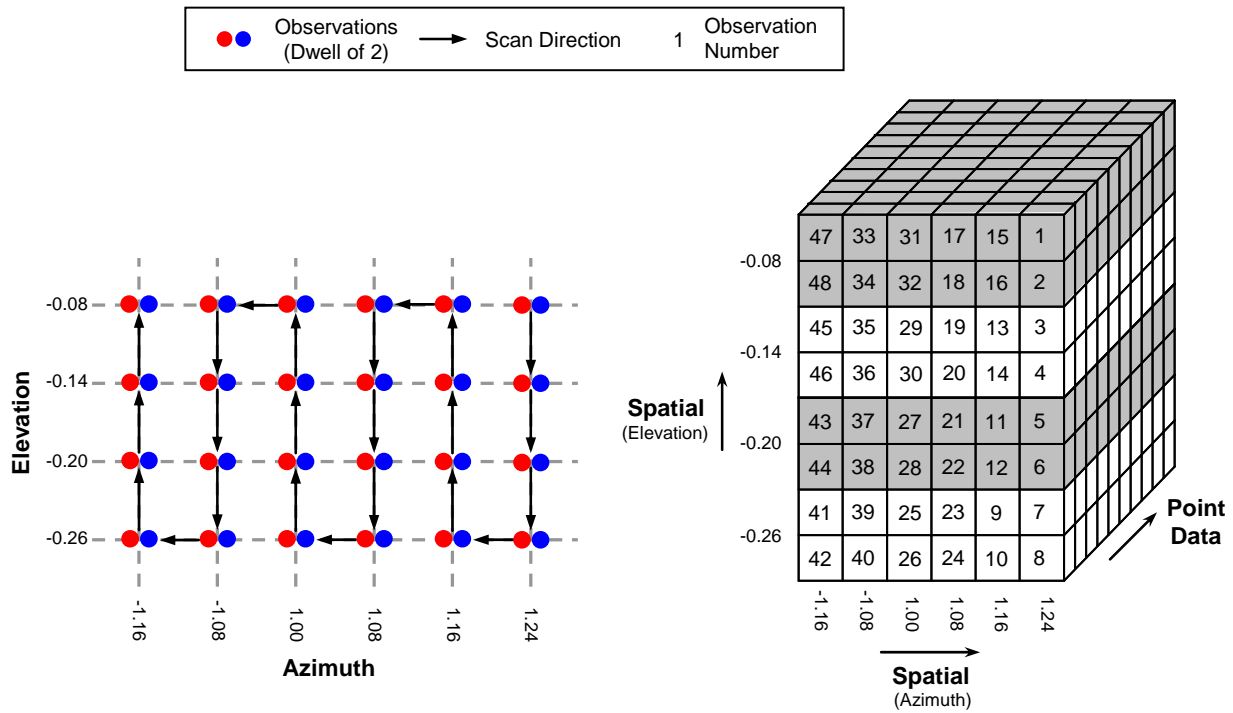


**Figure 2.2: Data acquired of a simple 6x4 image and the resulting cube core.**

**2.3.1 Dwell and Co-Adding of Multiple Observations**

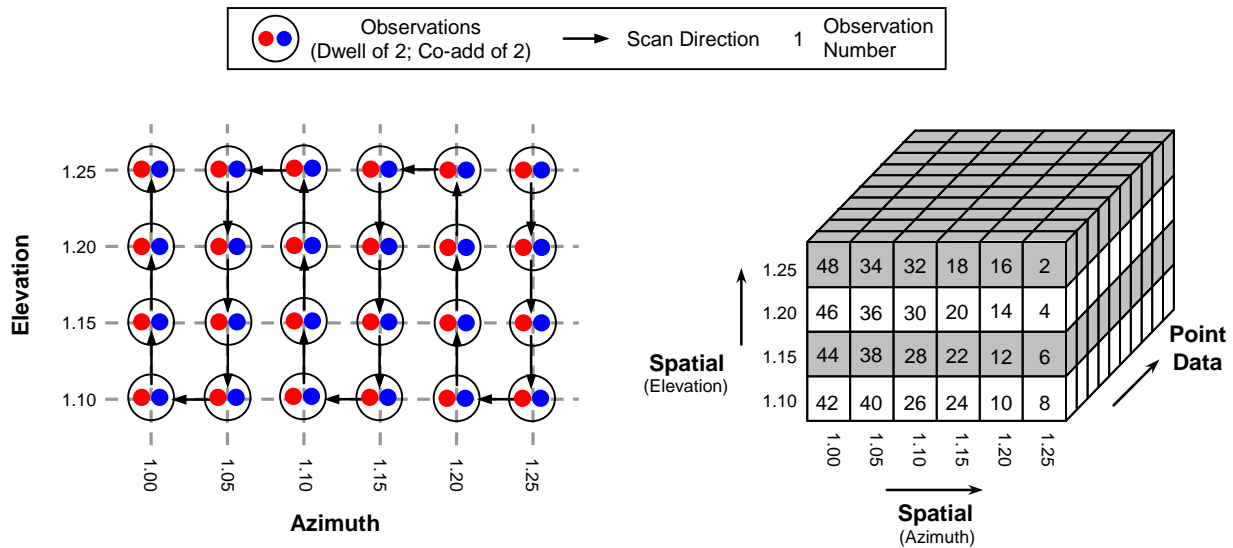
Mini-TES is capable of collecting multiple observations at a single location within a scene. These are downlinked either as individual observations, or as an average of the observations. Collecting multiple observations at a given location is referred to as dwelling at that location; averaging multiple observations is referred to as co-adding. Co-adding observations has the advantage of reducing both the signal-to-noise ratio (SNR), and the bandwidth requirement for downlinking the observation.

When Mini-TES is commanded to co-add multiple observations in an image, it should also be commanded to dwell at each location for an integer multiple of those observations. For example, a typical command may ask for a dwell of 4 observations with a co-add of every 2 observations. This example, and other valid combinations of dwell and co-add, will generate extra observations per grid point and will result in a cube with seemingly unusual dimensions. When reassembled, the extra observations are always added along the elevation spatial direction, as shown in the Figure 2.3 where a dwell of 2 observations per location is commanded without co-adding (i.e. co-add equals one):



**Figure 2.3: Data acquired with dwell of two and the resulting cube core. The gray bands indicate data at the same dwell.**

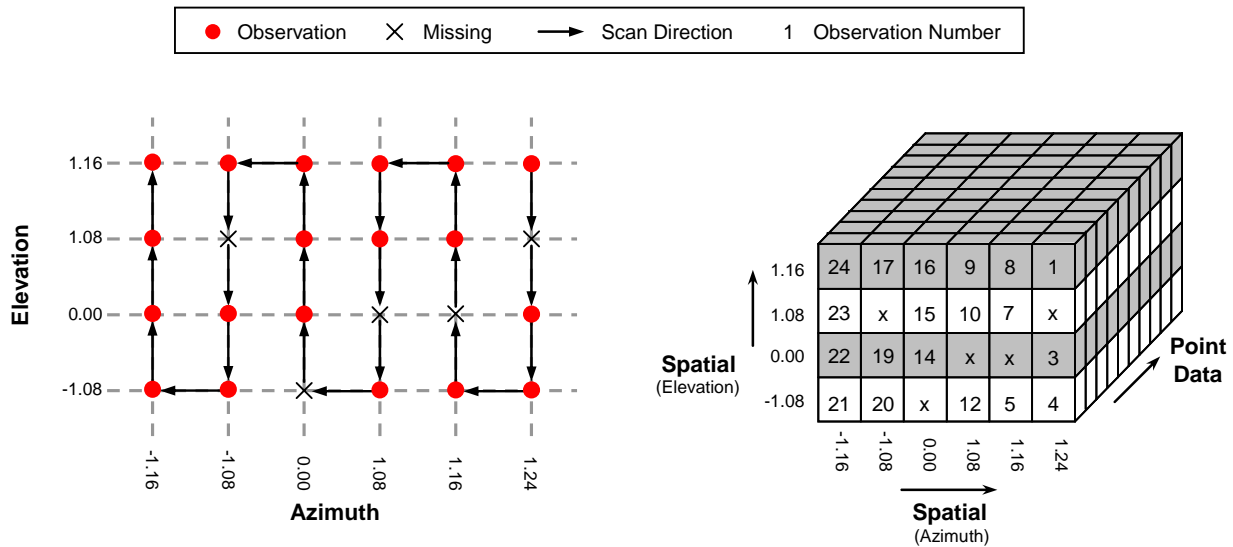
More often, a dwell of two would be paired with a co-add of two, as shown in Figure 2.4.



**Figure 2.4: Data acquired with dwell and co-add of two applied and the resulting cube core.**

### 2.3.2 Data Dropouts and Extraneous Data

Due to the complex nature of the instrument, its interplay with other subsystems on the spacecraft, and other foreseeable constraints, it is possible that some data is lost or corrupted during downlink. When this happens, the reassembled cube may be padded, as shown in Figure 2.5, to appropriately account for the missing data. The point data values are filled with the `CORE_NULL` value defined in the cube header; the back-plane values (`ICK`, `AZIMUTH`, `ELEVATION`, `LOCAL_TRUE_SOLAR_TIME`, etc.) are filled with zeros.



**Figure 2.5: Data acquisition with downlink dropouts and the resulting cube core.**

A related problem is also possible where some downlinked data is apparently extraneous, having duplicated observation numbers, duplicated spatial coordinates, or otherwise extending beyond the expected bounds of the observation sequence. If the extraneous data is adjacent to a missing record, the integrity of the reassembled cube is maintained by assuming that one of the records is incorrectly labeled and should be logically corrected to fill the gap.

The corrections are made either by renumbering, relocating, or rejecting the data as appropriate. If two records have the same observation number (`ICK`), the second record is renumbered and the details are retained in the `HISTORY` keywords `RENUMBERED_ICKS` and `RENUMBERED_ICK_NAME`. While taking the dwell parameters into account, if two records have the same spatial coordinates (`AZIMUTH` and `ELEVATION`), the second record is relocated and the details are retained in the `HISTORY` keywords `RELOCATED_ICKS` and `RELOCATED_ICK_NAME`. If there is no logical, empty location in the reassembled cube to store the extra record, it is rejected.

and the details are retained in the HISTORY keywords REJECTED\_ICKS and REJECTED\_ICK\_NAME.

### 2.3.3 Center of the Cube

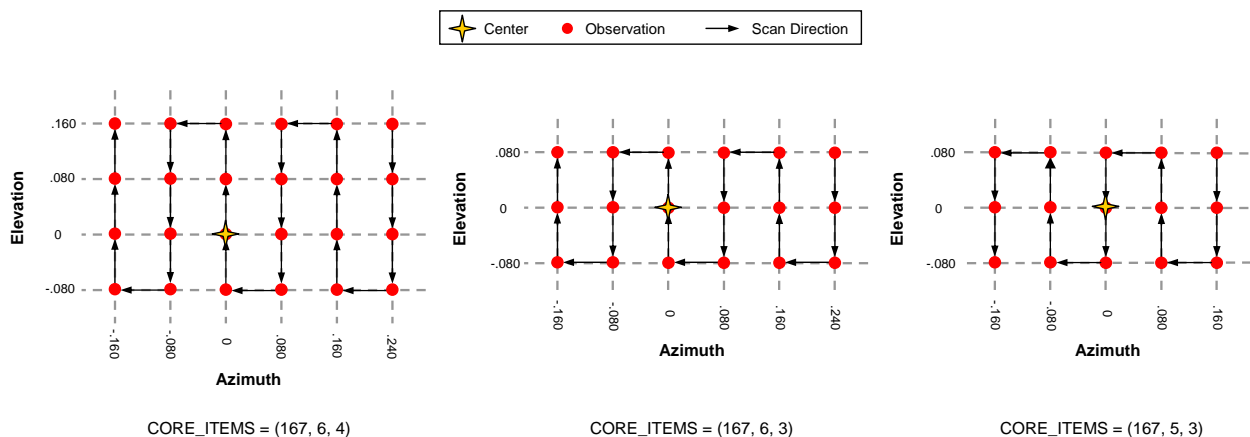
The target scene is reassembled using both the expected image dimensions, from the command sequence, and the spatial information downlinked with each observation. Commanded spatial information is available in the cube header keywords:

```

INST_CMD_CENTER_AZIMUTH
INST_CMD_CENTER_ELEVATION
INST_CMD_COLUMNS
INST_CMD_ROWS
INST_CMD_DWELL
INST_CMD_VERTICAL_SPACE
INST_CMD_HORIZONTAL_SPACE
INST_CMD_CO_ADD
    
```

All observations are downlinked with azimuth and elevation measured relative to the MER mast (see Section 2.5.3); these values are stored in cube back-planes. The spatial grid is built using an algorithm to fit the measured azimuth and elevation points, and the expected image dimensions together; the resulting core cube dimensions in order of (BAND, SAMPLE, LINE) are saved in the CORE\_ITEMS keyword. Following this convention, the azimuth spatial dimension correlates to the “SAMPLE” cube axis, and the elevation dimension correlates to the “LINE” cube axis.

Assuming no missing data or multiple observations, the commanded center of the scene is located at the exact center of the grid *only* if that location falls on one of the cube pixel values; otherwise the center of the scene is shifted so that the commanded center is associated with a nearby pixel (Figure 2.6). Note the affect that an even value of INST\_CMD\_ROWS or INST\_CMD\_COLUMNS has on the location of the commanded center as compared to an odd value for the same parameter.



**Figure 2.6: Location of commanded center (azimuth, elevation) for cubes of various dimensions**

While populating the data in the spatial grid, the software takes into account all data acquisition parameters, and labels each pixel with the correct azimuth, elevation, and tick values in the appropriate back-plane. Therefore, the center of the scene can be located by identifying the pixel with the back-plane values that most closely match the commanded center keyword values (INST\_CMD\_CENTER\_AZIMUTH, and INST\_CMD\_CENTER\_ELEVATION). Similarly, the azimuth and elevation range covered in the scene can be obtained from the corner pixel back-plane values.

In the scenes where multiple dwells are stored in individual pixels (i.e. incompletely co-added, as in Figure 2.3 above), the azimuth and elevation value is repeated in each pixel of the dwell. A search for the center of this scene will identify all pixels from the dwell at the commanded center location; as in the no dwell figures, this location may not be the exact center of the grid.

## 2.4 Data Processing

### 2.4.1 Data Processing Level

The Mini-TES data products comply with NASA processing level standards. Specifically, Mini-TES data products have been designated with the processing level standards shown in Table 2.2. All Mini-TES products are image cubes derived from the previous level product.

**Table 2.2: Processing Levels of Mini-TES Data Products**

Mini-TES Product	NASA	CODMAC	Description
N/A	Packet data	Raw – Level 1	Telemetry data stream as received at the ground station, with science and engineering data embedded.
<b>EDR</b>	Level-0	Edited – Level 2	Instrument science data (e.g., raw voltages, counts) at full resolution, time ordered, with duplicates and transmission errors removed.
<b>RDR</b>	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).
<b>BTR, EMR</b>	Level 1-B	Resampled – Level 4	Irreversibly transformed (e.g., resampled, remapped, calibrated) values of the instrument measurements (e.g., radiances, magnetic field strength).

Mini-TES Product	NASA	CODMAC	Description
N/A	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).
N/A	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.
N/A	Level 3	Derived – Level 5	Geophysical parameters mapped onto uniform space-time grids.

## 2.4.2 Data Product Generation

The Mini-TES EDRs will be generated by the Mini-TES team at ASU using the telemetry processing software “mtes2edr”. The EDRs produced will contain raw, uncalibrated data reconstructed from telemetry data products and formatted according to the EDR format defined in this SIS. Meta-data acquired from the telemetry data headers and a meta-data database will be used to populate the PDS label.

There will *not* be multiple versions of Mini-TES EDR data products. Initial processing will generate an EDR data product and pad any missing data with the value of `CORE_NULL`. Missing packets will be identified and reported for retransmission as “partial data sets”. The EDR data product will be placed into the OSS for distribution. When all retransmitted “partial datasets” have been received on the ground, the EDR file will be regenerated since the additional data can not be incorporated into an existing file. The original EDR will be *overwritten* with a new EDR, generated from both the original and retransmitted packets; this new EDR data product will replace the old file in the OSS for distribution.

RDRs, BTRs, and EMRs will be produced in succession by the Mini-TES Team and placed into the OSS for distribution. If a data product needs to be regenerated for any reason, the original version will be *overwritten* with the new version. The new version may have a version number revision in the file name (see section 2.4.4) and will contain updates to keyword values and History objects within the header.

### 2.4.2.1 Overview of Mini-TES EDR Software

ASU has produced telemetry processing software, `mtes2edr`, which generates Mini-TES EDR cubes from the raw packet data. The data stream must include a proper Instrument Data Product Header (IDPH), otherwise processing will be aborted. When executed, the software

... will discard partial records and records with suspicious lengths.

- ... will grid the data based on received azimuth and elevation data and the expected image dimensions.
- ... will pad the data grid with `CORE_NULL` (as defined in the cube header) to prevent the grid from developing non-uniform cubic dimensions; the back-plane values associated with the padded data are set to zero.
- ... will fill the various header fields in the label by using NAIF-kernels, when available.

#### **2.4.2.2 Overview of Mini-TES Higher Level Product Software**

ASU has produced data processing software, called “calibrate\_qube,” which converts EDR data into calibrated RDR data and the successive higher level data products: BTR and EMR. If necessary, interferogram data are converted via FFT into raw radiance spectra. Raw radiometric data is calibrated using available calibration target observations, both internal and external, and the known Mini-TES instrument response function as described in “The Mini-TES Data Processing Guide” [10]. The calibration software is executed on the ASU hyper-spectral analysis system called Davinci [11].

#### **2.4.3 Data Flow**

The Mini-TES EDRs are created from telemetry data products produced by the MER SSW team; the Mini-TES higher level products (RDRs, BTRs, and EMRs) are created from the Mini-TES EDR’s. All Mini-TES data products are generated by the Mini-TES team at ASU and stored on the OSS for electronic distribution. Mini-TES data products are available through MIPL’s File Exchange Interface (FEI) server for electronic distribution to remote sites via a secure subscription protocol. After a data validation period, the data products are collected with other science data and written to physical media for archiving with the Planetary Data System.

