Phoenix (PHX) Project

MET Pressure and Temperature
EDR and RDR

Software Interface Specification (SIS)

Version 1.5

JPL PHX-274-313
D-33236

Prepared by: ________________________________ **Cameron Dickinson** Phoenix MET Team	________________________________
Approved by: ________________________________ **Jim Whiteway** Instrument Co-Investigator, MET	________________________________ **Leslie Tamppari** Phoenix Project Scientist
________________________________ **Dr. Reta Beebe** Director, PDS Atmospheres Node	________________________________ **Ed Grayzeck** Project Manager, Planetary Data System

Revision and History Page

Rev.	Description	Date
0.02	Document Start	
0.5	Updating file naming and label formats	
0.9	Updated Flow descriptions and process	
0.9*	Post Review	05/14/07
1.0	PDS Labels	13/12/07
1.1	Updates incorporated	17/04/08
1.2	PDS labels	26/05/08
1.3	Inclusion of Pressure correction products	20/11/08

TBD Items

Section	Description

Acronyms, Abbreviations, and Definitions

CCSDS	Consultative Committee for Space Data Systems
CSA	Canadian Space Agency
DN	Digital Number
EDR	Elementary Data Record
EMH	EDR MET High Resolution Data
EML	EDR MET Low Resolution Data
FPGA	Field Programmable Gate Array
GDS	Ground Data Segment
MDA	MacDonald, Dettwiler and Associates Ltd.
MET	Meteorological Team
MPTMS	MET Pressure/Temperature Manager Software. The software in the PEB that records the data from the sensors and transmits it to the Lander
PEB	Payload Electronics Box.
PRT	Platinum Resistance Thermometer
PSI	Phoenix Science Interface Software Tool
PT	Pressure and Temperature
RDR	Reduced Data Record
RMH	RDR MET High Resolution Data
RML	RDR MET Low Resolution Data
wrt	With Respect To

1. Purpose and Scope of Document

This document provides users of the Phoenix MET (Meteorological) Pressure and Temperature data product with a detailed description of the product and a description of how it was generated, including data sources and destinations.

It is intended to provide enough information to enable users to read and understand the data product. The users for whom this document is intended are the scientists who will analyze the data, including those associated with the project and those in the general planetary science community.

The resulting data products consist of Temperature data in Kelvin at three elevations, and Barometric Pressure in Pascals measured at a single location. Data products may appear as either low resolution data (averaged every 512 seconds) or high resolution data (measurements taken every two seconds).

This Data Product SIS describes how the EDR is acquired by the MET-PT instrument, and how it is processed, formatted, labeled, and uniquely identified. The document discusses standards used in generating the product and software that may be used to access the product. The data product structure and organization is described in sufficient detail to enable a user to read the product. Finally, an example of a product label is provided.

2. Applicable Documents

[1] MET Mission Requirements Document

[2] MET Data Analysis Plan (DAP) V1.0

[3] MET Pressure/Temperature Manager Software (MPTMS) System Requirements Specification, Document # MDR-PHX-SG-7927

[4] MET Ground Station Software ICD, Document# MDR-PHX-ICD-7882

[5] Planetary Data System Standards Reference, Document# JPL D-7669 V3.7

[6] Planetary Science Data Dictionary Document, Document# JPL D-7116, Rev. E

[7] Instrument Paper – Taylor*, et al.,* Temperature, Pressure and Wind instrumentation on Phoenix MET *J. Geophys. Res*. (2008) E00A10, doi:10.1029/2007JE003015

[8] MET-P&T Science Team and PDS Atmospheres Node ICD, V1.1

[9] MET PT CCC Report

[10] MET PT Archive Volume SIS

3. Relationships with Other Interfaces

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

- MET Ground Segment software
- MET Ground Station Software ICD [4]
- MET Data Analysis Plan [2]

4. Data Product Characteristics and Environment

4.1 Instrument Overview

The Meteorology Package (MET) is a scientific payload of the Phoenix Mars Scout Lander. The MET will be capable of performing the following functions:

- Detecting and ranging scatterers in the Martian atmosphere vertically with respect to the lander deck.

- Measuring the barometric pressure of the Martian atmosphere at the Phoenix landing site.