#### **2.4.4 File Naming**

MER image and non-image data products follow a similar naming scheme which adheres to ISO 9660 Level II 27.3 filename convention for PDS standards compliance. Mini-TES EDR and all higher level standard data products follow the same naming convention to remain consistent with the rest of instruments.

Each MER EDR or higher-level data product can be uniquely identified by incorporating into the product filename the Rover Mission identifier, the Instrument identifier, the Starting Spacecraft Clock count (SCLK) of the camera event, the data Product Type, the Site location, the rover Position within the site, the Sequence number, the camera “Eye”, the spectral Filter, the product Creator identifier and a Version number. For non-camera data, like Mini-TES cubes, some fields do not apply. File names are formed according to the following template:

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

where,

**rover** = (1 integer) MER rover mission identifier.

Valid Values for Mini-TES products:

Mission Name	Value
MER-1 (Also known as: MER-B, Opportunity, or the spacecraft carrying MTES-2)	<b>1</b>
MER-2 (Also known as MER-A, Spirit, or the spacecraft carrying MTES-1)	<b>2</b>

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

Valid value for Mini-TES instrument is: "T".

(See documentation for other instruments for their corresponding values.)

**sclk** = (9 integers) Starting Spacecraft Clock time.

**prod** = (3 alpha characters) Product type. Indicates the product to be an EDR or one of several types of Non-projected RDRs.

Valid Values for Mini-TES products:

Data Product	Value
Mini-TES Experimental Data Record	"EDR"
Mini-TES Reduced Data Record	"RDR"
Mini-TES Spectra Emissivity Record	"EMR"
Mini-TES Brightness Temperature Record	"BTR"

(See documentation for other instruments for their corresponding values.)

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

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

Range 0 thru 99 – "00", "01", "02" ... "99"

Range 100 thru 1035 – "A0", "A1" ... "A9", "AA", "AB" ... "AZ",  
"B0", "B1" ... "ZZ"

Range 1036 thru 1295 – "0A", "0B" ... "0Z", "1A", "1B" ... "9Z"

Range 1296 or greater – "\_\_" (2 underscores)

Example value is "AK" for value of 120.

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

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

- Range 0 thru 99 – “00 ”, “01 ”, “02 ”... “99 ”
- Range 100 thru 1035 – “A0 ”, “A1 ” ... “A9 ”, “AA ”, “AB ”... “AZ ”, “B0 ”, “B1 ”... “ZZ ”
- Range 1036 thru 1295 – “0A ”, “0B ”...”0Z ”, “1A ”, “1B ”... “9Z ”
- Range 1296 or greater – “\_\_” (2 underscores)

Example value is “AK ” for value of 120.

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

Valid values for character (position 1) in field:

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

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

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

Example value is “P0268”.

**eye** = (1 alpha character) Camera eye.

Valid value for Mini-TES instrument is: “N” for not-applicable.

(See documentation for other instruments for their corresponding values.)

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

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

“M” - MIPL (OPGS) at JPL	“A” - Arizona State University
“S” - SOAS at JPL	“N” - NASA Ames Research Center
“C” - Cornell University	“J” - Johannes Gutenberg Univ., Germany
“F” - USGS at Flagstaff	“X” - Other
“P” - Max Planck Inst., Germany	

**ver** = (1 alphanumeric) Version identifier providing uniqueness for book keeping. The integer range **1-9** provides specification for the first nine versions, and the alpha range “A-Z” provides specification for version 10 thru 35. Note that all EDRs will have only one version (“1”)

**ext** = (3 alpha characters) PDS product type extension.  
Valid values for Mini-TES products is: "**QUB**" for SPECTRAL\_QUBE

#### Mini-TES Examples:

- |    |                                 |                                                                                                                                                                                        |
|----|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| a) | 1T123456789EDR0102P3003N0A1.QUB | Rover MER-1, Mini-TES instrument, starting spacecraft clock 123456789, EDR, Site 01, Position 02, Seq P3003, produced by Arizona State University, product version 1.                  |
| b) | 2T567894321RDR01__P3575N0A1.QUB | Rover MER-2, Mini-TES instrument, starting spacecraft clock 567894321, RDR, Site 01, Position 1300 <i>overflow</i> , Seq P3575, produced by ASU, product version 1.                    |
| c) | 1T765478468BTR__02P3183N0A1.QUB | Rover MER-1, Mini-TES instrument, starting spacecraft clock 765478468, BTR, Site 1450 <i>overflow</i> , Position 02, Seq P3183, produced by ASU, product version 1.                    |
| c) | 1T874721768EMR____P3576N0A1.QUB | Rover MER-1, Mini-TES instrument, starting spacecraft clock 874721768, RDR, Site 1555 <i>overflow</i> , Position 2001 <i>overflow</i> , Seq P3576, produced by ASU, product version 1. |

## 2.5 Standards Used in Generating Data Products

### 2.5.1 PDS Standards

The Mini-TES data products comply with Planetary Data System standards for file formats and labels, as specified in the PDS Standards Reference [7]. They also adhere to ISO 9660 Level II 27.3 filename convention for PDS standards compliance.

### 2.5.2 Time Standards

The PDS label for the Mini-TES data products uses keywords containing time values, such as: `START_TIME`, `STOP_TIME`, `START_SPACECRAFT_CLOCK_COUNT`, `STOP_SPACECRAFT_CLOCK_COUNT`, and `LOCAL_TRUE_SOLAR_TIME`. The time value standard used for each of these keywords is given in Appendix E.

It should be noted that the Mini-TES standard for `LOCAL_TRUE_SOLAR_TIME` differs significantly from the MER project standard for `LOCAL_SOLAR_TIME`. Mini-TES `LOCAL_TRUE_SOLAR_TIME` is obtained from the published NAIF kernels, while MER `LOCAL_SOLAR_TIME` is based on a local, hybrid solar time.

### **2.5.3 Coordinate Systems**

The Mini-TES EDR data products are not projected into any coordinate system. The image QUBEs are maintained in the raw raster order produced by the instrument, organized to group together the data from each sequence by azimuth and elevation. Azimuth and elevation are measured relative to the MER Mast, with azimuth perpendicular to the mast and elevation parallel with the mast.

### **2.5.4 Data Storage Conventions**

The Mini-TES data products contain uncompressed data. Integers and real numbers are stored in Sun format, with the most significant byte of each word occurring first. The PDS attached-label of a Mini-TES cube is stored as ASCII text. The cube data is stored in BIP (Band Interleaved by Plane) format.

## **2.6 Data Validation**

Validation of the Mini-TES standard data products will fall into two primary categories: automated and manual. Automated validation will be performed on every data product produced for the mission; manual validation will be performed on a subset of the EDR products and the majority of the RDR products.

Automated validation will be performed as a part of the archiving process after data has been received and will be done simultaneously with the archive volume validation. Validations performed will include such things as verification that the checksum in the label matches a calculated checksum for the data product (i.e., that the data product included in the archive is identical to that produced by the real-time process), a validation of the PDS syntax of the label, a check of the label values against the database and against the index tables included on the archive volume, and checks for internal consistency of the label items. The latter include such things as verifying that the product creation date is later than the earth received time, and comparing the geometry pointing information with the specified target. As problems are discovered and/or new possibilities identified for automated verification, they will be added to the validation procedure.

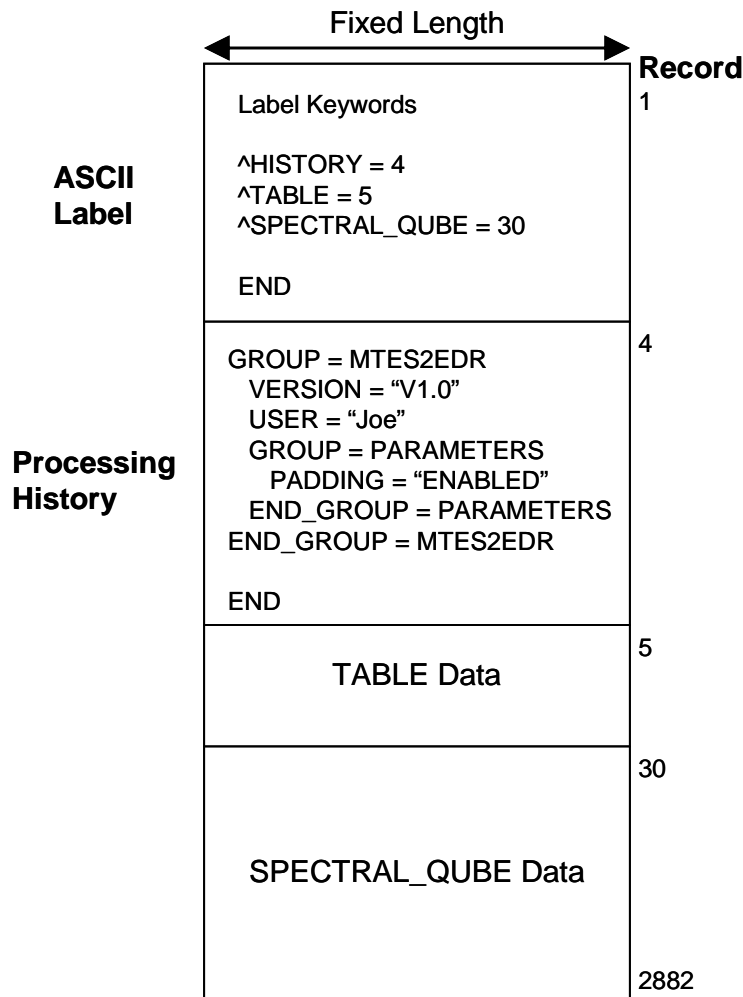
Manual validation of the data products will be performed both as spot-checking of data throughout the life of the mission, and comprehensive validation of a subset of the data (for example, several days worth of data). Validation in this case will include the visual inspection of the image or other data object for errors (like missing lines) not specified in the label parameters, verification that the image contents and apparent geometry matches that specified in the labels, verification that the product is viewable using the specified software tools, and a general check for any problems that might not have been anticipated in the automated validation procedure. Additionally, the effectiveness of the calibration routines will be visually evaluated for each of the higher level data products (RDRs, BTRs, and EMRs).

### 3. DETAILED DATA PRODUCT SPECIFICATIONS

#### 3.1 Data Product Structure and Organization

Each Mini-TES data product consists of a single file of header objects attached to an image cube, as shown in Figure 3.1. Sections 3.2 and 3.3 provide a complete description of each of the following parts of a data product:

1. an attached PDS label in ASCII format, composed of keyword-value pairs
2. a History object, describing the processing history that the data product has gone through; this ASCII object is a set of ODL-style statements, similar to the PDS keyword-value pairs
3. a table of binary, fixed-length records (available in EDRs only); table records are the internal calibration spectra and their associated telemetry values
4. an image cube, containing either the interferogram data or the radiance spectra; the data are organized by azimuth and elevation with housekeeping and telemetry values attached as suffix back-planes



### **Figure 3.1: Example of Mini-TES EDR QUBE attached label, HEADER data object, TABLE and image SPECTRAL\_QUBE**

The size of a data product will vary from sequence to sequence depending upon the size of the image commanded and the processing level completed. The average expected size of an EDR containing radiance data from a 2-hour panoramic scan will be approximately 1.7MB.

## **3.2 Label and Header Description**

### **3.2.1 PDS Label**

Every Mini-TES data file has a PDS ASCII label attached at the beginning. The PDS label describes the structure, content, and observation specifications of the data. Example Mini-TES labels are shown in Appendices A-C.

PDS labels are written in Object Description Language (ODL) [7]. PDS label statements have the form “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, as shown above in Figure 3.1.

### **3.2.2 History Object**

A HISTORY object is available in the header of each Mini-TES QUBE. Keywords in the PDS label define the location, size, and format of the HISTORY object. The HISTORY object itself is a structured series of text entries identifying all previous computer manipulations of the data in the file; the format is not intended to be compliant with PDS-ODL standards. HISTORY entries may include identification of source data, processes performed, processing parameters, and dates and times of processing. See Appendix D for an example of the entries and keywords used in the Mini-TES HISTORY objects.

### **3.2.3 PDS TABLE Object**

PDS TABLE objects are used only in the EDR data products. The TABLE object contains a variable number of fixed-length, binary records. Each file header contains a description of the table, and a pointer that indicates where the binary table data starts. The columns within the table are specified as a collection of PDS COLUMN objects. The description identifies each column in the table, detailing its name, starting position (in bytes), size (in bytes), data type, description, and scaling factors if applicable. A sample column definition follows:

```

OBJECT          = COLUMN
  NAME          = AZIMUTH
  DATA_TYPE    = IEEE_REAL
  START_BYTE    = 347
  BYTES         = 4
  DESCRIPTION   = "Scan mirror azimuth"
  UNIT          = "Radians"
END_OBJECT     = COLUMN

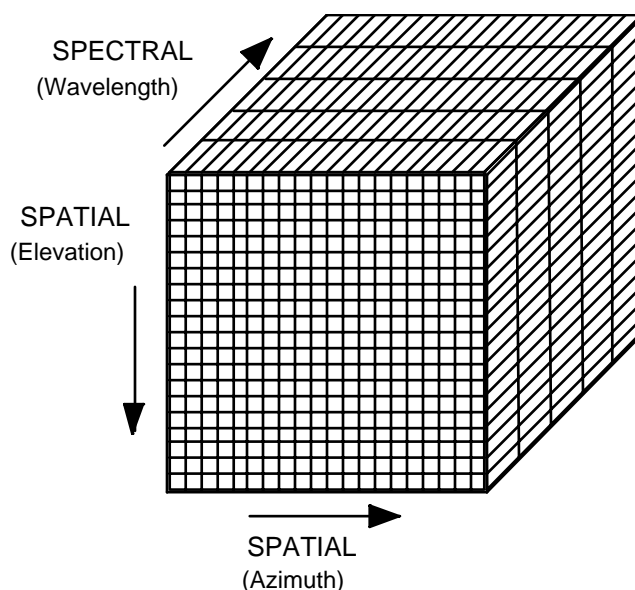
```

The CALIBRATION table in an EDR file stores the raw internal calibration records associated with the target data stored in the spectral cube. The records are sorted time-sequentially; the number of records depends on how many calibration observations were collected. An example of the PDS COLUMN description of this table is shown in Appendix A.

### 3.2.4 SPECTRAL QUBE Object

The Mini-TES data products make use of the PDS SPECTRAL\_QUBE object, adapted from the ISIS cube object and defined in the PDS Standards Reference [7].

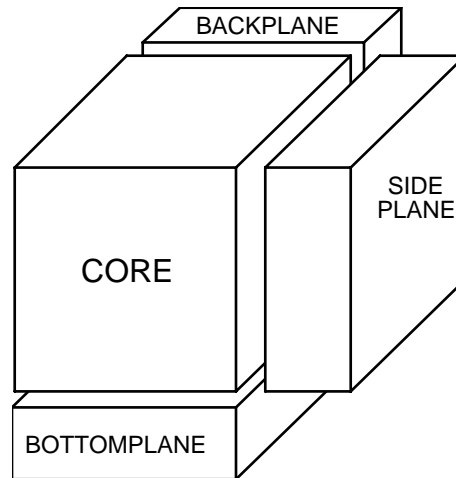
A SPECTRAL\_QUBE (abbreviated QUBE) is an array of sample values in three dimensions. The “core” of a Mini-TES QUBE is three-dimensional, with two spatial dimensions (azimuth and elevation) and one spectral dimension (wavelength or count), as shown conceptually in Figure 3.2. For radiance QUBEs, this format allows Mini-TES data to be simultaneously a set of images (at different wavelengths) of the same target area, and also a multi-point spectrum at each pixel in the target area.



**Figure 3.2: PDS SPECTRAL QUBE core structure**

Additional information may be stored in “suffix” planes (back, side, or bottom) as shown in Figure 3.3. In EDR QUBEs, suffix back-planes store the per-record telemetry values, which includes at least ICK, AZIMUTH, ELEVATION, and

LOCAL\_TRUE\_SOLAR\_TIME, but may include other values as well. As a general rule, only these four back-planes are carried forward from the EDR to any higher level products, however, some exceptions exist, as described in Section 3.3.4 and the appropriate header (Appendices B, or C).



**Figure 3.3: Exploded view of PDS SPECTRAL QUBE**

### 3.3 Data Record Format Description

A single Mini-TES experimental data record (EDR) is composed of fixed-length, raw data numbers (DN) and some number of telemetry words. There are two primary types of raw data records: interferogram records and radiance records. Sample headers of each type of EDR product is available in Appendix A.

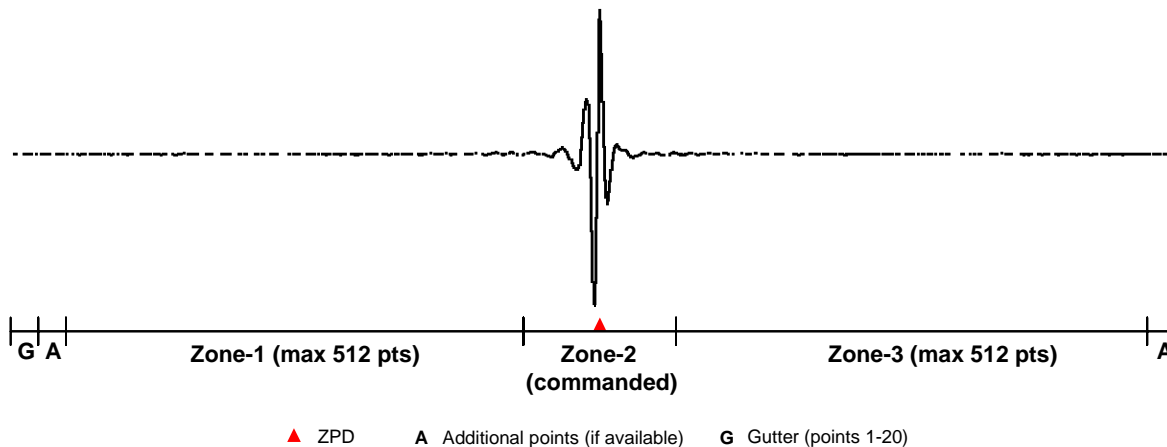
For either data type, the data values are stored in the EDR following one of two formats. The records from each internal calibration target observation are stored as a TABLE object (Section 3.2.3) in the header of the SPECTRAL\_QUBE containing the associated data. The records of all other observations are stored in the SPECTRAL\_QUBE object with the data values occupying the QUBE's CORE. The number of points (BAND axis of CORE) for either data record is selectable by a flight-software parameter, but cannot change during a sequence and, therefore, is constant within a single file. The nominal values for the number of points downlinked for either raw interferogram or radiance records is given in Table 2.1.

#### 3.3.1 Interferogram Experimental Data Records (EDR)

An interferogram is the basic data unit produced by a Mini-TES observation, however, due to the high bandwidth required to downlink data in this format, many interferograms are converted to raw radiance spectra by in-flight software. In the event that the interferogram data is selected for downlink, the interferogram data may be transmitted in either the full, uncompressed format, including the raw voltages from all points, or in a compressed format, using onboard RICE compression algorithms.

Compression of an interferogram record is a multi-part process involving the several editing and truncation steps:

- 1) The approximate ZPD point,  $Z$ , of the interferogram is determined by the in-flight software.
- 2) An equal number of interferogram data points,  $N$ , is selected from either side of the ZPD, where  $N$  is a command parameter defined per image. This set of data points, totaling  $N*2+1$  including  $Z$ , is identified as Zone-2 (Figure 3.2).
- 3) The remaining interferogram points are divided into two zones:  
 Zone-1 extends from the left of Zone-2 through data point 21 or 512 total data points, whichever is reached first. Data points 1-20 are significantly noisy and thus, are excluded from Zone-1 (Figure 3.2).  
 Zone-3 extends from the right of Zone-2 through the last data point or 512 total data points, whichever is reached first (Figure 3.2).
- 4) Each of the three zones is individually compressed using the onboard RICE compression algorithm. After downlinking, the interferogram record is uncompressed during the process of generating the EDR.



**Figure 3.2: Zones in a RICE compressed interferogram**

### 3.3.2 Radiance Experimental Data Records (EDR)

If commanded, a Mini-TES interferogram can be converted by the in-flight FFT software into a raw radiance spectrum; the raw spectrum is then downlinked following one of two available formats, described as “long” and “short”. A long packet contains numerous housekeeping and diagnostic telemetry words in addition to a raw radiance spectrum. The short packet follows the same format but contains only the bare minimum of telemetry words, in an effort to reduce the downlink overhead. A common mode for

downlinking data is the “mixed mode”, where internal calibration data is downlinked as long packets and scene data is downlinked as short packets.

### 3.3.3 Telemetry Records

The format of the telemetry records generated with each observation record is outlined in Table 3.1. Unless otherwise noted, all fields listed are available with all interferogram records and the raw radiance records downlinked as long packets; the indicated subset of fields is available with raw radiance records downlinked as short packets.

Each available telemetry value is stored in a separate back-plane attached to the EDR SPECTRAL\_CUBE. The back-plane is identified in the BAND\_SUFFIX\_NAME keyword with the same name as the corresponding TABLE column.

**Table 3.1: Mini-TES Telemetry Record Format**

TABLE Column Name	Description (units)	Bytes	Data Format	Short Packets
ICK	Count of 2-second intervals since instrument power-on.	4	MSB_INTEGER	yes
AZIMUTH	Pointing mirror azimuth (radians)	4	IEEE_REAL	yes
ELEVATION	Pointing mirror elevation (radians)	4	IEEE_REAL	yes
SPEC_EXP (Radiance only)	Scaling multiplier to convert 16-bit radiance spectrum into floating point number.	4	MSB_UNSIGNED_INTEGER	yes
NPTS	Number of fringe clock counts	4	MSB_INTEGER	yes
ZPD_MINMAX	Channel number of zero path difference as computed on the spacecraft	4	MSB_UNSIGNED_INTEGER	yes
ZONE1_WIDTH (RICE IFGM only)	Width of Zone-1 which includes points to the left of the ZPD	4	MSB_INTEGER	
ZONE3_WIDTH (RICE IFGM only)	Width of Zone-3 which includes points to the right of the ZPD	4	MSB_INTEGER	
ZPD (Radiance only)	Channel number of zero path difference around which the interferogram was folded before executing in-flight FFT	4	MSB_UNSIGNED_INTEGER	yes
COADD	Count of the spectra that have been acquired at the same location and averaged together to get this spectra	4	MSB_UNSIGNED_INTEGER	yes
CASE_TEMP_1	Case temperature - 1 (K)	4	IEEE_REAL	
CASE_TEMP_2	Case temperature - 2 (K)	4	IEEE_REAL	
INT_CAL_TEMP_1 (CALIB targets only)	Internal calibration target temperature - 1 (K)	4	IEEE_REAL	
INT_CAL_TEMP_2 (CALIB targets only)	Internal calibration target temperature - 2 (K)	4	IEEE_REAL	
EXT_CAL_TEMP_1 (CALIB targets only)	External calibration target temperature - 1 (K)	4	IEEE_REAL	
EXT_CAL_TEMP_2 (CALIB targets only)	External calibration target temperature - 2 (K)	4	IEEE_REAL	

TABLE Column Name	Description (units)	Bytes	Data Format	Short Packets
MIRROR_TEMP	Mirror temperature (K)	4	IEEE_REAL	yes
CAL_RESISTOR_TEMP	Calibration resistor temperature (K)	4	IEEE_REAL	
TLM1	+5V Secondary Power (V)	4	IEEE_REAL	
TLM2	-5V Secondary Power (V)	4	IEEE_REAL	
TLM3	+15V Secondary Power (V)	4	IEEE_REAL	
TLM4	-15V Secondary Power (V)	4	IEEE_REAL	
TLM5	+10V Control Power (V)	4	IEEE_REAL	
TLM6	-10V Control Power (V)	4	IEEE_REAL	
TLM7	+12V Postamp Power (V)	4	IEEE_REAL	
TLM8	-12V Postamp Power (V)	4	IEEE_REAL	
TLM9	Detector Temperature (K)	4	IEEE_REAL	yes
TLM10	Motor Temperature (K)	4	IEEE_REAL	
TLM11	Beam-splitter/Optics Temperature (K)	4	IEEE_REAL	
TLM12	Laser Diode 1 Temperature (K)	4	IEEE_REAL	
TLM13	Laser Diode 2 Temperature (K)	4	IEEE_REAL	
TLM14	Fringe Peak Amplitude (V)	4	IEEE_REAL	

### 3.3.4 Higher Level Data Records (RDR, BTR, and EMR)

The RDR is the radiometrically calibrated radiance derived from the raw EDR data. If the EDR is composed of interferogram records, then ground FFT processing converts the interferograms into raw radiance spectra before continuing with the radiometric calibration. The calibrated radiance data occupies the core portion of the cube, maintaining the same azimuth and elevation organization as in the EDR. The binary table of raw calibration records is not propagated from the EDR into the RDR. Limited telemetry values are stored in the back-planes: `ICK`, `AZIMUTH`, `ELEVATION`, `LOCAL_TRUE_SOLAR_TIME`, and `ZPD`. Additional back-planes, such as `PHASE_INVERT_FLAG`, `RINGING_FLAG`, and `RINGING_AMPLITUDE`, contain quality assessment values; see “The Mini-TES Data Processing Guide” [10] for a complete list and description of available quality values.

The BTR and EMR products are derived from the radiometrically calibrated RDR data, and contain calculated brightness-temperature and emissivity, respectively. The structure of these cubes is identical to the source RDR, with azimuth and elevation organized data occupying the core and limited telemetry and quality values in the back-planes. The EMR cubes have one additional standard back-plane containing derived target temperature (`TARGET_TEMP`).

## APPENDIX A: EXAMPLE OF A MINI-TES EDR LABEL

Two examples Mini-TES EDR (raw data product) labels are provided here: the first from an interferogram EDR QUBE, and the second from radiance spectral QUBE. A full explanation of the keywords and their valid values is available in Appendix E.

### Appendix A.1 RICE Compressed Interferogram EDR Label

```

PDS_VERSION_ID          = PDS3

/* FILE DATA ELEMENTS */

RECORD_TYPE             = FIXED_LENGTH
RECORD_BYTES           = 2326
FILE_RECORDS           = 94
LABEL_RECORDS          = 6

/* DATA POINTERS */

^HISTORY                = 7
^TABLE                  = 9
^SPECTRAL_QUBE         = 50

/* IDENTIFICATION DATA ELEMENTS */

DATA_SET_ID             = "MER2-M-MTES-2-EDR-V1.0"
DATA_SET_NAME           = "MER2 MARS MINIATURE THERMAL EMISSION
                          SPECTROMETER EDR V1.0"
PRODUCT_ID              = "2T135349084EDR2900P3662N0A1"
PRODUCT_TYPE            = "EDR"
PRODUCT_VERSION_ID      = "V2.1 D-22847"
PRODUCT_CREATION_TIME   = 2004-07-08T00:55:32Z
RELEASE_ID              = "0001"
PRODUCER_INSTITUTION_NAME = "ARIZONA STATE UNIVERSITY"
ROVER_MOTION_COUNTER    = (29, 0, 0, 218, 0)
ROVER_MOTION_COUNTER_NAME = (SITE, DRIVE, IDD, PMA, HGA)
SEQUENCE_ID             = "p3662"
SEQUENCE_VERSION_ID     = 0
COMMAND_SEQUENCE_NUMBER = 1
INSTRUMENT_ID           = "MINI-TES"
INSTRUMENT_NAME         = "MINIATURE THERMAL EMISSION SPECTROMETER"
INSTRUMENT_TYPE         = "THERMAL INFRARED SPECTROMETER"
INSTRUMENT_HOST_ID      = "MER2"
INSTRUMENT_HOST_NAME    = "MARS EXPLORATION ROVER 2"
MISSION_NAME            = "MARS EXPLORATION ROVER"
MISSION_PHASE_NAME      = "PRIMARY MISSION"
TARGET_NAME             = "MARS"
TARGET_TYPE             = "PLANET"
OBSERVATION_ID          = "1862270976"
PLANET_DAY_NUMBER       = 101
LOCAL_TRUE_SOLAR_TIME   = "17:06:50"
SOLAR_LONGITUDE         = 20.0217
SPACECRAFT_CLOCK_CNT_PARTITION = 1
SPACECRAFT_CLOCK_START_COUNT = 135349084.355
SPACECRAFT_CLOCK_STOP_COUNT = 135349278.355
START_TIME              = 2004-04-16T00:56:17.970Z
STOP_TIME               = 2004-04-16T00:59:31.970Z

/* TELEMETRY DATA ELEMENTS */

APPLICATION_PROCESS_ID = 31
APPLICATION_PROCESS_NAME = "MINI-TES"
APPLICATION_PROCESS_SUBTYPE_ID = 0
EARTH_RECEIVED_START_TIME = 2004-107T11:00:56.082Z
EARTH_RECEIVED_STOP_TIME = 2004-107T11:01:22.528Z
EXPECTED_PACKETS        = 41
RECEIVED_PACKETS        = 41
TELEMETRY_FORMAT_ID     = RWR
TELEMETRY_FMT_EXTENSION_TYPE = LONG

```

```

TELEMETRY_PROVIDER_ID = "SSW_MER_DP_V1.0"
TELEMETRY_SOURCE_NAME = "p3662-000-0001_001_0135349084-091.dat"
TELEMETRY_SOURCE_TYPE = "DATA PRODUCT"
TLM_INST_DATA_HEADER_ID = 5

```

```
/* INSTRUMENT COMMAND PARAMETERS */
```

```

INST_LINEAR_MOTOR_STATUS_FLAG = ON
INST_FIELD_OF_VIEW = 20<MRAD>
INST_GAIN_STATE = LOW
INST_OPTICAL_SWITCH_STATE = REDUNDANT
INST_LASER_1_STATUS_FLAG = ON
INST_LASER_2_STATUS_FLAG = OFF
INST_LASER_HEATER_STATUS_FLAG = OFF
INST_SPARE_BIT_FLAG = OFF
INSTRUMENT_COORDINATE_ID = "RVR BODY AZEL"
INSTRUMENT_COORDINATE = (0.000000<RAD>, -0.698000<RAD>)
INSTRUMENT_COORDINATE_NAME = (AZIMUTH, ELEVATION)
INST_CMD_CENTER_AZIMUTH = 3.137870<RAD>
INST_CMD_CENTER_ELEVATION = -0.698000<RAD>
INST_CMD_COLUMNS = 1
INST_CMD_ROWS = 3
INST_CMD_DWELL = 15
INST_CMD_VERTICAL_SPACE = 0.040000<RAD>
INST_CMD_HORIZONTAL_SPACE = 0.000000<RAD>
INST_CMD_CO_ADD = 1
INST_CMD_CAL_FREQUENCY = 200
INST_CMD_CAL_DWELL = 20
INST_CMD_CAL_CO_ADD = 1
INST_CMD_PHASE_ALGORITHM_NAME = RSS
INST_CMD_LOW_CHANNEL = 34
INST_CMD_HIGH_CHANNEL = 200
INST_CMD_ZONE2_WIDTH = 101

```

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

```

GROUP = ROVER_COORDINATE_SYSTEM
COORDINATE_SYSTEM_INDEX = (29, 0)
COORDINATE_SYSTEM_INDEX_NAME = (SITE, DRIVE)
COORDINATE_SYSTEM_NAME = "ROVER_FRAME"
ORIGIN_OFFSET_VECTOR = (0.000000, 0.000000, 0.000000)
ORIGIN_ROTATION_QUATERNION = (0.577599, -0.016995, -0.018844,
0.815926)
POSITIVE_AZIMUTH_DIRECTION = CLOCKWISE
POSITIVE_ELEVATION_DIRECTION = UP
QUATERNION_MEASUREMENT_METHOD = COARSE
REFERENCE_COORD_SYSTEM_INDEX = (29)
REFERENCE_COORD_SYSTEM_NAME = (SITE)
END_GROUP = ROVER_COORDINATE_SYSTEM

```

```
/* HISTORY OBJECT */
```

```

OBJECT = HISTORY
BYTES = 2518
HISTORY_TYPE = CUSTOM
INTERCHANGE_FORMAT = ASCII
END_OBJECT = HISTORY

```

```
/* TABLE DATA ELEMENTS */
```

```

OBJECT = TABLE
NAME = CALIBRATION
INTERCHANGE_FORMAT = BINARY
ROWS = 40
COLUMNS = 14
ROW_BYTES = 2342

OBJECT = COLUMN
NAME = IFGM

```

```

START_BYTE      = 1
BYTES           = 2186
DATA_TYPE       = MSB_UNSIGNED_INTEGER
ITEMS           = 1093
ITEM_BYTES      = 2
SCALING_FACTOR  = 1.0
DESCRIPTION     = "Array of interferogram values"
UNIT            = DIMENSIONLESS
END_OBJECT      = COLUMN

OBJECT          = COLUMN
NAME            = ICK
DATA_TYPE       = MSB_INTEGER
START_BYTE      = 2187
BYTES           = 4
DESCRIPTION     = "Number of scans that have occurred since
the last power on."
END_OBJECT      = COLUMN

OBJECT          = COLUMN
NAME            = AZIMUTH
DATA_TYPE       = IEEE_REAL
START_BYTE      = 2191
BYTES           = 4
UNIT            = "radians"
DESCRIPTION     = "Scan mirror azimuth"
END_OBJECT      = COLUMN

OBJECT          = COLUMN
NAME            = ELEVATION
DATA_TYPE       = IEEE_REAL
START_BYTE      = 2195
BYTES           = 4
UNIT            = "radians"
DESCRIPTION     = "Scan mirror elevation"
END_OBJECT      = COLUMN

OBJECT          = COLUMN
NAME            = NPTS
DATA_TYPE       = MSB_INTEGER
START_BYTE      = 2199
BYTES           = 4
DESCRIPTION     = "The number of fringe clocks or data words
for this ick"
END_OBJECT      = COLUMN

OBJECT          = COLUMN
NAME            = ZPD_MINMAX
DATA_TYPE       = MSB_UNSIGNED_INTEGER
START_BYTE      = 2203
BYTES           = 4
DESCRIPTION     = "Zero path difference column number as per
the flight algorithm"
END_OBJECT      = COLUMN

OBJECT          = COLUMN
NAME            = EXTERNAL_TEMPERATURES
ALIAS_NAME      = TEMPS
DATA_TYPE       = IEEE_REAL
START_BYTE      = 2207
BYTES           = 32
ITEMS           = 8
ITEM_BYTES      = 4
UNIT            = "K"
DESCRIPTION     = "Spacecraft temperatures: 1: Mini-TES
Case - 1 2: Mini-TES Case - 2 3: Internal Calibration
Target - 1 4: Internal Calibration Target - 2 5: External
Calibration Target - 1 6: External Calibration Target - 2 7:
Mirror 8: Calibration Resistor"
END_OBJECT      = COLUMN

```

```

OBJECT          = COLUMN
NAME            = INSTRUMENT_TELEMETRY
ALIAS_NAME     = TLM
DATA_TYPE      = IEEE_REAL
START_BYTE     = 2239
BYTES          = 56
ITEMS          = 14
ITEM_BYTES     = 4
DESCRIPTION    = "Instrument Telemetry: 1: +5 Volt
Secondary Power (V) 2: -5 Volt Secondary Power (V) 3: +15
Volt Secondary Power (V) 4: -15 Volt Secondary Power (V) 5:
+10 Volt Control Power (V) 6: -10 Volt Control Power (V) 7:
+12 Volt Postamp Power (V) 8: -12 Volt Postamp Power (V) 9:
Detector Temperature (K) 10: Motor Temperature (K) 11: Optics
Temperature (K) 12: Laser Diode 1 Temperature (K) 13: Laser
Diode 2 Temperature (K) 14: Fringe Peak Amplitude (V)"
END_OBJECT     = COLUMN

OBJECT          = COLUMN
NAME            = ENTROPY
DATA_TYPE      = MSB_UNSIGNED_INTEGER
START_BYTE     = 2295
BYTES          = 12
ITEMS          = 3
ITEM_BYTES     = 4
DESCRIPTION    = "RICE Compressed Data Entropy"
UNIT           = "BITS"
END_OBJECT     = COLUMN

OBJECT          = COLUMN
NAME            = CMPR_MODE
DATA_TYPE      = MSB_UNSIGNED_INTEGER
START_BYTE     = 2307
BYTES          = 12
ITEMS          = 3
ITEM_BYTES     = 4
DESCRIPTION    = "RICE Compression Mode: 0: Split 1:
Fundamental Sequence 2: Code 3: Compact"
END_OBJECT     = COLUMN

OBJECT          = COLUMN
NAME            = CMPR_LEN
DATA_TYPE      = MSB_INTEGER
START_BYTE     = 2319
BYTES          = 12
ITEMS          = 3
ITEM_BYTES     = 4
DESCRIPTION    = "RICE compressed data length: Number of
bytes downlinked as RICE compressed data."
UNIT           = "BYTES"
END_OBJECT     = COLUMN

OBJECT          = COLUMN
NAME            = LOCAL_TRUE_SOLAR_TIME
DATA_TYPE      = IEEE_REAL
START_BYTE     = 2331
BYTES          = 4
DESCRIPTION    = "Local solar time at current ick."
UNIT           = "HOURS"
END_OBJECT     = COLUMN

OBJECT          = COLUMN
NAME            = ZONE1_WIDTH
DATA_TYPE      = MSB_INTEGER
START_BYTE     = 2335
BYTES          = 4
DESCRIPTION    = "Zone-1 of rice compressed interferogram."
UNIT           = "DIMENSIONLESS"
END_OBJECT     = COLUMN

OBJECT          = COLUMN