- Measuring the atmospheric temperature of the Martian atmosphere at three elevations at the Phoenix landing site.

This document provides users of the MET-Pressure and Temperature (PT) data products with a detailed description of the product and a description of how it was generated, including data sources and destinations. The data products for the MET Lidar are described in a separate SIS.

4.1.1 Pressure/Temperature Operation Summary

Pressure and Temperature measurements will be taken concurrently, and the instrument can be operated overnight even while the Lander computer is off.

The PT instrument can be in one of the following mutually exclusive operating states, with respect to data collection:

- Off - The instrument is not powered.

- Idle - The instrument is powered, but no data is collected.

- Record - Pressure and temperature data is recorded, compressed, and stored in flash memory.

- Transmit Stored Data - The previously stored data is transmitted to the Lander computer. No data is collected during this time.

- Transmit Cyclic - Pressure and temperature data is immediately transmitted to the Lander computer.

For science data collection, the PT instrument may be set into one of two recording modes: Averaging or Full Rate. In Full Rate mode, the data is recorded or transmitted every two seconds. In Averaging mode, data is collected into the instrument's RAM every two seconds. After 256 samples (256 frame counts), the average, standard deviation, minimum, and maximum of all the samples are recorded or transmitted unless an "event" is detected (see Section 4.1.4, below).

4.1.2 Pressure Sensor

The MET pressure sensor measures Barometric pressure over the range 500 to 1200 Pascals. This subsystem consists of a unit mounted on the Lander's upper Payload Electronics Box (PEB) and electronics and for receiving, processing, and storing the sensor data, mounted inside the PEB. The pressure sensor contains three independent capacitive sensing elements within the unit, of which one is selected as the nominal sensing element for calculating barometric pressure.

The Phoenix pressure sensor unit comprises eight (8) sensor channels of four (4) types of capacitive components: pressure, temperature and reference sensor heads as well as a constant capacitance. The pressure transducers are a variant of the Silicon capacitive absolute pressure sensor head (*Barocap*) produced by Vaisala, Inc. The varying ambient pressure bends a thin Silicon diaphragm inducing changes in the sensor head capacitance. Three pressure heads are mounted on the device: one LL-type (high-resolution) and two RSP1-type (high-stability) *Barocap*s. Pressure sensor unit housekeeping temperature is measured by two 2 Vaisala capacitive *Thermocap* temperature sensor heads. The constant sensor head is a high-stability capacitor (calibrated during the pressure sensor unit calibration). It provides a fixed capacitance and is mainly used to check the stability and performance of the whole measurement system. The reference sensor heads are high-stability capacitors and provide also fixed capacitances. The pressure sensor unit includes two 2 references. They are used in the pressure measurement in combination with the *Thermocap* and *Barocap* sensor heads to enable compression of the pressure data.

The average, standard deviation, minimum and maximum values will be calculated once every 256 frames (512 seconds), with processing being performed onboard using algorithms proprietary to Vaisala Inc. The quantity stored on board the MET-PT instrument will be in SI units (Pa, Pa^2). Should the difference between minimum and maximum values within a 512 second period exceed a pre-set threshold (trigger) for either the Pressure or Temperature sensors, the raw data will be stored instead as High

Resolution data (along with the time of the event), otherwise the HR data will be purged from memory.

4.1.3 Temperature Sensors

The MET Temperature sensors measure absolute Temperature over the range 140 to 280 Kelvin. This subsystem consists of three C-frame thermocouples mounted on the MET Mast at 243.2, 493.2 and 993.2mm above the lander deck (herein referenced as 250, 500 and 1000mm, respectively), and a Platinum Resistive Thermometer (PRT) at the base of the Mast. The thermocouple wires terminate at the mast Isothermal Block (IB). For each thermocouple, a pair of Teflon Insulated copper extension wires takes the signal from Isothermal Block to The Payload Electronics Box (PEB). The temperature gradient between the Isothermal Block and the PEB will create an EMF on the copper wires, since they are not ideally homogeneous.