```

```

NAME = ZONE3_WIDTH
DATA_TYPE = MSB_INTEGER
START_BYTE = 2339
BYTES = 4
DESCRIPTION = "Zone-3 of rice compressed interferogram."
UNIT = "DIMENSIONLESS"
END_OBJECT = COLUMN

END_OBJECT = TABLE

/* QUBE DATA ELEMENTS */

OBJECT = SPECTRAL_QUBE
AXES = 3
AXIS_NAME = (BAND, SAMPLE, LINE)
CORE_ITEMS = (1093, 1, 45)
CORE_ITEM_BYTES = 2
CORE_ITEM_TYPE = MSB_UNSIGNED_INTEGER
CORE_NAME = IFGM
CORE_UNIT = DIMENSIONLESS
CORE_BASE = 0.0
CORE_MULTIPLIER = 1.0
CORE_NULL = 16#0#
SUFFIX_ITEMS = (35, 0, 0)
SUFFIX_BYTES = 4
BAND_SUFFIX_NAME = (ICK, AZIMUTH, ELEVATION, NPTS, ZPD_MINMAX,
CASE_TEMP_1, CASE_TEMP_2, MIRROR_TEMP,
CAL_RESISTOR_TEMP, TLM1, TLM2, TLM3, TLM4,
TLM5, TLM6, TLM7, TLM8, TLM9, TLM10, TLM11,
TLM12, TLM13, TLM14, ZONE1_ENTROPY,
ZONE2_ENTROPY, ZONE3_ENTROPY,
ZONE1_CMPR_MODE, ZONE2_CMPR_MODE,
ZONE3_CMPR_MODE, ZONE1_CMPR_LEN,
ZONE2_CMPR_LEN, ZONE3_CMPR_LEN,
LOCAL_TRUE_SOLAR_TIME, ZONE1_WIDTH, ZONE3_WIDTH)
BAND_SUFFIX_UNIT = (DIMENSIONLESS, RADIANS, RADIANS,
DIMENSIONLESS, DIMENSIONLESS, K, K, K, K,
V, V, V, V, V, V, V, V, K, K, K, K,
K, V, BITS, BITS, BITS, DIMENSIONLESS,
DIMENSIONLESS, DIMENSIONLESS, BYTES, BYTES,
BYTES, HOURS, DIMENSIONLESS, DIMENSIONLESS)
BAND_SUFFIX_ITEM_BYTES = (4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 4)
BAND_SUFFIX_ITEM_TYPE = (MSB_INTEGER, IEEE_REAL, IEEE_REAL,
MSB_INTEGER, MSB_INTEGER, IEEE_REAL,
IEEE_REAL, IEEE_REAL, IEEE_REAL,
IEEE_REAL, IEEE_REAL, IEEE_REAL,
IEEE_REAL, IEEE_REAL, IEEE_REAL,
IEEE_REAL, IEEE_REAL, IEEE_REAL,
IEEE_REAL, IEEE_REAL,
MSB_UNSIGNED_INTEGER,
MSB_UNSIGNED_INTEGER,
MSB_UNSIGNED_INTEGER,
MSB_UNSIGNED_INTEGER,
MSB_UNSIGNED_INTEGER,
MSB_UNSIGNED_INTEGER,
MSB_UNSIGNED_INTEGER,
MSB_UNSIGNED_INTEGER,
MSB_UNSIGNED_INTEGER,
MSB_UNSIGNED_INTEGER,
MSB_UNSIGNED_INTEGER, IEEE_REAL,
MSB_INTEGER, MSB_INTEGER)

END_OBJECT = SPECTRAL_QUBE

END
```

**Appendix A.2 Radiance EDR Label**

```

PDS_VERSION_ID          = PDS3

/* FILE DATA ELEMENTS */

RECORD_TYPE             = FIXED_LENGTH
RECORD_BYTES            = 454
FILE_RECORDS            = 413
LABEL_RECORDS           = 37

/* DATA POINTERS */

^HISTORY                 = 38
^TABLE                   = 51
^SPECTRAL_CUBE           = 114

/* IDENTIFICATION DATA ELEMENTS */

DATA_SET_ID              = "MER2-M-MTES-2-EDR-V1.0"
DATA_SET_NAME            = "MER_2 MARS MINIATURE THERMAL EMISSION
                           SPECTROMETER EDR V1.0"
PRODUCT_ID               = "2T135323533EDR2800P3576N0A1"
PRODUCT_TYPE             = "EDR"
PRODUCT_VERSION_ID       = "V2.1 D-22847"
PRODUCT_CREATION_TIME    = 2004-07-08T00:55:25Z
RELEASE_ID               = "0001"
PRODUCER_INSTITUTION_NAME = "ARIZONA STATE UNIVERSITY"
ROVER_MOTION_COUNTER     = (28, 0, 0, 118, 0)
ROVER_MOTION_COUNTER_NAME = (SITE, DRIVE, IDD, PMA, HGA)
SEQUENCE_ID              = "p3576"
SEQUENCE_VERSION_ID      = 0
COMMAND_SEQUENCE_NUMBER  = 1
INSTRUMENT_ID            = "MINI-TES"
INSTRUMENT_NAME          = "MINIATURE THERMAL EMISSION SPECTROMETER"
INSTRUMENT_TYPE          = "THERMAL INFRARED SPECTROMETER"
INSTRUMENT_HOST_ID       = "MER2"
INSTRUMENT_HOST_NAME     = "MARS EXPLORATION ROVER 2"
MISSION_NAME              = "MARS EXPLORATION ROVER"
MISSION_PHASE_NAME       = "PRIMARY MISSION"
TARGET_TYPE              = "PLANET"
OBSERVATION_ID           = "1862402048"
PLANET_DAY_NUMBER        = 101
LOCAL_TRUE_SOLAR_TIME    = "10:12:17"
SOLAR_LONGITUDE          = 19.8826
SPACECRAFT_CLOCK_CNT_PARTITION = 1
SPACECRAFT_CLOCK_START_COUNT = 135323533.418
SPACECRAFT_CLOCK_STOP_COUNT = 135324317.418
START_TIME                = 2004-04-15T17:50:27.256Z
STOP_TIME                 = 2004-04-15T18:03:31.255Z

/* TELEMETRY DATA ELEMENTS */

APPLICATION_PROCESS_ID    = 31
APPLICATION_PROCESS_NAME  = "MINI-TES"
APPLICATION_PROCESS_SUBTYPE_ID = 0
EARTH_RECEIVED_START_TIME = 2004-107T01:58:17.560Z
EARTH_RECEIVED_STOP_TIME  = 2004-107T01:58:49.164Z
EXPECTED_PACKETS         = 49
RECEIVED_PACKETS         = 49
TELEMETRY_FORMAT_ID      = RCP
TELEMETRY_FMT_EXTENSION_TYPE = LONG
TELEMETRY_PROVIDER_ID    = "SSW_MER_DP_V1.0"
TELEMETRY_SOURCE_NAME     = "p3576-000-0001_001_0135323533-107.dat"
TELEMETRY_SOURCE_TYPE     = "DATA PRODUCT"
TLM_INST_DATA_HEADER_ID  = 5

/* INSTRUMENT COMMAND PARAMETERS */

INST_LINEAR_MOTOR_STATUS_FLAG = ON

```

```

INST_FIELD_OF_VIEW = 20<MRAD>
INST_GAIN_STATE = LOW
INST_OPTICAL_SWITCH_STATE = REDUNDANT
INST_LASER_1_STATUS_FLAG = ON
INST_LASER_2_STATUS_FLAG = OFF
INST_LASER_HEATER_STATUS_FLAG = OFF
INST_SPARE_BIT_FLAG = OFF
INSTRUMENT_COORDINATE_ID = "LL AZEL"
INSTRUMENT_COORDINATE = (0.000000<RAD>, 0.873000<RAD>)
INSTRUMENT_COORDINATE_NAME = (AZIMUTH, ELEVATION)
INST_CMD_CENTER_AZIMUTH = 1.096194<RAD>
INST_CMD_CENTER_ELEVATION = 0.340372<RAD>
INST_CMD_COLUMNS      = 1
INST_CMD_ROWS         = 3
INST_CMD_DWELL        = 100
INST_CMD_VERTICAL_SPACE = 0.174500<RAD>
INST_CMD_HORIZONTAL_SPACE = 0.000000<RAD>
INST_CMD_CO_ADD       = 1
INST_CMD_CAL_FREQUENCY = 200
INST_CMD_CAL_DWELL    = 20
INST_CMD_CAL_CO_ADD   = 1
INST_CMD_PHASE_ALGORITHM_NAME = RSS
INST_CMD_LOW_CHANNEL  = 34
INST_CMD_HIGH_CHANNEL = 200

/* COORDINATE SYSTEM STATE: ROVER */

GROUP                = ROVER_COORDINATE_SYSTEM
COORDINATE_SYSTEM_INDEX = (28, 0)
COORDINATE_SYSTEM_INDEX_NAME = (SITE, DRIVE)
COORDINATE_SYSTEM_NAME = "ROVER_FRAME"
ORIGIN_OFFSET_VECTOR = (0.000000, 0.000000, 0.000000)
ORIGIN_ROTATION_QUATERNION = (0.501043, -0.008716, 0.019397,
                             0.865161)
POSITIVE_AZIMUTH_DIRECTION = CLOCKWISE
POSITIVE_ELEVATION_DIRECTION = UP
QUATERNION_MEASUREMENT_METHOD = COARSE
REFERENCE_COORD_SYSTEM_INDEX = (28)
REFERENCE_COORD_SYSTEM_NAME = (SITE)
END_GROUP              = ROVER_COORDINATE_SYSTEM

/* HISTORY OBJECT */

OBJECT                = HISTORY
BYTES                 = 5679
HISTORY_TYPE          = CUSTOM
INTERCHANGE_FORMAT    = ASCII
END_OBJECT

/* TABLE DATA ELEMENTS */

OBJECT                = TABLE
NAME                  = CALIBRATION
INTERCHANGE_FORMAT    = BINARY
ROWS                  = 60
COLUMNS              = 15
ROW_BYTES             = 470

OBJECT                = COLUMN
NAME                  = RAW_RADIANCE
START_BYTE            = 1
BYTES                 = 334
DATA_TYPE             = MSB_INTEGER
ITEMS                 = 167
ITEM_BYTES           = 2
SCALING_FACTOR        = 0.000061035156250
DESCRIPTION           = "Array of raw radiance values"
UNIT                  = "transformed volts"
END_OBJECT

```

```

OBJECT          = COLUMN
  NAME          = ICK
  DATA_TYPE    = MSB_INTEGER
  START_BYTE    = 335
  BYTES         = 4
  DESCRIPTION   = "Number of scans that have occurred since
the last power on."
END_OBJECT     = COLUMN

OBJECT          = COLUMN
  NAME          = AZIMUTH
  DATA_TYPE    = IEEE_REAL
  START_BYTE    = 339
  BYTES         = 4
  UNIT          = "radians"
  DESCRIPTION   = "Scan mirror azimuth"
END_OBJECT     = COLUMN

OBJECT          = COLUMN
  NAME          = ELEVATION
  DATA_TYPE    = IEEE_REAL
  START_BYTE    = 343
  BYTES         = 4
  UNIT          = "radians"
  DESCRIPTION   = "Scan mirror elevation"
END_OBJECT     = COLUMN

OBJECT          = COLUMN
  NAME          = SPEC_EXP
  START_BYTE    = 347
  BYTES         = 4
  DATA_TYPE    = MSB_UNSIGNED_INTEGER
  DESCRIPTION   = "Spectral exponent for radiance values"
END_OBJECT     = COLUMN

OBJECT          = COLUMN
  NAME          = NPTS
  DATA_TYPE    = MSB_INTEGER
  START_BYTE    = 351
  BYTES         = 4
  DESCRIPTION   = "The number of fringe clocks or data words
for this ick"
END_OBJECT     = COLUMN

OBJECT          = COLUMN
  NAME          = ZPD
  DATA_TYPE    = MSB_UNSIGNED_INTEGER
  START_BYTE    = 355
  BYTES         = 4
  DESCRIPTION   = "Zero path difference column number"
END_OBJECT     = COLUMN

OBJECT          = COLUMN
  NAME          = ZPD_MINMAX
  DATA_TYPE    = MSB_UNSIGNED_INTEGER
  START_BYTE    = 359
  BYTES         = 4
  DESCRIPTION   = "Zero path difference column number as per
the flight algorithm"
END_OBJECT     = COLUMN

OBJECT          = COLUMN
  NAME          = COADD
  DATA_TYPE    = MSB_UNSIGNED_INTEGER
  START_BYTE    = 363
  BYTES         = 4
  DESCRIPTION   = "Co-add count for spectra"
END_OBJECT     = COLUMN

OBJECT          = COLUMN

```

```

NAME                = EXTERNAL_TEMPERATURES
ALIAS_NAME          = TEMPS
DATA_TYPE           = IEEE_REAL
START_BYTE          = 367
BYTES               = 32
ITEMS               = 8
ITEM_BYTES          = 4
UNIT                = "K"
DESCRIPTION         = "Spacecraft temperatures: 1: Mini-TES
Case - 1 2: Mini-TES Case - 2 3: Internal Calibration
Target - 1 4: Internal Calibration Target - 2 5: External
Calibration Target - 1 6: External Calibration Target - 2 7:
Mirror 8: Calibration Resistor"
END_OBJECT          = COLUMN

OBJECT              = COLUMN
NAME                = INSTRUMENT_TELEMETRY
ALIAS_NAME          = TLM
DATA_TYPE           = IEEE_REAL
START_BYTE          = 399
BYTES               = 56
ITEMS               = 14
ITEM_BYTES          = 4
DESCRIPTION         = "Instrument Telemetry: 1: +5 Volt
Secondary Power (V) 2: -5 Volt Secondary Power (V) 3: +15
Volt Secondary Power (V) 4: -15 Volt Secondary Power (V) 5:
+10 Volt Control Power (V) 6: -10 Volt Control Power (V) 7:
+12 Volt Postamp Power (V) 8: -12 Volt Postamp Power (V) 9:
Detector Temperature (K) 10: Motor Temperature (K) 11: Optics
Temperature (K) 12: Laser Diode 1 Temperature (K) 13: Laser
Diode 2 Temperature (K) 14: Fringe Peak Amplitude (V)"
END_OBJECT          = COLUMN

OBJECT              = COLUMN
NAME                = ENTROPY
DATA_TYPE           = MSB_UNSIGNED_INTEGER
START_BYTE          = 455
BYTES               = 4
DESCRIPTION         = "RICE Compressed Data Entropy"
UNIT                = "BITS"
END_OBJECT          = COLUMN

OBJECT              = COLUMN
NAME                = CMPR_MODE
DATA_TYPE           = MSB_UNSIGNED_INTEGER
START_BYTE          = 459
BYTES               = 4
DESCRIPTION         = "RICE Compression Mode: 0: Split 1:
Fundamental Sequence 2: Code 3: Compact"
END_OBJECT          = COLUMN

OBJECT              = COLUMN
NAME                = CMPR_LEN
DATA_TYPE           = MSB_INTEGER
START_BYTE          = 463
BYTES               = 4
DESCRIPTION         = "RICE compressed data length: Number of
bytes downlinked as RICE compressed data."
UNIT                = "BYTES"
END_OBJECT          = COLUMN

OBJECT              = COLUMN
NAME                = LOCAL_TRUE_SOLAR_TIME
DATA_TYPE           = IEEE_REAL
START_BYTE          = 467
BYTES               = 4
DESCRIPTION         = "Local solar time at current ick."
UNIT                = "HOURS"
END_OBJECT          = COLUMN

END_OBJECT          = TABLE

```

```

/* QUBE DATA ELEMENTS */

OBJECT          = SPECTRAL_QUBE
AXES            = 3
AXIS_NAME      = (BAND, SAMPLE, LINE)
CORE_ITEMS     = (167, 1, 300)
CORE_ITEM_BYTES = 2
CORE_ITEM_TYPE = MSB_INTEGER
CORE_NAME      = RAW_RADIANCE
CORE_UNIT      = "transformed volts"
CORE_BASE      = 0.0
CORE_MULTIPLIER = 0.000061035156250
CORE_NULL      = 16#7FFF#
SUFFIX_ITEMS   = (30, 0, 0)
SUFFIX_BYTES   = 4
BAND_SUFFIX_NAME = (ICK, AZIMUTH, ELEVATION, SPEC_EXP, NPTS,
                   ZPD, ZPD_MINMAX, COADD, CASE_TEMP_1,
                   CASE_TEMP_2, MIRROR_TEMP,
                   CAL_RESISTOR_TEMP, TLM1, TLM2, TLM3, TLM4,
                   TLM5, TLM6, TLM7, TLM8, TLM9, TLM10, TLM11,
                   TLM12, TLM13, TLM14, ENTROPY, CMPR_MODE,
                   CMPR_LEN, LOCAL_TRUE_SOLAR_TIME)

BAND_SUFFIX_UNIT = (DIMENSIONLESS, RADIANS, RADIANS,
                   DIMENSIONLESS, DIMENSIONLESS,
                   DIMENSIONLESS, DIMENSIONLESS,
                   DIMENSIONLESS, K, K, K, K, V, V, V, V, V,
                   V, V, V, K, K, K, K, K, V, BITS,
                   DIMENSIONLESS, BYTES, HOURS)

BAND_SUFFIX_ITEM_BYTES = (4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
                          4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
                          4, 4)

BAND_SUFFIX_ITEM_TYPE = (MSB_INTEGER, IEEE_REAL, IEEE_REAL,
                        MSB_UNSIGNED_INTEGER, MSB_INTEGER,
                        MSB_INTEGER, MSB_INTEGER,
                        MSB_UNSIGNED_INTEGER, IEEE_REAL,
                        IEEE_REAL, IEEE_REAL, IEEE_REAL,
                        IEEE_REAL, IEEE_REAL, IEEE_REAL,
                        IEEE_REAL, IEEE_REAL, IEEE_REAL,
                        IEEE_REAL, IEEE_REAL, IEEE_REAL,
                        MSB_UNSIGNED_INTEGER,
                        MSB_UNSIGNED_INTEGER,
                        MSB_UNSIGNED_INTEGER, IEEE_REAL)

GROUP          = BAND_BIN
BAND_BIN_ORIGINAL_BAND = (34, 35, 36, 37, 38, 39, 40, 41, 42, 43,
                          44, 45, 46, 47, 48, 49, 50, 51, 52, 53,
                          54, 55, 56, 57, 58, 59, 60, 61, 62, 63,
                          64, 65, 66, 67, 68, 69, 70, 71, 72, 73,
                          74, 75, 76, 77, 78, 79, 80, 81, 82, 83,
                          84, 85, 86, 87, 88, 89, 90, 91, 92, 93,
                          94, 95, 96, 97, 98, 99, 100, 101, 102,
                          103, 104, 105, 106, 107, 108, 109, 110,
                          111, 112, 113, 114, 115, 116, 117, 118,
                          119, 120, 121, 122, 123, 124, 125, 126,
                          127, 128, 129, 130, 131, 132, 133, 134,
                          135, 136, 137, 138, 139, 140, 141, 142,
                          143, 144, 145, 146, 147, 148, 149, 150,
                          151, 152, 153, 154, 155, 156, 157, 158,
                          159, 160, 161, 162, 163, 164, 165, 166,
                          167, 168, 169, 170, 171, 172, 173, 174,
                          175, 176, 177, 178, 179, 180, 181, 182,
                          183, 184, 185, 186, 187, 188, 189, 190,
                          191, 192, 193, 194, 195, 196, 197, 198,
                          199, 200)

BAND_BIN_CENTER = (339.5, 349.486, 359.471, 369.456,
                  379.441, 389.427, 399.412, 409.397,
                  419.383, 429.368, 439.353, 449.339,

```

459.324, 469.309, 479.294, 489.28,  
 499.265, 509.25, 519.236, 529.221,  
 539.206, 549.192, 559.177, 569.162,  
 579.147, 589.133, 599.118, 609.103,  
 619.089, 629.074, 639.059, 649.045,  
 659.03, 669.015, 679.001, 688.986,  
 698.971, 708.956, 718.942, 728.927,  
 738.912, 748.898, 758.883, 768.868,  
 778.854, 788.839, 798.824, 808.809,  
 818.795, 828.78, 838.765, 848.751,  
 858.736, 868.721, 878.707, 888.692,  
 898.677, 908.662, 918.648, 928.633,  
 938.618, 948.604, 958.589, 968.574,  
 978.56, 988.545, 998.53, 1008.52, 1018.5,  
 1028.49, 1038.47, 1048.46, 1058.44,  
 1068.43, 1078.41, 1088.4, 1098.38,  
 1108.37, 1118.35, 1128.34, 1138.32,  
 1148.31, 1158.29, 1168.28, 1178.27,  
 1188.25, 1198.24, 1208.22, 1218.21,  
 1228.19, 1238.18, 1248.16, 1258.15,  
 1268.13, 1278.12, 1288.1, 1298.09,  
 1308.07, 1318.06, 1328.05, 1338.03,  
 1348.02, 1358, 1367.99, 1377.97, 1387.96,  
 1397.94, 1407.93, 1417.91, 1427.9,  
 1437.88, 1447.87, 1457.85, 1467.84,  
 1477.82, 1487.81, 1497.8, 1507.78,  
 1517.77, 1527.75, 1537.74, 1547.72,  
 1557.71, 1567.69, 1577.68, 1587.66,  
 1597.65, 1607.63, 1617.62, 1627.6,  
 1637.59, 1647.57, 1657.56, 1667.55,  
 1677.53, 1687.52, 1697.5, 1707.49,  
 1717.47, 1727.46, 1737.44, 1747.43,  
 1757.41, 1767.4, 1777.38, 1787.37,  
 1797.35, 1807.34, 1817.32, 1827.31,  
 1837.3, 1847.28, 1857.27, 1867.25,  
 1877.24, 1887.22, 1897.21, 1907.19,  
 1917.18, 1927.16, 1937.15, 1947.13,  
 1957.12, 1967.1, 1977.09, 1987.08,  
 1997.06)

BAND\_BIN\_UNIT = "CM\*-1"  
 END\_GROUP = BAND\_BIN  
 END\_OBJECT = SPECTRAL\_CUBE  
 END

## APPENDIX B: EXAMPLE OF MINI-TES RDR LABEL

An example Mini-TES RDR (calibrated data product) label is provided here. A full explanation of the keywords and their valid values is available in Appendix E.

```

PDS_VERSION_ID      = PDS3

/* FILE DATA ELEMENTS */

RECORD_TYPE         = FIXED_LENGTH
RECORD_BYTES        = 712
FILE_RECORDS        = 29
LABEL_RECORDS       = 14

/* DATA POINTERS */

^HISTORY            = 15
^SPECTRAL_CUBE      = 20

/* IDENTIFICATION DATA ELEMENTS */

DATA_SET_ID         = "MER2-M-MTES-3-RDR-V1.0"
DATA_SET_NAME       = "MER2 MARS MINIATURE THERMAL EMISSION
SPECTROMETER RDR V1.0"
PRODUCT_ID          = "2T139516417RDR6104P3575N0A1"
PRODUCT_TYPE        = "RDR"
PRODUCT_VERSION_ID  = "1.0"
PRODUCT_CREATION_TIME = 2004-07-08T23:16:43Z
RELEASE_ID          = "0001"
PRODUCER_INSTITUTION_NAME = "ARIZONA STATE UNIVERSITY"
ROVER_MOTION_COUNTER = (61, 4, 0, 109, 0)
ROVER_MOTION_COUNTER_NAME = (SITE, DRIVE, IDD, PMA, HGA)
SEQUENCE_ID         = "p3575"
SEQUENCE_VERSION_ID = 10
COMMAND_SEQUENCE_NUMBER = 1
INSTRUMENT_ID       = "MINI-TES"
INSTRUMENT_NAME     = "MINIATURE THERMAL EMISSION SPECTROMETER"
INSTRUMENT_TYPE     = "THERMAL INFRARED SPECTROMETER"
INSTRUMENT_HOST_ID  = "MER2"
INSTRUMENT_HOST_NAME = "MARS EXPLORATION ROVER 2"
MISSION_NAME        = "MARS EXPLORATION ROVER"
MISSION_PHASE_NAME  = "PRIMARY MISSION"
TARGET_NAME         = "MARS"
TARGET_TYPE         = "PLANET"
OBSERVATION_ID      = "26676428801"
PLANET_DAY_NUMBER   = 148
LOCAL_TRUE_SOLAR_TIME = "16:00:55"
SOLAR_LONGITUDE     = 42.1105
SPACECRAFT_CLOCK_CNT_PARTITION = 1
SPACECRAFT_CLOCK_START_COUNT = 139516417.340
SPACECRAFT_CLOCK_STOP_COUNT = 139516537.340
START_TIME          = 2004-06-03T06:31:44.331Z
STOP_TIME           = 2004-06-03T06:33:44.331Z

/* TELEMETRY DATA ELEMENTS */

APPLICATION_PROCESS_ID = 31
APPLICATION_PROCESS_NAME = "MINI-TES"
APPLICATION_PROCESS_SUBTYPE_ID = 0
EARTH_RECEIVED_START_TIME = 2004-155T09:13:57.661Z
EARTH_RECEIVED_STOP_TIME = 2004-155T09:14:35.512Z
EXPECTED_PACKETS      = 24
RECEIVED_PACKETS     = 24
TELEMETRY_FORMAT_ID  = RWR
TELEMETRY_FMT_EXTENSION_TYPE = LONG
TELEMETRY_PROVIDER_ID = "SSW_MER_DP_V1.0"
TELEMETRY_SOURCE_NAME = "p3575-010-0001_001_0139516417-087.dat"
TELEMETRY_SOURCE_TYPE = "DATA PRODUCT"
TLM_INST_DATA_HEADER_ID = 5

```

```

/* INSTRUMENT COMMAND PARAMETERS */

INST_LINEAR_MOTOR_STATUS_FLAG = ON
INST_FIELD_OF_VIEW = 20<MRAD>
INST_GAIN_STATE = LOW
INST_OPTICAL_SWITCH_STATE = REDUNDANT
INST_LASER_1_STATUS_FLAG = ON
INST_LASER_2_STATUS_FLAG = OFF
INST_LASER_HEATER_STATUS_FLAG = OFF
INST_SPARE_BIT_FLAG = OFF
INSTRUMENT_COORDINATE_ID = "LL AZEL"
INSTRUMENT_COORDINATE = (0.000000<RAD>, -0.349000<RAD>)
INSTRUMENT_COORDINATE_NAME = (AZIMUTH, ELEVATION)
INST_CMD_CENTER_AZIMUTH = 1.918518<RAD>
INST_CMD_CENTER_ELEVATION = -0.307765<RAD>
INST_CMD_COLUMNS = 1
INST_CMD_ROWS = 1
INST_CMD_DWELL = 10
INST_CMD_VERTICAL_SPACE = 0.000000<RAD>
INST_CMD_HORIZONTAL_SPACE = 0.000000<RAD>
INST_CMD_CO_ADD = 1
INST_CMD_CAL_FREQUENCY = 200
INST_CMD_CAL_DWELL = 20
INST_CMD_CAL_CO_ADD = 1
INST_CMD_PHASE_ALGORITHM_NAME = RSS
INST_CMD_LOW_CHANNEL = 34
INST_CMD_HIGH_CHANNEL = 200
INST_CMD_ZONE2_WIDTH = 101

/* COORDINATE SYSTEM STATE: ROVER */

GROUP = ROVER_COORDINATE_SYSTEM
COORDINATE_SYSTEM_INDEX = (61, 4)
COORDINATE_SYSTEM_INDEX_NAME = (SITE, DRIVE)
COORDINATE_SYSTEM_NAME = "ROVER_FRAME"
ORIGIN_OFFSET_VECTOR = (0.000000, 0.000000, 0.000000)
ORIGIN_ROTATION_QUATERNION = (0.818207, -0.005802, -0.021094,
                             0.574508)
POSITIVE_AZIMUTH_DIRECTION = CLOCKWISE
POSITIVE_ELEVATION_DIRECTION = UP
QUATERNION_MEASUREMENT_METHOD = COARSE
REFERENCE_COORD_SYSTEM_INDEX = (61)
REFERENCE_COORD_SYSTEM_NAME = (SITE)
END_GROUP = ROVER_COORDINATE_SYSTEM

/* HISTORY OBJECT */

OBJECT = HISTORY
BYTES = 2947
HISTORY_TYPE = CUSTOM
INTERCHANGE_FORMAT = ASCII
END_OBJECT = HISTORY

/* QUBE DATA ELEMENTS */

OBJECT = SPECTRAL_QUBE
AXES = 3
AXIS_NAME = (BAND, SAMPLE, LINE)
CORE_ITEMS = (167, 1, 10)
CORE_ITEM_BYTES = 4
CORE_ITEM_TYPE = IEEE_REAL
CORE_NAME = CALIBRATED_RADIANCE
CORE_UNIT = "W CM**-2 SR**-1 /CM**-1"
CORE_BASE = 0.0
CORE_MULTIPLIER = 1.000000000000
CORE_NULL = 16#0#
SUFFIX_ITEMS = (11, 0, 0)
SUFFIX_BYTES = 4
BAND_SUFFIX_NAME = (ICK, AZIMUTH, ELEVATION, LOCAL_TRUE_SOLAR_TIME,
                   MISSING_CAL_FLAG, PHASE_INVERT_FLAG,

```

```

RINGING_FLAG, RINGING_AMPLITUDE,
IFGM_SPIKE_FLAG, INVERTED_SPECTRA_FLAG,
ZPD)
BAND_SUFFIX_UNIT      = (DIMENSIONLESS, RADIANS, RADIANS, BYTES,
DIMENSIONLESS, DIMENSIONLESS,
DIMENSIONLESS, DIMENSIONLESS,
DIMENSIONLESS, DIMENSIONLESS,
DIMENSIONLESS)
BAND_SUFFIX_ITEM_BYTES = (4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4)
BAND_SUFFIX_ITEM_TYPE = (MSB_INTEGER, IEEE_REAL, IEEE_REAL,
IEEE_REAL, MSB_UNSIGNED_INTEGER,
MSB_UNSIGNED_INTEGER,
MSB_UNSIGNED_INTEGER, IEEE_REAL,
MSB_UNSIGNED_INTEGER,
MSB_UNSIGNED_INTEGER,
MSB_UNSIGNED_INTEGER)

GROUP                  = BAND_BIN
BAND_BIN_ORIGINAL_BAND = (35, 36, 37, 38, 39, 40, 41, 42, 43, 44,
45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
55, 56, 57, 58, 59, 60, 61, 62, 63, 64,
65, 66, 67, 68, 69, 70, 71, 72, 73, 74,
75, 76, 77, 78, 79, 80, 81, 82, 83, 84,
85, 86, 87, 88, 89, 90, 91, 92, 93, 94,
95, 96, 97, 98, 99, 100, 101, 102, 103,
104, 105, 106, 107, 108, 109, 110, 111,
112, 113, 114, 115, 116, 117, 118, 119,
120, 121, 122, 123, 124, 125, 126, 127,
128, 129, 130, 131, 132, 133, 134, 135,
136, 137, 138, 139, 140, 141, 142, 143,
144, 145, 146, 147, 148, 149, 150, 151,
152, 153, 154, 155, 156, 157, 158, 159,
160, 161, 162, 163, 164, 165, 166, 167,
168, 169, 170, 171, 172, 173, 174, 175,
176, 177, 178, 179, 180, 181, 182, 183,
184, 185, 186, 187, 188, 189, 190, 191,
192, 193, 194, 195, 196, 197, 198, 199,
200, 201)
BAND_BIN_CENTER       = (339.5, 349.486, 359.471, 369.456,
379.441, 389.427, 399.412, 409.397,
419.383, 429.368, 439.353, 449.339,
459.324, 469.309, 479.294, 489.28,
499.265, 509.25, 519.236, 529.221,
539.206, 549.192, 559.177, 569.162,
579.148, 589.133, 599.118, 609.103,
619.089, 629.074, 639.059, 649.045,
659.03, 669.015, 679, 688.986, 698.971,
708.956, 718.942, 728.927, 738.912,
748.898, 758.883, 768.868, 778.854,
788.839, 798.824, 808.809, 818.795,
828.78, 838.765, 848.751, 858.736,
868.721, 878.706, 888.692, 898.677,
908.662, 918.648, 928.633, 938.618,
948.604, 958.589, 968.574, 978.56,
988.545, 998.53, 1008.52, 1018.5,
1028.49, 1038.47, 1048.46, 1058.44,
1068.43, 1078.41, 1088.4, 1098.38,
1108.37, 1118.35, 1128.34, 1138.32,
1148.31, 1158.3, 1168.28, 1178.27,
1188.25, 1198.24, 1208.22, 1218.21,
1228.19, 1238.18, 1248.16, 1258.15,
1268.13, 1278.12, 1288.1, 1298.09,
1308.07, 1318.06, 1328.05, 1338.03,
1348.02, 1358, 1367.99, 1377.97, 1387.96,
1397.94, 1407.93, 1417.91, 1427.9,
1437.88, 1447.87, 1457.85, 1467.84,
1477.82, 1487.81, 1497.8, 1507.78,
1517.77, 1527.75, 1537.74, 1547.72,
1557.71, 1567.69, 1577.68, 1587.66,
1597.65, 1607.63, 1617.62, 1627.6,
1637.59, 1647.57, 1657.56, 1667.55,

```

```
1677.53, 1687.52, 1697.5, 1707.49,  
1717.47, 1727.46, 1737.44, 1747.43,  
1757.41, 1767.4, 1777.38, 1787.37,  
1797.35, 1807.34, 1817.32, 1827.31,  
1837.3, 1847.28, 1857.27, 1867.25,  
1877.24, 1887.22, 1897.21, 1907.19,  
1917.18, 1927.16, 1937.15, 1947.13,  
1957.12, 1967.1, 1977.09, 1987.07,  
1997.06)  
END_GROUP          = BAND_BIN  
END_OBJECT         = SPECTRAL_CUBE  
END
```

## APPENDIX C: EXAMPLE OF MINI-TES BTR AND EMR LABELS

Mini-TES brightness-temperature (BTR) and emissivity (EMR) products are derived from the radiometrically-calibrated data record (RDR) product. The RDR label is propagated into the new products with minimal updates.