The data acquired by the temperature sensor (T_{ref}, T_{250}, T_{500} and T_{1000} in units of digital numbers, DN) will be used to calculate the average, standard deviation, minimum and maximum values once every 256 frames (512 seconds). This data will be stored as the Low Resolution unless the difference between minimum and maximum values exceeds a pre-set threshold (trigger) for either the Pressure or Temperature sensors. In this case, the raw data will be recorded as High Resolution 2 sec data.

The PRT voltage is calibrated onboard the MET-PT instrument, and then employed to calculate the relative voltage of each thermocouple at the isothermal reference junction. This value is than converted to temperature in units of Digital Number (DN) and stored onboard the MET-PT instrument for later transmission. 1 DN = 0.01K.

4.1.4 Events

When in the Averaging mode, the MET-PT software will examine the data received over the last 256-sample and look for events in the data. If an event is found in either of the pressure or temperature data, all 256 data samples from that block for both pressure and temperature are recorded.

Events are defined as:

- A temperature event is defined as a change of 15 Kelvin over 512 seconds, triggered at any of the three mast levels.
- A pressure event is defined as a 1 Pascal pressure difference over 512 seconds.

Both thresholds are configurable; the values given are the defaults. These values will likely be adjusted to fit within an allotted operational data volume envelope.

4.1.5 Known Issues and Idiosyncrasies

- The pressure sensor infrequently reports pressure values 100-10000 times larger than expected when in averaging mode, thus causing the data to be recorded as an event. The high resolution 2 second data that is saved will not be affected.
- Pressure sensor data was affected by the placement of a keep alive heater relative to the pressure sensor.

4.2 Data Product Overview

The Phoenix Mission uses CODMAC data processing level descriptions.

4.2.1 Level 2 EDRs

These EDRs are essentially identical to the telemetry messages sent from the MET-PT instrument to the Lander computer as defined in [1], converted to ASCII and with the addition of a Lander timestamp (see section 4.4.3 for timestamp details.) and commanded parameters (such as threshold values).

Data that had been compressed in the Record state is decompressed.

4.2.2 Level 3 RDRs

Level 3 Data Products (RDRs) have had the following changes as compared to the Level 2 EDRs:

1. The data has been changed and reordered where required to be consistent with [2].

2. The Frame Count is converted to the duration of each measurement in Earth seconds.

3. Temperature Digital Numbers have been converted to degrees Kelvin.

A second Level 3 Data Product (RDR) has been created, formed from the Level 3 product above and ancillary, housekeeping data. This product has corrected the pressure values for effects of a keep alive heater places in close proximity to the pressure sensor unit. This was discovered upon the surface by the duty cycle oscillations of the heater, as observed in the data. This correction is a first attempt to retrieve pressure values more accurately, but is included in the initial data release for users of the data.

4.3 Data Processing

This section provides general information about the processing of MET data sets. Details specific to each data set are found in Section 5

4.3.1 Data Processing Level

Table 1 defines the data products in terms of their associated CODMAC processing levels.

Table 1 - Pressure/Temperature Data Products

CODMAC Processing Level	Description	Standard or Special	Pressure/Temperature Data Products		Responsibility
Raw – Level 1	Telemetry data stream as received at the ground station, with science and engineering data embedded.	n/a	SFDU (Standard Formatted Data Unit) packets of compressed binary MET pressure/temperature data.		TDS (JPL)
Edited Level 2 **(EDR)**	Instrument science data (e.g., raw voltages, counts) at full resolution, time ordered, with duplicates and transmission errors removed.	Standard	**MET Low-Resolution Pressure/Temperature Level 2 EDR (EML)** • Uncompressed MET pressure/temperature data consisting of a time-tagged sequence of PTAvgCyclicMsg telemetry messages as defined in the Software ICD (MDR-PHX-ICD-7781). • PDS-compatible ASCII format	**MET High-Resolution Pressure/Temperature Level 2 EDR (EMH)** • Uncompressed MET pressure/temperature data consisting of a time-tagged sequence of PTEventMsg and PTFullCyclicMsg telemetry messages as defined in the Software ICD (MDR-PHX-ICD-7781). • PDS-compatible ASCII format	MET Ground Segment (CSA)
Calibrated Level 3 **(RDR)**	Level 2 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).	Standard	**Meteorology Pressure/Temperature Low-Resolution (RML)** • Equation processed (if required) MET low-resolution pressure/temperature data derived from the Level 2 EDR. • PDS compatible ASCII format as defined in the Phoenix MET P&T SIS.	**Meteorology Pressure/Temperature High-Resolution (RMH)** • Equation processed (if required) MET low-resolution pressure/temperature data derived from the Level 2 EDR. • PDS compatible ASCII format as defined in the Phoenix MET P&T SIS.	MET Ground Segment (CSA)
		Special	**Meteorology Pressure-Corrected / Temperature High-Resolution (RMC)** • Data corrected for effects of heater placement.	**MET Ancillary Pressure Sensor Temperature Data (RMA)** • Internal Pressure Sensor Temperatures employed in Pressure correction	