Modifying the values of the following keywords (as highlighted in bold) from an RDR label (Appendix B) will convert it into a BTR label:

```

/* IDENTIFICATION DATA ELEMENTS */
DATA_SET_ID          = "MER2-M-MTES-4-BTR-V1.0"
DATA_SET_NAME        = "MER2 MARS MINIATURE THERMAL EMISSION
                        SPECTROMETER BTR V1.0"
PRODUCT_ID           = "1T128449092BTR0100P3004N0A1"
PRODUCT_TYPE         = "BTR"
PRODUCT_VERSION_ID   = "1.0"
PRODUCT_CREATION_TIME = 2003-11-05T21:21:07Z
                      :
                      :
                      :
/* QUBE DATA ELEMENTS */
OBJECT               = SPECTRAL_QUBE
                      :
                      :
CORE_NAME            = BRIGHTNESS TEMPERATURE
CORE_UNIT            = "Kelvin"
                      :
                      :
SUFFIX_ITEMS         = (10, 0, 0)
                      :
BAND_SUFFIX_NAME     = (ICK, AZIMUTH, ELEVATION, LOCAL_TRUE_SOLAR_TIME,
                        MISSING_CAL_FLAG, PHASE_INVERT_FLAG,
                        RINGING_FLAG, RINGING_AMPLITUDE,
                        IFGM_SPIKE_FLAG, INVERTED_SPECTRA_FLAG)

```

Modifying the values of the following keywords (as highlighted in bold) from an RDR label (Appendix B) will convert it into a EMR label:

```

/* IDENTIFICATION DATA ELEMENTS */
DATA_SET_ID          = "MER2-M-MTES-4-EMR-V1.0"
DATA_SET_NAME        = "MER2 MARS MINIATURE THERMAL EMISSION
                        SPECTROMETER EMR V1.0"
PRODUCT_ID           = "1T128449092EMR0100P3004N0A1"
PRODUCT_TYPE         = "EMR"
PRODUCT_VERSION_ID   = "1.0"
PRODUCT_CREATION_TIME = 2003-11-05T21:21:18Z
                      :
                      :
                      :
/* QUBE DATA ELEMENTS */
OBJECT               = SPECTRAL_QUBE
                      :
                      :
CORE_NAME            = EMISSIVITY
CORE_UNIT            = "DIMENSIONLESS"
                      :
                      :
SUFFIX_ITEMS         = (11, 0, 0)
SUFFIX_BYTES         = 4
BAND_SUFFIX_NAME     = (ICK, AZIMUTH, ELEVATION, LOCAL_TRUE_SOLAR_TIME,
                        MISSING_CAL_FLAG, PHASE_INVERT_FLAG,
                        RINGING_FLAG, RINGING_AMPLITUDE,
                        IFGM_SPIKE_FLAG, INVERTED_SPECTRA_FLAG,

```

```
                                TARGET_TEMP)
BAND_SUFFIX_UNIT      = (DIMENSIONLESS, RADIANS, RADIANS, BYTES,
                        DIMENSIONLESS, DIMENSIONLESS,
                        DIMENSIONLESS, DIMENSIONLESS,
                        DIMENSIONLESS, DIMENSIONLESS,
                        KELVIN)
BAND_SUFFIX_ITEM_BYTES = (4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4)
BAND_SUFFIX_ITEM_TYPE = (MSB_INTEGER, IEEE_REAL, IEEE_REAL,
                        IEEE_REAL, MSB_UNSIGNED_INTEGER,
                        MSB_UNSIGNED_INTEGER,
                        MSB_UNSIGNED_INTEGER, IEEE_REAL,
                        MSB_UNSIGNED_INTEGER,
                        MSB_UNSIGNED_INTEGER, IEEE_REAL)
```

## APPENDIX D: HISTORY OBJECT

The HISTORY data object is described within the QUBE labels by the following keywords which define the structure of the embedded data object:

```

BYTES
HISTORY_TYPE
INTERCHANGE_FORMAT

```

Each program that operates on the data product will generate a new "history entry" and will concatenate the new entry onto the existing HISTORY object. All HISTORY objects follow this basic format, where the values have been replaced with keyword descriptions:

```

GROUP                = Name of the program that generated the history entry
                    (may be in a slightly modified form, due to PDS
                    naming conventions restrictions).

SOFTWARE_NAME        = Name of the program.
SOFTWARE_VERSION_ID  = Program version number.
SOFTWARE_DESC        = Program generated description and execution notes.
DATE_TIME            = Date and time, in UTC standard format, that the
                    program was executed. [yyyy-mm-ddThh:mm:ssZ]
NODE_NAME            = Machine name on which this program was ran.
USER_NAME            = User who ran this program.
PROCESSING_HISTORY_TEXT = Standard text associated with this step of
                    processing.
USER_NOTE            = User supplied brief description of program; may be
                    blank.

```

*Additional keyword value pairs may be present here, as necessary.*

```

GROUP                = Optional sub-group used to delineate the statements
                    specifying the parameters of the program.
    Keyword           = Value.
END_GROUP            = Sub-group close statement; value same as used in the
                    matching GROUP open statement.

```

*Additional sub-groups may follow, as necessary.*

```

END_GROUP            = Name of the program that generated the history entry
                    (same as used in the matching GROUP statement).

```

The following is an example of a Mini-TES EDR history entry:

```

GROUP                = MTES2EDR
PROGRAM_NAME         = "mtes2edr"
PROGRAM_VERSION_ID   = "v3.15"
DATE_TIME            = 2004-07-08T00:55:25Z
NODE_NAME            = "meramtes1x "
USER_NAME            = "John Smith"
SOFTWARE_DESC        = "mTES data -> EDR converter"
PROCESSING_HISTORY_TEXT = "CODMAC LEVEL 1 TO LEVEL 2 CONVERSION VIA
ASU MTES2EDR"
USER_NOTE            = "UNK"
INPUT_RECORD_COUNT   = 360
REJECTED_RECORDS     = (25, 79, "BOUNDS_EXCEEDED")
REJECTED_RECORDS_NAME = ("RECORD_NUMBER", "ICK",
"REJECTION_REASON")
RELOCATED_ICKS       = (50, 107, -1.47486, -0.149879, -1.4753,
-0.129933)
RELOCATED_ICK_NAME   = ("REC_NO", "ICK", "FROM_AZ", "FROM_EL",
"TO_AZ", "TO_EL")
RENUMBERED_ICKS     = (120, 921, 922)

```

```
RENUMBERED_ICK_NAME = ("REC_NO", "FROM_ICK", "TO_ICK")

GROUP                = PARAMETERS
  SPICE_FILE_NAME    = "chronos.mer2_ops"
END_GROUP            = PARAMETERS

END_GROUP            = MTES2EDR
```

The following is an example of a Mini-TES RDR, BTR, or EMR history entry:

```
GROUP = CALIBRATE_QUBE
  PROGRAM_NAME = "calibrate_qube"
  PROGRAM_VERSION_ID = "3.11"
  XPROGRAM_NAME = "davinci"
  XPROGRAM_VERSION_ID = "1.21"
  PROCESSED_DATE = "Jul 8,04 01:08"
  USER_NAME = "jsmith"
  NODE_NAME = "merantes1x "

GROUP = PARAMETERS
  CAL_OPTION = 3
  DOWNWELLING = "bb0k"
  EM_FILTER_SIZE = 7
  EM_THRESHOLD_T_COLD = 220
  EM_THRESHOLD_T_WARM = 230
  EM_WAVE1 = 500
  EM_WAVE1_COLD = 450
  EM_WAVE2 = 1400
  EM_WAVE2_COLD = 900
  FFT_ALGORITHM = 0
  INTERNAL_CAL1 = 1
  INTERNAL_CAL2 = 0
  INTERNAL_CAL_OPTION = 2
  INTERPOLATE_OPTION = 2
  INVERT_IRF_OPTON = 0
  INVERT_SCENE_OPTION = 0
  INVERTED_SCENE = 0
  MAX_TIME = 43200
  MTES = 1
  PHASE_INVERT_OPTION = 0
END_GROUP = PARAMETERS
END_GROUP = CALIBRATE_QUBE
```

## APPENDIX E: KEYWORD DEFINITIONS

The TABLE data elements and the QUBE data elements are well defined within the full PDS label, a sample of which is given in Appendix A. Definitions of various keywords which are not well defined are given below:

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
APPLICATION_PROCESS_ID	Identifies the source/process which created the data.	integer		31	TELEMETRY (Class)	EMD:product_metadata:a:apid
APPLICATION_PROCESS_NAME	Provides the name associated with the source/process which created the data.  Note: For Mars Pathfinder, the queues were distinguished on the basis of type and priority of data.	string (256)		"MINI-TES" if APPLICATION_PROCESS_ID is 31  "16#XXXX#" otherwise	TELEMETRY (Class)	String value based on APPLICATION_PROCESS_ID  Constant if APPLICATION_PROCESS_ID is 31, computed string otherwise.
APPLICATION_PROCESS_SUBTYPE_ID	Identifies the source/subprocess which created the data.	integer		0 (which is MTES_SCI)	TELEMETRY (Class)	EMD:product_metadata:a:subtype
AZIMUTH	Azimuth (or X-axis) of the scan mirror.	real	Radians		TABLE (Class)	Record header
BAND_BIN_CENTER	Center of bands corresponding to the band numbers given in BAND_BIN_ORIGINAL_BAND	real array	cm <sup>-1</sup>		SPECTRAL QUBE (Class)	Calculated from BAND_BIN_ORIGINAL_BAND values
BAND_BIN_ORIGINAL_BAND	Band numbers associated with the bands acquired/stored in the qube.  IDPH:low_channel_limit is zero-based, i.e. channel 0 is the FFT DC-term. IDPH:high_channel_limit is one higher than the last channel included as part of the BAND_BIN_ORIGINAL_BAND.	integer array		0.. approx 1110 (typically: 34..200 for RCP data)	SPECTRAL QUBE (Class)	Filled in with a step size of one from IDPH:low_channel_limit to IDPH:high_channel_limit - 1
BAND_SUFFIX_NAME	Lists names of the suffix bands in order of their occurrence.	string array		See individual products for examples.	SPECTRAL QUBE (Class)	Computed based upon product type.
BYTES	Size of the HISTORY object which follows the PDS label	integer			HISTORY (Class)	Calculated
CASE_TEMP_1 CASE_TEMP_2	See EXTERNAL_TEMPERATURES.				SPECTRAL_QUBE (Class)	Record header
CAL_RESISTOR_TEMP	See EXTERNAL_TEMPERATURES.				SPECTRAL_QUBE (Class)	Record header

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
CMPR_LEN	Length of RICE Compressed data in bytes.  <b>Note:</b> Applies to RCP & RWR products only.	RCP: integer  RWR: integer array[3]	Bytes		TABLE (Class) SPECTRAL QUBE (Class)	Record Header
CMPR_MODE	RICE Compression mode used to compress the spectra for the particular ick with which this mode is associated with.  <b>Note:</b> Applies to RCP & RWR products only.	RCP: integer  RWR: integer array[3]		0: RICE_SPLIT 1: RICE_FS 2: RICE_CODE 3: COMPACT	TABLE (Class) SPECTRAL QUBE (Class)	Byproduct of decompression
COADD	The actual co-add achieved per-spectra as compared to the commanded co-add.	integer			TABLE (Class) SPECTRAL_QUBE (Class)	Record header.
COMMAND_SEQUENCE_NUMBER	Provides a numeric identifier for a sequence of commands sent to a spacecraft or instrument.  <b>Note:</b> For MER, this is the command number which identifies the specific generating command within the specified sequence.	integer			IDENTIFICATION (Class)	IDPH:contxt_id

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
COORDINATE_SYSTEM_INDEX	<p>Instance of the coordinate frame in which the values herein are expressed. This is a group of integers that can be used to record and track the movement of a rover during surface operations.</p> <p>When in a COORDINATE_SYSTEM_STATE group, this keyword identifies which instance of the coordinate frame named by COORDINATE_SYSTEM_NAME is being defined by the group. This index is a set of integers which serve to identify coordinate system instances in a mission-specific manner.</p> <p>For MER, the indices are based on the ROVER_MOTION_COUNTER. This counter is incremented each time the rover moves (or may potentially have moved, e.g. due to arm motion). The full counter may have up to 5 values (SITE, DRIVE, IDD, PMA, HGA), but normally only the first value (for SITE frames) or the first two values (for LOCAL_LEVEL or ROVER frames) are used for defining coordinate system instances. It is legal to use any number of indices to describe a coordinate system instance, however.</p> <p>Example: COORDINATE_SYSTEM_INDEX = (1,3,2,3,2)</p> <p>For mini-TES, this keyword is only applicable within the ROVER_COORDINATE_SYSTEM group, where the COORDINATE_SYSTEM is set to "ROVER_FRAME."</p>	integer array[5]			ROVER_COORDINATE_SYSTEM (Group)	<p>Based on COORDINATE_SYSTEM_NAME.</p> <p>If COORDINATE_SYSTEM_NAME =ROVER_FRAME then use SITE and DRIVE index =APXS_FRAME or =MB_FRAME or =MI_FRAME or =RAT_FRAME or then use SITE, DRIVE, IDD, PMA and HGA indices</p>
COORDINATE_SYSTEM_INDEX_NAME	<p>An array of the formal names identifying each integer specified in COORDINATE_SYSTEM_INDEX.</p> <p>For mini-TES, this keyword is only applicable within the ROVER_COORDINATE_SYSTEM group, where the COORDINATE_SYSTEM is set to "ROVER_FRAME."</p>	string array[5]		("SITE", "DRIVE", "IDD", "PMA", "HGA")	ROVER_COORDINATE_SYSTEM (Group)	<p>Static Value: Group Dependant Should match the number of values in the COORDINATE_SYSTEM_INDEX</p>

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
COORDINATE_SYSTEM_NAME	<p>Provides the full name of the coordinate system to which the state vectors are referenced.</p> <p>When in a COORDINATE_SYSTEM_STATE group, this keyword provides the full name of the coordinate system being defined by the group. The rest of the keywords in the group describe how this coordinate system is related to some other (the "reference"). Non-unique coordinate systems (such as "SITE" for rover missions), which have multiple instances using the same name, also require COORDINATE_SYSTEM_INDEX to completely identify the coordinate system.</p> <p>For mini-TES, this keyword is only applicable within the ROVER_COORDINATE_SYSTEM group, where the COORDINATE_SYSTEM is set to "ROVER_FRAME."</p>	string[30]		<p>"ROVER_FRAME", "SITE_FRAME", "MAST_FRAME", "LOCAL_LEVEL_FRAME",</p> <p><u>IDD only</u> 0="MI_FRAME" 1="RAT_FRAME" 2="MB_FRAME" 3="APXS_FRAME",</p>	ROVER_COORDINATE_SYSTEM (Group)	Group dependant Source: IDPH:Frame. (ROVER_FRAME)
DATA_SET_ID	<p>A unique alphanumeric identifier for a data set or a data product. The DATA_SET_ID value for a given data set or product is constructed according to flight project naming conventions. In most cases the DATA_SET_ID is an abbreviation of the DATA_SET_NAME.</p> <p>Note: In the PDS, the values for both DATA_SET_ID and DATA_SET_NAME are constructed according to standards outlined in the Standards Reference.</p>	string(40)		<p>"MER1-M-MTES-2-EDR-V1.0", "MER1-M-MTES-3-RDR-V1.0", "MER1-M-MTES-4-BTR-V1.0", "MER2-M-MTES-4-EMR-V1.0", "MER2-M-MTES-2-EDR-V1.0", "MER2-M-MTES-3-RDR-V1.0", "MER2-M-MTES-4-BTR-V1.0", "MER2-M-MTES-4-EMR-V1.0"</p>	IDENTIFICATION (Class)	<p>Constructed using template: a-b-c-d-e-f. Where: "a" is params.txt: DATA_SET_ID_SC_PART_XXX "b" is set to "M" "c" is param.txt: DATA_SET_ID_INST_PART_XXX "d" is the product's CODMAC level "e" is the three-digit encoding of the product-type "f" is set to "V1.0" and version will increment with a rerelease of the data-set.</p>

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
DATA_SET_NAME	<p>The data_set_name element provides 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. Example value: MR9/VO1/VO2 MARS IMAGING SCIENCE SUBSYSTEM/VIS 5 CLOUD V1.0.</p>	string(60)		<p>"MER1 MARS MINIATURE THERMAL EMISSION SPECTROMETER EDR V1.0",</p> <p>"MER1 MARS MINIATURE THERMAL EMISSION SPECTROMETER RDR V1.0",</p> <p>"MER1 MARS MINIATURE THERMAL EMISSION SPECTROMETER BTR V1.0",</p> <p>"MER1 MARS MINIATURE THERMAL EMISSION SPECTROMETER EMR V1.0",</p> <p>"MER2 MARS MINIATURE THERMAL EMISSION SPECTROMETER EDR V1.0",</p> <p>"MER2 MARS MINIATURE THERMAL EMISSION SPECTROMETER RDR V1.0",</p>	IDENTIFICATION (Class)	<p>Constructed using template: a-b-c-d-e. Where:                      "a" is params.txt: DATA_SET_NAME_S C_PART_XXX                      "b" is set to "MARS"                      "c" is paramx.txt: DATA_SET_NAME_I NST_PART_XXX                      "d" is the product type                      "e" is set to "V1.0" and version will increment with a rerelease of the data-set.</p>

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
(DATA_SET_NAME)				"MER2 MARS MINIATURE THERMAL EMISSION SPECTROMETER BTR V1.0",  "MER2 MARS MINIATURE THERMAL EMISSION SPECTROMETER EMR V1.0"		
EARTH_RECEIVED_START_TIME	Provides the beginning time at which telemetry was received during a time period of interest. This is represented in UTC system format.	datetime		YYYY-MM-DDThh:mm:ss[.fff]Z	TELEMETRY (Class)	Taken as the first of EMD:product_metadata:part_list:part:ert
EARTH_RECEIVED_STOP_TIME	Provides the ending time at which telemetry was received during a time period of interest. This is represented in UTC system format.	datetime		YYYY-MM-DDThh:mm:ss[.fff]Z	TELEMETRY (Class)	Taken as the last of EMD:product_metadata:part_list:part:ert
ELEVATION	Elevation (Y-axis) of the scan mirror.	real	Radians	-0.872665 .. +0.523599 (or -50° .. +30°) for scenes  < -1.04720 (or -60°) for calibration	TABLE (Class)	Record header
ENTROPY	Number of bits used as entropy bits during RICE Compression of data. This number of bits are used to split the input words while compressing the data in RICE_SPLIT mode.  <b>Note:</b> Applies to RCP & RWR products only.	RCP: integer  RWR: integer array[3]	Bits	0 or more	TABLE (Class) SPECTRAL QUBE (Class)	Byproduct of decompression
EXPECTED_PACKETS	Total number of packets expected to be received as part of the data product as per the EMD file.	integer		0 or more	TELEMETRY (Class)	EMD:product_metadata:part_list:total

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
EXTERNAL_TEMPERATURES	Spacecraft Temperatures 1 – Mini-TES Case Temperature 1 2 – Mini-TES Case Temperature 2 3 – Internal Calibration Target Temperature 1 4 – Internal Calibration Target Temperature 2 5 – External Calibration Target Temperature 1 6 – External Calibration Target Temperature 2 7 – Mirror Temperature 8 – Calibration Resistor Temperature  <b>Note:</b> 1..6,8 apply to RAW, SPT, RCP, RWR extension-LONG only, while 7 applies to all data record types.	real array[8]	Kelvin		TABLE (Class)	Calculated from corresponding values stored in the record header
FILE_RECORDS	Indicates the number of physical file records, including both label records and data records.  <b>Note:</b> In the PDS the use of FILE_RECORDS along with other file-related data elements is fully described in the Standards Reference.	integer		Positive integer greater than zero	FILE DATA ELEMENT (Class)	Calculated
HISTORY_TYPE	Type of the history object which follows the PDS label	string		CUSTOM	HISTORY (Class)	Static Value
ICK	Number of scans that have occurred since the last power on.  <b>Note:</b> Does not apply to RAW data.	integer			TABLE (Class)	Record header
ID	Identifier of the mineral spectrum in the global mineral library maintained in ASU.	string			MINERAL LIBRARY (Class)	User defined
IFGM_SPIKE_FLAG	Source data quality flag. Applicable to higher level data products. See "The Mini-TES Data Processing Guide" for details.	Boolean (represented as integer)		0 – no spike in ifgm 1 – spike in ifgm	SPECTRAL_OUBE (Class)	Byproduct of calibration
INST_CMD_CAL_CO_ADD	Commanded value of co-add for internal calibration observations. This gives the number of calibration observations that will be averaged together (into one observation) before it is downlinked.	integer		1..255	INSTRUMENT COMMAND (Class)	IDPH:mtes:cal_co_add_limit
INST_CMD_CAL_DWELL	Commanded value of the number of icks to collect during a calibration observation.	integer		1..255	INSTRUMENT COMMAND (Class)	IDPH:mtes:cal_dwell

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
INST_CMD_CAL_FREQUENCY	<p>Commanded value of the minimum number of icks that have to expire from the end of the last internal calibration look before a new set of calibration looks are taken.</p> <p>The calibration frequency is in some sense a measure of our confidence in the accuracy of the internal calibration target.</p> <p>From the flight software: Calibration observations come at inconsistent frequency through the course of an image. Following guidelines drive the occurrence of a cal:</p> <ul style="list-style-type: none"> <li>• A sync counter cal_countdown is initialized on arrival at the cal target to a value of cal_dwell + cal_freq</li> <li>• On every ick (or sync) cal_countdown is decremented</li> <li>• At the end of cal_dwell the cal_countdown equals cal_freq</li> <li>• mTES moves back to continue with scene observations, while cal_countdown decrements</li> <li>• When cal_countdown becomes zero, a cal_flag is set.</li> <li>• Actual cal does not happen until the end of the current scene dwell finishes</li> </ul>	integer		1..65535	INSTRUMENT COMMAND (Class)	IDPH:mtes:cal_freq
INST_CMD_CENTER_AZIMUTH	Commanded value of center azimuth of the qube data.	real	radians	0..2 $\pi$ (or full 360°)	INSTRUMENT COMMAND (Class)	IDPH:mtes:mtes_az
INST_CMD_CENTER_ELEVATION	Commanded value of center elevation of the qube data.	real	radians	-0.872665 .. 0.523599 (or -50° .. +30°)	INSTRUMENT COMMAND (Class)	IDPH:mtes:mtes_el
INST_CMD_CO_ADD	Commanded value of: scene spectra to average together.	integer		1..255	INSTRUMENT COMMAND (Class)	IDPH:mtes:co_add_limit
INST_CMD_COLUMNS	Commanded value of the number of columns (or azimuth steps) to acquire.	integer		1..65535	INSTRUMENT COMMAND (Class)	IDPH:mtes:cols
INST_CMD_DWELL	Commanded value of the number of icks to acquire at one azimuth, elevation.	integer		1..255	INSTRUMENT COMMAND (Class)	IDPH:mtes:dwell

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
INST_CMD_HIGH_CHANNEL	Commanded value of the end channel number to acquire minus one.	integer		Nominal value: INST_CMD_LOW_CHANNEL + 167 - 1	INSTRUMENT COMMAND (Class)	Computed as: IDPH:mtes:hi_chan_limit - 1
INST_CMD_HORIZONTAL_SPACE	Commanded value of horizontal space between acquired columns.	real	radians		INSTRUMENT COMMAND (Class)	IDPH:mtes:h_space
INST_CMD_LOW_CHANNEL	Commanded value of the start channel number to acquire. This value is zero-based, i.e. it starts at zero. Where, channel zero is the DC component that comes out of the FFT data, when the source data product (from which the EDR was built) was RCP.	integer		Nominal value: 35	INSTRUMENT COMMAND (Class)	IDPH:mtes:lo_chan_limit
INST_CMD_PHASE_ALGORITHM_NAME	Commanded value of the phase correction algorithm to use on board.	string(10)		0 - NONE 1 - MERTZ 2 - RSS	INSTRUMENT COMMAND (Class)	IDPH:mtes:phase_correction_algorithm
INST_CMD_ROWS	Commanded value of the number of rows (or elevation steps) to acquire.	integer		1..255	INSTRUMENT COMMAND (Class)	IDPH:mtes:rows
INST_CMD_VERTICAL_SPACE	Commanded value of the vertical space between consecutive rows	real	radians		INSTRUMENT COMMAND (Class)	IDPH:mtes:v_space
INST_CMD_ZONE2_WIDTH	This field gives the commanded length of the center-burst region of the interferogram, which is compressed separately from zone-1 and zone-2. For details, see ZONE1_WIDTH.  Note: Applicable to RWR products only	integer			INSTRUMENT COMMAND (Class)	Decompressed Data Record Header

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
INSTRUMENT_COORDINATE	<p>An array of coordinate parameters.</p> <p>For MER, the parameters will be a set of azimuth and elevation value (radians) or a set of xyz position parameters (m). If the INSTRUMENT_COORDINATE_ID is an IVP these values are ignored.</p> <p>The units can be identified in the Units column by applying the following indices for the frames defined in keyword INSTRUMENT_COORDINATE_ID:</p> <p>0 = "NONE"                      1 = "MAST AZEL"                      2 = "RVR BODY AZEL"                      3 = "RVR BODY 3DPNT"                      4 = "LL AZEL"                      5 = "LL 3DPNT"                      6 = "SITE 3DPNT"                      7 = "IVP_EARTH"                      8 = "IVP_PHOBOS"                      9 = "IVP_DEIMOS"                      10 = "IVP_MARS_P3"                      11 = "IVP_MARS_P4"</p> <p>Note: IDPH version 4 and higher products only</p>	<p>NULL for index 0</p> <p>double array [2] for indices 1,2,4,8</p> <p>double array[3] for indices 3,5,6</p>	<p>Radians for index 1,2,4,8</p> <p>Meters for indices 3,5,6</p>		INSTRUMENT COMMAND (Class)	IDPH:{Coord 1, Coord 2, Coord 3}
INSTRUMENT_COORDINATE_ID	<p>Identifies the frame in which the INSTRUMENT_COORDINATE values are given.</p> <p>Note: IDPH version 4 and higher products only</p>	string		<p>0 – NONE                      1 – MAST AZEL                      2 – RVR BODY AZEL                      3 – ROVER BODY 3DPNT                      4 – LL AZEL                      5 – LL 3DPNT                      6 – SITE 3DPNT                      7 – IVP_EARTH                      8 – IVP_PHOBOS                      9 – IVP_DEIMOS                      10 – IVP_MARS_P3                      11 – IVP_MARS_P4</p>	INSTRUMENT COMMAND (Class)	IDPH:Frame

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
INSTRUMENT_COORDINATE_NAME	<p>Name associated with each coordinate INSTRUMENT_COORDINATE.</p> <p>Following list gives the INSTRUMENT_COORDINATE_IDs and their associated coordinate names:</p> <p>Note: IDPH version 4 and higher products only</p>	<p>NULL for index 0</p> <p>string array [2] for indices 1,2,4,8</p> <p>string array[3] for indices 3,5,6</p>		<p>NULL for index 0</p> <p>(MAST AZIMUTH, MAST MIRROR ACTUATOR ANGLE) for index 1</p> <p>(AZIMUTH, ELEVATION) for index 2, 4, 8</p> <p>(X,Y,Z) for index 3, 5, 6</p>	INSTRUMENT COMMAND (Class)	Lookup based on IDPH:Frame
INST_FIELD_OF_VIEW	Gives the Instantaneous Field of View (or the IFOV) of the instrument used while acquiring the data product	real	milli-radians	8 mrad 20 mrad	INSTRUMENT COMMAND (Class)	<p>IDPH:Command Word(Bit 1)</p> <p>Note: bits numbered 0-7</p>
INST_GAIN_STATE	Selects the gain state of the mini-TES analog signal amplifier.	string		LOW HIGH	INSTRUMENT COMMAND (Class)	<p>IDPH:Command Word(Bit 2)</p> <p>1 – “LOW” 0 – “HIGH”</p>
INSTRUMENT_HOST_ID	Provides a unique identifier for the host where an instrument is located. This host can be either a spacecraft or an earth base (e.g., and observatory or laboratory on the earth). Thus, INSTRUMENT_HOST_ID can contain values which are either SPACECRAFT_ID values or EARTH_BASE_ID values.	string(6)		“MER2”, “MER1”	IDENTIFICATION (Class)	<p>presets.txt:INSTRUMENT_HOST_ID_XXX</p> <p>with defaults built into the code.</p>
INSTRUMENT_HOST_NAME	<p>Provides the full name of the host on which an instrument is based. This host can be either a spacecraft or an earth base. Thus, the INSTRUMENT_HOST_NAME element can contain values which are either SPACECRAFT_NAME values or EARTH_BASE_NAME values.</p> <p>Note that mosaics may contain more than one value in an array.</p>	string		<p>“MARS EXPLORATION ROVER A”</p> <p>“MARS EXPLORATION ROVER B”</p>	IDENTIFICATION (Class)	<p>presets.txt:INSTRUMENT_HOST_NAME_XX X</p> <p>With defaults built into the code.</p>

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
INSTRUMENT_ID	<p>Provides an abbreviated name or acronym which identifies an instrument.</p> <p>Note: INSTRUMENT_ID is not a unique identifier for a given instrument. Note also that the associated INSTRUMENT_NAME element provides the full name of the instrument.</p> <p>Example values: IRTM (for Viking Infrared Thermal Mapper), PWS (for plasma wave spectrometer).</p>	string(12)		"MINI-TES"	IDENTIFICATION (Class)	Fixed.
INST_LASER_1_STATUS_FLAG	<p>Primary Laser status. This is one of the two 980nm monochromatic lasers which generates the A/D sampling frequency.</p> <p>Notes:</p> <ul style="list-style-type: none"> <li>• Only one laser can be on at one time. If both the lasers are commanded to be turned on, the backup laser will be turned on and the primary will be turned off.</li> <li>• If the michaelson motor is turned off, then both the lasers are turned off, irrespective of the commanded laser status</li> <li>• If both the lasers are commanded to be off, then michaelson motor off is asserted</li> </ul>	string		ON OFF	INSTRUMENT COMMAND (Class)	IDPH:Command Word(Bit 4)  0 – "OFF" 1 – "ON"
INST_LASER_2_STATUS_FLAG	<p>Backup Laser status. (see notes for the INST_LASER_1_STATUS)</p>	string		ON OFF	INSTRUMENT COMMAND (Class)	IDPH:Command Word(Bit 5)  0 – "OFF" 1 – "ON"
INST_LASER_HEATER_STATUS_FLAG	<p>Status of the laser heater</p> <p>Purpose?</p>	string		ON OFF	INSTRUMENT COMMAND (Class)	IDPH:Command Word(Bit 6)  0 – "OFF" 1 – "ON"
INST_LINEAR_MOTOR_STATUS_FLAG	<p>Status of the Michelson Motor (see notes for the INST_LASER_1_STATUS)</p>	string		ON OFF	INSTRUMENT COMMAND (Class)	IDPH:Command Word(Bit 0)  0 – "OFF" 1 – "ON"
INSTRUMENT_NAME	<p>Provides the full name of the instrument associated with the INSTRUMENT_ID.</p>	string		"MINIATURE THERMAL EMISSION SPECTROMETER "	IDENTIFICATION (Class)	Fixed

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
INST_OPTICAL_SWITCH_STATE	Indicates which of the optical switches is being used by the mini-TES scanning mirror to dither. The scanning mirror dithers between the home/retrace position and the optical switch until the start of a scan data acquisition. Note that there are two such switches, for the sake for redundancy, in case one fails.	string		PRIMARY REDUNDANT	INSTRUMENT COMMAND (Class)	IDPH:Command Word(Bit 3)  0 – "REDUNDANT" 1 – "PRIMARY"
INST_SPARE_BIT_FLAG	Status of spare-bit within the command-word	string		ON OFF	INSTRUMENT COMMAND (Class)	IDPH:Command Word(Bit 7)
INSTRUMENT_TELEMETRY	Instrument Telemetry 1 – +5 Volt Secondary Power 2 – -5 Volt Secondary Power 3 – +15 Volt Secondary Power 4 – -15 Volt Secondary Power 5 – +10 Volt Control Power 6 – -10 Volt Control Power 7 – +12 Volt Postamp Power 8 – -12 Volt Postamp Power 9 – Detector Temperature 10 – Motor Temperature 11 – Optics Temperature 12 – Laser Diode 1 Temperature 13 – Laser Diode 2 Temperature 14 – Fringe Peak Amplitude  <b>Note:</b> 1..8,10..14 apply for SPT, RAW, RWR and RICE ext-LONG type data only; 9 applies to all input data types.	real array[14]	Volts[8] Kelvin[5] Volts[1]		TABLE (Class)	Raw data – for RAW input data  Record header – for all other types of input data
INSTRUMENT_TYPE	Identifies the type of an instrument.  Example values: POLARIMETER, RADIOMETER, REFLECTANCE SPECTROMETER, VIDICON CAMERA.  Note that mosaics may contain more than one value in an array.	string		"THERMAL INFRARED SPECTROMETER"	IDENTIFICATION (Class)	Static Value
INTERCHANGE_FORMAT	The interchange_format element represents the manner in which data items are stored. Example values: BINARY, ASCII.	string		ASCII BINARY	HISTORY (Class) TABLE (Class)	Static Value
INVERTED_SPECTRA_FLAG	Source data quality flag. Applicable to higher level data products. See "The Mini-TES Data Processing Guide" for details.	Boolean (represented as integer)		0 – not-inverted 1 – inverted	SPECTRAL_QUEBE (Class)	Byproduct of calibration

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
LABEL_RECORDS	<p>Indicates the number of physical file records that contain only (PDS) label information.</p> <p>Note: The use of label_records along with other file-related data elements is fully described in the Standards Reference.</p>	integer		1 or more	FILE DATA ELEMENT (Class)	Calculated according to the size of the PDS label of the file.
LOCAL_TRUE_SOLAR_TIME	<p>Local true solar time, or LTST, is one of two types of solar time used to express the time of day at a point on the surface of a planetary body. LTST is measured relative to the true position of the Sun as seen from a point on the planet's surface.</p> <p>The coordinate system used to define LTST has its origin at the center of the planet. Its Z-axis is the north pole vector (or spin axis) of the planet. The X-axis is chosen to point in the direction of the vernal equinox of the planet's orbit. (The vernal or autumnal equinox vectors are found by searching the planetary ephemeris for those times when the vector from the planet's center to the Sun is perpendicular to the planet's north pole vector. The vernal equinox is the time when the Sun appears to rise above the planet's equator.)</p> <p>Positions of points in this frame can be expressed as a radius and areocentric 'right ascension' and 'declination' angles. The areocentric right ascension angle, or ARA, is measured positive eastward in the equatorial plane from the vernal equinox vector to the intersection of the meridian containing the point with the equator. Similarly, the areocentric declination is the angle between the equatorial plane and the vector to the point. LTST is a function of the difference between the ARAs of the vectors to the Sun and to the point on the planet's surface. Specifically,</p> $LTST = (a(P) - a(TS)) * (24 / 360) + 12$ <p>where, LTST = the local true solar time in true solar hours</p>	<p>string (in IDENTIFICATION)</p> <p>real (in TABLE and SPECTRAL_CUBE)</p>	Hours (in TABLE and SPECTRAL_CUBE)	hh:mm:ss (in IDENTIFICATION)	IDENTIFICATION (Class) TABLE (Class) SPECTRAL_CUBE (Class)	Calculated: ET, SPICE: SCLK Kernel, Landing-Site Kernel, P Kernel

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
	<p>a(P) = ARA of the point on the planet's surface in deg                      a(TS) = ARA of the true sun in deg</p> <p>The conversion factor of 24/360 is applied to transform the angular measure in decimal degrees into hours-minutes-seconds of arc. This standard representation divides 360 degrees into 24 hours, each hour into 60 minutes, and each minute into 60 seconds of arc. The hours, minutes, and seconds of arc are called 'true solar' hours, minutes, and seconds when used to measure LTST. The constant offset of 12 hours is added to the difference in ARAs to place local noon (12:00:00 in hours, minutes, seconds) at the point where the Sun is directly overhead; at this time, the ARA of the true sun is the same as that of the surface point so that a(P) - a(TS) = 0.</p> <p>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. See also LOCAL_MEAN_SOLAR_TIME.</p> <p>(Definition adapted from [VAUGHAN1995].)</p> <p>This element replaces the older MPF_LOCAL_TIME, which should no longer be used.</p>					
MIRROR_TEMP	See EXTERNAL_TEMPERATURES				SPECTRAL_QUBE (Class)	Record header.
MISSING_CAL_FLAG	Signifies the fact that an internal calibration observation bracketing this observation is missing. This flag is applicable to higher level data products only. See "The Mini-TES Data Processing Guide" for details.	Boolean (represented as integer)		0 – not-missing 1 – missing	SPECTRAL_QUBE (Class)	Byproduct of calibration