4.3.2 Data Product Generation

MET data products will be generated by the MET Team led by Co-Investigator Whiteway at York University.

The majority of the data processing for the MET-PT instrument is performed in the flight segment, either by FPGA hardware or MPTMS software within the PEB, as specified in [3]. Under command from the MET Device Manager software (written by JPL) on the Lander computer, the MET-PT takes measurements and nominally saves the data to instrument internal Flash memory (Record state). The system may also be configured to transmit the data immediately to the Lander instead, if required (Transmit Cyclic state).

In the nominal case where data is recorded to flash, it is first compressed. In either case, data is ultimately transmitted from the MET-PT to the Lander computer using the format described in [1]. The messages are replicated exactly as sent, with the addition of CCSDS header information applied by the MET Device Manager on the Lander. Included in this CCSDS information is a timestamp representing the time that data collection started and when collection was stopped.

The data is then stored on the Lander and held for periodic download. The stored telemetry data are downloaded from the Lander for relay to the Deep Space Network (DSN). Data received from the DSN are inserted into the Jet Propulsion Laboratory's (JPL) Telemetry Data System (TDS). The University of Arizona (UA) queries the TDS for the most recent telemetry dataset. The dataset is output to a spooler that passes data to the UA. Raw telemetry data are received by the UA, and a number of automated computer processes are run to place data in the MET directory of the UA Science LAN in an ASCII formatted file described in [4].

The MET-GDS software prepends the Timestamp, and writes the data out to files in the format given in Section 5. These are the Level 2 EDRs.

4.3.3 Data Flow

The following diagram (Fig 1) illustrates the data flow:

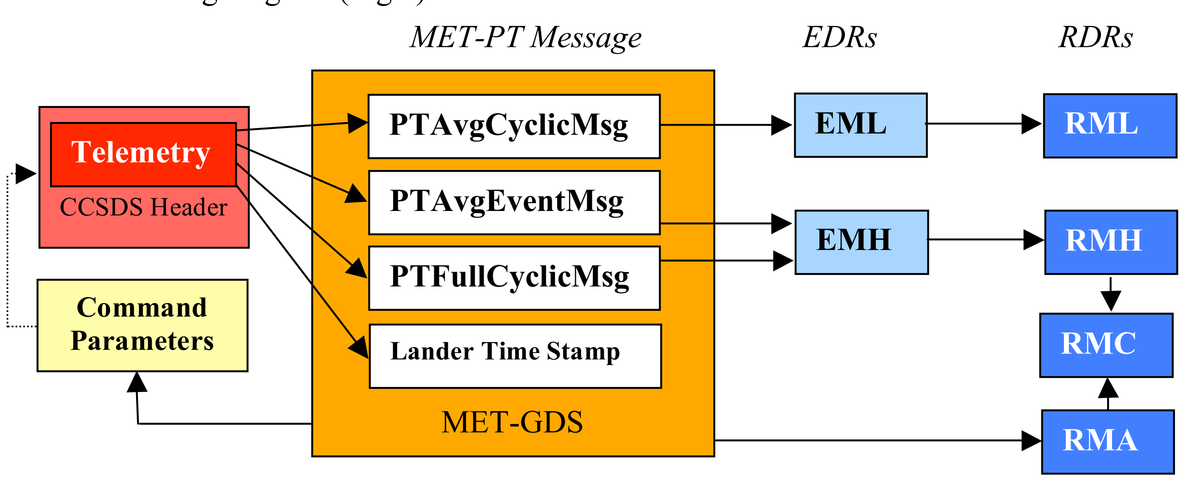


Table 2 MET Standard Data Products

Product Type	Time span covered	Generation Interval	Expected Size of Product	Total Data Volume for Primary Mission.
EML	1 Sol	90 sols	500 kB	45 MB
EMH	1 Sol	90 sols	3.5 MB	315 MB
RML	1 Sol	90 sols	500 kB	45 MB
RMH	1 Sol	90 sols	3.5 MB	315 MB
RMC	1 Sol	90 sols	3.5 MB	315 MB
RMA	1 Sol	90 sols	10 kB	1 MB

4.4 Standards Used in Generating Data Products

4.4.1 PDS Standards

All EDR and RDR products will comply with the Planetary Data System's standards as specified in the PDS Standards Reference [5]. All label keywords are PDS compliant and registered in the PDS dictionary.

4.4.2 File Naming

The EDR and RDR formats described in this document follow the Phoenix product filenaming conventions as described in Appendix D. Filenames, by definition, will be PDS compliant. Additional identification information will be contained in the PDS label as described in Appendix A.

4.4.3 Time Standards

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

SCET	Spacecraft event time. The time when an even occurred on-board, in GMT
SCLK	Spacecraft Clock, an on-board counter which increments once every 100 milliseconds, with origin, set to zero, at midnight on 1-Jan-1980.
Local True Solar Time (LTST)	Local True Solar Time; LTST is the local solar time expressed by the number of local solar days (SOLs) from a landing date and using a "24-hour" clock readout within the current local solar day (HR:MN:SC); LTST is a true local solar time and computed using MARS24 : SOL 12 12:00:00

Local Mean Solar Time (LMST)	Local Mean Solar Time; LMST is the mean local solar time expressed by the number of local solar days (SOLs) from a landing date and using a "24-hour" clock readout within the current local solar day (HR:MN:SC); LMST is an average local solar time and computed using MARS24 : SOL 12 12:00:01
Coordinated Universal Time (UTC)	Coordinated Universal Time: yyyy-mm-ddThh:mm:ss.sss.

The MET device returns a message containing the PT Frame Count (FRAME_COUNT) for every command sent by the Lander to the MET PT device. The Frame count is a running integer count since MET-PT power up, with each frame = 2.0 seconds. A CCSDS header, containing the Lander timestamp (UTC), is attached to each such message.

The PT data collection start time is calculated simply as:

$START\_TIME_{UTC} = UTC + 2sec * FRAME\_COUNT^{*}$
or
$START\_TIME_{LTST,LMST} = [UTC + 2sec * FRAME\_COUNT^{*}]_{LTST,LMST}$,

with LTST and LMST indicating a time conversion Mars Local Solar Time (using MARS24, see Appendix F), and FRAME_COUNT* is the Frame Count at the start of data acquisition. LTST uses a Mars Longitude of 125.75W (actual, landed longitude) while LMST uses 126.65W (mission epoch, expected landing site longitude).

The time at which each profile was collected is given in the .TAB files as DURATION, which is the number of seconds since START_TIME for each profile, with the interval between profiles being the integration period (PERIOD_DURATION). The UTC time of each profile is thus simply:

Collection Time of each measurement = START_TIME + DURATION,

with the caveat that DURATION will be provided in Earth seconds.

The MET-PT integration period, (nominally 512 or 2 seconds, for averaging and full rate data, respectively) refers to the time at the end of the averaging block in the Level 2 EDRs and Level 3 RDRs.

4.4.4 Data Storage Conventions

See PT Archive Volume SIS [10]

4.5 Data Validation

The MET EDR and RDR data product design, as described in this SIS, is subject to PDS peer review. The peer review will be done well in advance of actual production, to allow time for changes in the design as needed. This SIS document will be updated to show any such changes.

Validation of MET EDR and RDR products during production will be done according to specifications in the Phoenix Archive Plan and the MET Team – PDS Atmospherics Node ICD [8]. The MET Team will validate the science content of the data products, and the Atmospheres Node will validate the products for compliance with PDS standards and for conformance with the design specified in this SIS.