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
MISSION_NAME	Identifies a major planetary mission or project. A given planetary mission may be associated with one or more spacecraft.  Note that mosaics may contain more than one value in an array.	string		"MARS EXPLORATION ROVER"	IDENTIFICATION (Class)	Static Value
MISSION_PHASE_NAME	Provides the commonly-used identifier of a mission phase.	string(30)		"CRUISE", "EXTENDED MISSION", "PRIMARY MISSION", "ATLO", "ORT1", "ORT2", "TBD"	IDENTIFICATION (Class)	Operator Supplied Parameter
NPTS	Actual number of interferogram points acquired for a data point. Due to the mechanics of the instrument, this value can be different for each pixel acquired. For RWR products, this is the sum of the zone widths. In other words, the actual number of interferogram points acquired is not available.	integer			TABLE (Class) SPECTRAL_CUBE (Class)	Input ".dat" file.
OBSERVATION_ID	Uniquely identifies a scientific observation within a data set. It is set via the data product context ID - which doesn't necessarily map to a specific object - it's just used to group various instrument data sets together via a common keyword.	integer			IDENTIFICATION (Class)	IPDH:contxt_id
ORIGIN_OFFSET_VECTOR	Specifies the offset from the reference coordinate system's origin to the origin of the coordinate system being defined by the enclosing COORDINATE_SYSTEM_STATE group. In other words, it is the location of the current system's origin as measured in the reference system.	real [3]			ROVER_COORDINATE_SYSTEM (Group)	IDPH:{rvr_pos1,rvr_pos2,rvr_pos3}

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
ORIGIN_ROTATION_QUATERNION	<p>Provides an array of four values that specifies the rotation of the coordinate system being defined by the enclosing COORDINATE_SYSTEM_STATE group, relative to the reference system.</p> <p>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 <math>s = \cos(\theta/2)</math> and <math>v(n) = \sin(\theta/2) * a(n)</math>. Theta is the angle of rotation and a is the (x,y,z) vector around which the rotation occurs.</p>	real [4]			ROVER_COORDINATE_SYSTEM (Group)	IDPH:{rvr_quat1,rvr_quat2,rvr_quat3,rvr_quat4}  In this value, the quaternion is laid out as follows: <(q1,q2,q3),q4> where q4 is the scalar. When stored in the header it is flipped to <q4,(q1,q2,q3)>.
PDS_VERSION_ID	<p>Represents the version number of the PDS standards document that is valid when a data product label is created. Values for the PDS_version_id are formed by appending the integer for the latest version number to the letters 'PDS'.</p> <p>Examples: PDS3, PDS4.</p>	string(6)		"PDS3"	PDS required	PDS
PHASE_INVERT_FLAG	Source data quality flag. Applicable to higher level data products. See "The Mini-TES Data Processing Guide" for details.	Boolean (represented as integer)		0 – phase not inverted 1 – phase inverted	SPECTRAL_CUBE (Class)	Byproduct of calibration
PLANET_DAY_NUMBER	Indicates the number of sidereal days (rotation of 360 degrees) elapsed since a reference day (e.g., the day on which a landing vehicle set down). Days are measured in rotations of the planet in question from the reference day (which is day zero).	integer			IDENTIFICATION (Class)	Calculated using misc.txt, SCLK, LSK, SPK.
POSITIVE_AZIMUTH_DIRECTION	Provides the direction in which azimuth is measured in positive degrees for an observer on the surface of a body. The azimuth is measured with respect to the elevational reference plane. A value of CW indicates that Azimuth is measured positively Clockwise, and CCW indicates that Azimuth increases positively Counter-clockwise.	string		"CLOCKWISE" "COUNTERCLOCKWISE"	ROVER_COORDINATE_SYSTEM (Group)	Static Value "CLOCKWISE"

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
POSITIVE_ELEVATION_DIRECTION	The positive_elevation_direction element provides the direction in which elevation is measured in positive degrees for an observer on the surface of a body. The elevation is measured with respect to the azimuthal reference plane. A value of UP or ZENITH 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 or NADIR 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" "DOWN" "ZENITH" "NADIR"	ROVER_COORDINATE_SYSTEM (Group)	Static Value "UP"
PRODUCER_INSTITUTION_NAME	Identifies a university, research center, NASA center or other institution associated with the production of a data set. This would generally be an institution associated with the element PRODUCER_FULL_NAME.	string(60)		"ARIZONA STATE UNIVERSITY"	IDENTIFICATION (Class)	Static Value
PRODUCT_CREATION_TIME	Defines the UTC system format time when a product was created. Formation rule: YYYY-MM-DDThh:mm:ss[.fff]Z	string			IDENTIFICATION (Class)	System Call
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 less the extension.	IDENTIFICATION (Class)	Calculated from the output file name
PRODUCT_TYPE	The PRODUCT_TYPE data element identifies the type or category of a data product within a data set.	string		"MINUTES_EDR" – Experimental Data Record  "MINUTES_RDR" – Reduced Data Record  "MINUTES_BTR" – Brightness Temperature Record  "MINUTES_EMR" – Emissivity Record	IDENTIFICATION (class)	Static Value

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
PRODUCT_VERSION_ID	<p>Identifies the version of an individual product within a data set.</p> <p>Example: 1.0, 2A, 1.2.3C.</p> <p>Note: This is not the same as the data set version that is an element of the DATA_SET_ID value.</p> <p>PRODUCT_VERSION_ID is intended for use within AMMOS to identify separate iterations of a given product, which may have a unique FILE_NAME.</p> <p>(see source for building rules)</p>	string(12)		"V1.0 D-22847"	IDENTIFICATION (Class)	<p>EDR: Built from SIS Version + SIS Document Number</p> <p>Version of higher level data products (RDR, BTR, EMR) defined by MINI-TES Team</p>
QUATERNION_MEASUREMENT_METHOD	<p>Specifies the quality of the rover orientation estimate.</p> <p>"UNKNOWN" - The attitude should simply not be trusted. This is the initial grade given on Landing, for example.</p> <p>"TILT_ONLY" - The attitude estimate is only good for tilt determination (2-axis knowledge). Activities which require azimuth knowledge should be careful.</p> <p>"COARSE" - The attitude estimate is "complete" (it has all three axes) but is crude. This can occur because a sun-gaze has not yet been performed or because some event (such as traverses or IDD activity) have reduced the quality of the estimate (a.k.a. "ThreeAxisCoarse").</p> <p>"FINE" - Sun-gaze completed successfully, and the attitude estimate is sufficient for pointing HGA (a.k.a. "ThreeAxisFine"). Rover quaternion quality flag.</p>	string		0 – UNKNOWN 1 – TILT_ONLY 2 – COARSE 3 – FINE	COORDINATE SYSTEM STATE: ROVER (Class)	IDPH:rvr_quat_qlty
RAW_RADIANCE	<p>Array of raw radiance values</p> <p>Note: Applies to RCP data products only.</p>	real array	w/m <sup>2</sup>		TABLE (Class)	Record data
RECEIVED_PACKETS	<p>Count of the packets actually received as compared to what was expected (see EXPECTED_PACKETS)</p>	integer		0 or more	TELEMETRY (Class)	Calculated by counting EMD:product_metadata:part_list:part

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
RECORD_BYTES	<p>Indicates the number of bytes in a physical file record, including record terminators and separators.</p> <p>Note: In the PDS, the use of record_bytes, along with other file-related data elements is fully described in the Standards Reference.</p>	integer		0 or more	FILE DATA ELEMENT(Class)	Calculated
RECORD_TYPE	<p>Indicates the record format of a file.</p> <p>Note: In the PDS, when record_type is used in a detached label file it always describes its corresponding detached data file, not the label file itself. The use of record_type along with other file-related data elements is fully described in the PDS Standards Reference.</p>	string(20)		"FIXED_LENGTH"	FILE DATA ELEMENT(Class)	Static Value
REFERENCE_COORD_SYSTEM_INDEX	<p>Identifies which instance of the coordinate system named by REFERENCE_COORD_SYSTEM_NAME is the reference coordinate system for the group in which the keyword occurs. This index is a set of integers which serve to identify coordinate system instances in a mission-specific manner.</p> <p>For MER, the indices are based on the ROVER_MOTION_COUNTER. This counter is incremented each time the rover moves (or may potentially have moved, e.g. due to arm motion). The full counter may have up to 5 values (SITE, DRIVE, IDD, PMA, HGA), but normally only the first value (for SITE frames) or the 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.</p> <p>See also REFERENCE_COORD_SYSTEM_NAME and COORDINATE_SYSTEM_INDEX.</p>	integer array[5]			ROVER_COORDINATE_SYSTEM (Group)	Value of the SITE index from the COORDINATE_SYSTEM_INDEX

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
REFERENCE_COORD_SYSTEM_NAME	Provides the full name of the reference coordinate system for the group in which the keyword occurs. All vectors and positions relating to 3-D space within the enclosing group are expressed using this reference coordinate system. Non-unique coordinate systems (such as "SITE" for rover missions), which have multiple instances using the same name, also require REFERENCE_COORD_SYSTEM_INDEX to completely identify the reference coordinate system.  For MER, the reference is usually a SITE frame.	string(20)		"ROVER_FRAME", "SITE_FRAME", "MAST_FRAME", "LOCAL_LEVEL_FRAME";	ROVER_COORDINATE_SYSTEM (Group)	Fixed value: SITE
RELEASE_ID	A unique identifier associated with the specific release of the dataset. The first release is given a RELEASE_ID value of "0001" and incremented for each subsequent release.	string(4)		0001 – Aug 2004 0002 – Oct 2004 ...	IDENTIFICATION (Class)	Set by MER Project; corresponds with release dates described in the MER Archive Plan (JPL D-19658)
RINGING_FLAG	Source data quality flag. Applicable to higher level data products. See "The Mini-TES Data Processing Guide" for details.	Boolean (represented as integer)		0 – not-ringing 1 – ringing	SPECTRAL_CUBE (Class)	Byproduct of calibration
RINGING_AMPLITUDE	Source data quality value. Applicable to higher level data products. See "The Mini-TES Data Processing Guide" for details.	real			SPECTRAL_CUBE (Class)	Byproduct of calibration

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
ROVER_MOTION_COUNTER	<p>A set of integers which describe a (potentially) unique location (position/orientation) for a rover. Each time something happens that moves, or could potentially move, the rover, a new motion counter value is created. This includes intentional motion due to drive commands, as well as potential motion due to other articulating devices, such as arms or antennae. This motion counter (or part of it) is used as a reference to define instances of coordinate systems which can move such as SITE or ROVER frames. The motion counter is defined in a mission-specific manner. Although the original intent was to have incrementing indices (e.g. MER), the motion counter could also contain any integer values which conform to the above definition, such as time or spacecraft clock values.</p> <p>For MER, the motion counter consists of five values. In order, they are Site, Drive, IDD, PMA, and HGA. The Site value increments whenever a new major Site frame is declared. The Drive value increments any time intentional driving is done. Each of those resets all later indices to 0 when they increment. The IDD, PMA, and HGA increment whenever the corresponding articulation device moves. It is TBD whether IDD, PMA, and HGA are independent of each other, or reset the others to 0 in a hierarchical manner when they are incremented.</p> <p>Conceptually, a sixth value could be added by ground processing to indicate unintentional slippage (e.g. the wind blew the rover off a rock). This sixth value will never occur in telemetry but might occur in certain RDR's.</p>	integer array[5]			IDENTIFICATION (Class)	IDPH:rvr_site_idx, IDPH:rvr_move_idx, IDPH:rvr_idd_idx, IDPH:rvr_pma_idx, IDPH:rvr_hga_idx  <b>Note:</b> move is also referred to as drive or position index.
ROVER_MOTION_COUNTER_NAME	An array that provides the formal names identifying each integer in ROVER_MOTION_COUNTER.	string array[5]		("SITE", "DRIVE", "IDD", "PMA", "HGA")	IDENTIFICATION (Class)	Static Value
SEQUENCE_ID	Provides an identification of the spacecraft sequence associated with the given product. This element replaces the older seq_id, which should no longer be used.	string(30)			IDENTIFICATION (Class)	EMD:product_metadat a:seq_letter + EMD:product_metadat a:seq_number

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
SEQUENCE_VERSION_ID	Provides the version identifier for a particular observation sequence used during planning or data processing.	string(30)			IDENTIFICATION (Class)	EMD:product_metadata:seq_version
SOLAR_LONGITUDE		real			IDENTIFICATION (Class)	Calculated: misc.txt, SCLK, LSK, SPK
SPACECRAFT_CLOCK_CNT_PARTITION	Indicates the clock partition active for the SPACECRAFT_CLOCK_START_COUNT and SPACECRAFT_CLOCK_STOP_COUNT elements.	integer		1	IDENTIFICATION (Class)	Static Value
SPACECRAFT_CLOCK_START_COUNT	Provides the value of the spacecraft clock at the beginning of a time period of interest.  Format is dddddd.ddd measured in units of seconds and stored internally as a floatin point number  Note: In the PDS, sclk_start_counts have been represented in the following ways: Voyager - Flight Data Subsystem (FDS) clock count (floating point 7.2) Mariner 9 - Data Automation Subsystem, Mariner 10 - FDS - spacecraft_clock	string(30)			IDENTIFICATION (Class)	Calculated from IDPH:bgn_time
SPACECRAFT_CLOCK_STOP_COUNT	Provides the value of the spacecraft clock at the end of a time period of interest.  Format is dddddd.ddd measured in units of seconds and stored internally as a floatin point number  Note that this value is estimated from downlinked data as it is not available directly.	string(30)			IDENTIFICATION (Class)	Calculated  SPACECRAFT_CLOCK_START_COUNT + 2 * (#ICKs downlinked)
SPEC_EXP	When the EDR data is downlinked as spectra (RCP), each spectra is downloaded as shorts with a 2^SPEC_EXP multiplier taken out of it. SPEC_EXP represents this multiplier, the base for which is already known, i.e. 2.0.	integer			TABLE (Class) SPECTRAL_CUBE (Class)	One value per downloaded spectra from the input ".dat" file.
START_TIME	Provides the date and time of the beginning of an event or observation (whether it be a spacecraft, ground-based, or system event) in UTC system format. Formation rule: "YYYY-MM-DDThh:mm:ss[.fff]"	string			IDENTIFICATION (Class)	Calculated from SPACECRAFT_CLOCK_START_COUNT using SCLK
STOP_TIME	Provides the date and time of the beginning of an event or observation (whether it be a spacecraft, ground-based, or system event) in UTC system format. Formation rule: "YYYY-MM-DDThh:mm:ss[.fff]"	string			IDENTIFICATION (Class)	Calculated from SPACECRAFT_CLOCK_STOP_COUNT using SCLK

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
TARGET_NAME	Identifies a target. The target may be a planet, satellite, ring, region, feature, asteroid or comet. See TARGET_TYPE.	string(30)		"MARS", "CALIBRATION"	IDENTIFICATION (Class)	User Parameter
TARGET_TYPE	Identifies the type of a named target.	string		"CALIBRATION", "DUST", "N/A", "SUN", "PLANET"	IDENTIFICATION (Class)	User Parameter
TELEMETRY_FORMAT_ID	Telemetry format code	string(3)		113="RAW" - Interferogram  114="SPT" - Complex Spectrum  115="RCP" - RICE_compressed Spectrum  116="RWR" - RICE compressed interferogram	TELEMETRY (Class)	IDPH:flavor
TELEMETRY_FMT_EXTENSION_TYPE	Telemetry format extension provides additional information about what kind of telemetry is collected during scene looks versus calibration looks.	string(5)		LONG (Scene: Long Telemetry; Cal: Long Telemetry) SHORT (Scene: Short Telemetry; Cal: Short Telemetry) MIXED (Scene: Short Telemetry; Cal: Long Telemetry)	TELEMETRY (Class)	Calculated from IDPH:flavor
TELEMETRY_PROVIDER_ID	Identifies the provider and version of the telemetry data used in the generation of this data.	string		"SSW MER_DP V1.0"	TELEMETRY (Class)	Constant
TELEMETRY_SOURCE_NAME	Name of the source of the telemetry used in create of this data set.	string			TELEMETRY (Class)	Name of the input data product.
TELEMETRY_SOURCE_TYPE	Classifies the source of the telemetry used in creation of this data set.	string(12)		"SFDU" "DATA PRODUCT"	TELEMETRY (Class)	Constant (since only one value applies)
TLM_INST_DATA_HEADER_ID	Indicates the version of the instrument specific information provided with telemetry data products. Incremented by FSW whenever there is a change to the header structure.	integer			TELEMETRY (Class)	IDPH:version
TLM1 TLM2 : TLM14	See INSTRUMENT_TELEMETRY				SPECTRAL_QUBE (Class)	Record header.

Keyword Name	Definition	Type	Units	Valid Values	Location in PDS Label	Source
ZONE1_WIDTH	<p>Width of zone-1 in the RICE compressed interferogram data.</p> <p>This applies to RWR data products only in which the raw-interferograms are compressed using the RICE compression algorithm and down-linked without converting them to spectra.</p> <p>Before compressing the data, the interferogram is divided into three zones. Zone-2 contains the center-burst (or the middle part of the interferogram). It is a commandable parameter. Zone-1 starts immediately to the left of zone-2 (interferogram acquired left to right) and extends far enough to grab either a maximum of 512 points or until the 21st point from the start of the interferogram is hit. The first 20 points of the interferogram are tossed away in this type of a data product. Zone-3 starts immediately after zone-2 and extends upto a maximum of 512 points or until the 21<sup>st</sup> last point is hit.</p> <p><b>Note:</b> Applies to RWR data only</p>	integer			SPECTRAL_QUBE (Class)	Decompressed Data Record Header
ZONE3_WIDTH	See ZONE1_WIDTH	integer			SPECTRAL_QUBE (Class)	Decompressed Data Record Header
ZPD	<p>Zero path difference column number. This number gives the index around which the interferogram was folded before converting it into spectra (by passing it through FFT).</p> <p><b>Note:</b> Applies to SPT &amp; RCP EDRs only in which case this value is equal to ZPD_MINMAX.</p>	integer			TABLE (Class) SPECTRAL_QUBE (Class)	Record Data
ZPD_MINMAX	<p>Estimated zero path difference column number or the estimated center-burst column number within the interferogram acquired on board. This number assumes no trimming of interferogram. This value is filled using the flight version of the zpd calculation algorithm.</p> <p><b>Note:</b> Applies to RWR, SPT &amp; RCP data only</p>	integer			TABLE (Class) SPECTRAL_QUBE (Class)	Record Data