5. Detailed Data Product Specifications

5.1 Overview

There are two types of data produced by the MET-PT instrument. The first is low resolution data consisting of data collected at 0.5 Hz being averaged over 256 samples. Therefore the average, standard deviation, maximum, and minimum of the pressure sensor and each of three temperature sensors (and PRT block) are produced every 512 seconds. The data contained in the EDR data products of this type (EML) are generally in engineering type units such as DN, while the higher level RDR data product (RML) contains units of time and Kelvin.

The second type of data is high-resolution data, produced when the MET-PT has been instructed to record all data collected at 0.5 Hz or when an event (see Section 4.1.4) is found while in the averaging mode. Similarly, the data contained in this EDR data product (EMH) are generally in engineering type units such as DN, while the higher level RDR data product (RMH) contains units of time and Kelvin.

5.1.1 Label and Header Common characteristics

PDS header label file (.LBL) will be human-readable ASCII files containing Identification Data Elements, History Data Elements and Commanded Parameters. Descriptions of the label elements can be found in Appendix A, a sample .LBL file can be found in Appendix B.

5.1.2 Data File Common Characteristics

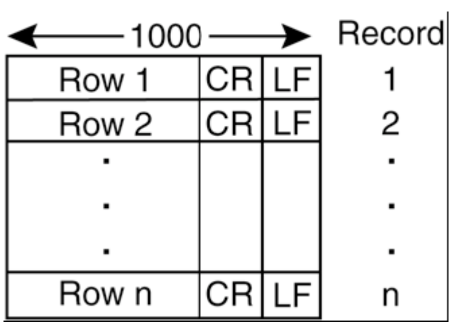
The data files will be human-readable ASCII files (.TAB), delimited by commas (ASCII code 44) stored in columns of fixed-width records with a (<CR><LF>) delimiters at the end of each row. Descriptions of the label elements can be found in Appendix C.

The following label fragment illustrates the general characteristics of the recommended ASCII format for a table with 1000-byte records:

(See example label in Appendix A)

```
RECORD_TYPE = FIXED_LENGTH
RECORD_BYTES = 1000
…
OBJECT = TABLE
INTERCHANGE_FORMAT = ASCII
ROW_BYTES = 1000
...
END_OBJECT = TABLE
```

Corresponding .TAB file:



5.2 MET Low-Resolution Pressure/Temperature Level 2 EDR (EML)

The EDR record contains telemetry of the average, standard deviation, minimum and maximum value of pressure, the three temperature values, (corresponding to the three thermocouple positions on the MET mast) and the reference temperature.

Telemetry of these entries are expected at sampling intervals of every 256 readings (512 seconds), including overnight, **except** under these circumstances:

1. The PT instrument was not commanded to record data during this time. For example, during transmission of stored telemetry, reception of software uploads or other operational reasons

2. An "event" was found in the data, in these cases a series of 256 High-Resolution Pressure/Temperature Level 2 EDR entries appears instead for this

time period (see Section 4.1.4). Low Resolution data will be recreated in the MET-GDS from High Resolution data.

3. The recorded data was not successfully transmitted back to Earth.

The record will also contain an ID describing if the threshold values triggered the storage of high resolution data. In such an event, High Resolution Data will be converted by the MET-GDS to Low Resolution equivalent data (i.e 512 second averaged data including min, max and standard deviation). High resolution data that is not evenly divisible by 512 seconds (i.e. any remaining measurements collected using the "Full Resolution" Mode) will not be included in the Low Resolution data product; however operationally an attempt will be made to collect Full Resolution data at integer values of 512 seconds.

This temporal values will be provided as a FRAME COUNT (see section 4.4.3 for details), or the number of 2 second frames since instrument power up occurring at the end of each averaging period.

Commanded parameters describing the unique MET telemetry ID; the collection mode (averaging); the specific pressure sensor; the temperature and pressure event threshold values; the duration of each acquisition period (512 seconds); and the number of such periods are also provided.

The Averaged data will be stored using the formats described in Appendix A and Appendix C, for label and data files, respectively. Each EDR is comprised or one record with tabulated data (.TAB) provided in the following columns:

"EML" – Low Resolution						
Temporal	*Pressure*	*Temp 250*	*Temp 500*	*Temp 1000*	*Temp Ref*	*Trigger*
Frame Count	Average / Standard Deviation / Minimum / Maximum					Sensor

5.3 MET High-Resolution Pressure/Temperature Level 2 EDR (EMH)

The EDR record contains telemetry of the pressure, three temperature differences (corresponding to the three thermocouple positions on the MET mast) and the reference temperature at 2 second intervals.

This EDR is acquired by one of two operational modes:

- **Averaging Mode:** entries occur when the PT instrument is otherwise recording averaged PT data, but an event (See Section 4.1.4) has been detected in the last 512-second interval. These entries do not appear at any fixed rate. This entry contains arrays of 256 of each of the parameters.

- **Full Rate Mode:** entries occur when the PT instrument has been specifically commanded to report full resolution data. These entries are expected with timestamps at intervals of every 2 seconds, operationally, an attempt will be made to collect data at an integer value of 512 seconds.

This temporal values will be provided as a FRAME COUNT (see section 4.4.3 for details), or the number of 2 second frames since instrument power up occurring at the end of each averaging period.

Commanded parameters describing the unique MET telemetry ID; the collection mode (averaging); the specific pressure barocap; the temperature and pressure event threshold values; the duration of each acquisition period (2 seconds); and the number of such periods are also provided.

The High resolution data from either message will be stored using the formats described in Appendix A and Appendix C, for label and data files, respectively.

An EMH data product will only be created if an Event is triggered while in Averaging Model or when the MET-PT is commanded to collect high resolution 2 second data.

Each EDR is comprised or one record with tabulated data (.TAB) provided in the following columns:

"EMH" – High Resolution					
Temporal	*Pressure*	*Temp 250*	*Temp 500*	*Temp 1000*	*Temp Ref*
Frame Count	Value				

5.4 MET Low-Resolution Pressure/Temperature Level 3 RDR (RML)

This RDR is derived from the corresponding EML data product, and will retain the same general file structure. The FRAME_COUNT values will be converted into units of DURATION in earth seconds since START_TIME (UTC); all temperature DNs will be converted to units of Kelvin.

Each RDR is comprised or one record with tabulated data (.TAB) provided in the following columns:

"RML" – Low Resolution						
Temporal	*Pressure*	*Temp 250*	*Temp 500*	*Temp 1000*	*Temp Ref*	*Trigger*
Duration	Average / Standard Deviation / Minimum / Maximum					Sensor

5.5 MET High-Resolution Pressure/Temperature Level 3 RDR (RMH)

This RDR is derived from the corresponding EMH data product, and will retain the same general file structure. The FRAME_COUNT values will be converted into units of DURATION in Earth seconds since START_TIME (UTC); all temperature DN will be converted to units of Kelvin.

Entries collected using the Averaging mode will appear in a block of 256, each 2 seconds apart, and will not appear at any fixed rate (i.e. the rate will depend on the threshold values and the atmospheric conditions).

Entries collected using the Full Rate mode will appear as a single continuous block, each 2 seconds apart.

An RMH data product will only be created if an Event is triggered while in Averaging Model or when the MET-PT is commanded to collect high resolution 2 second data.

Each RDR is comprised or one record with tabulated data (.TAB) provided in the following columns:

"RMH" – High Resolution					
Temporal	*Pressure*	*Temp 250*	*Temp 500*	*Temp 1000*	*Temp Ref*
Duration	Value				

5.6 MET High-Resolution Pressure-Corrected/Temperature Level 3 RDR (RMC)

This RDR is derived from the corresponding RMH data product (retaining the same file structure), and internal temperature data measured from within the pressure sensor housing (RMA).

Upon commencing surface operations, it was determined that a keep alive heater from the lidar caused interference with the pressure sensor, observable as duty cycling within the data, and particularly during night time hours. As a means to correct the data, an internal (housekeeping) thermocap measurement, made within the pressure sensor housing, is employed as an estimated measurement of the pressure sensor. The details are as follows:

> Conversion of the raw Barocap sensor output (referred to as y_p-values) to a pressure value is strongly dependent on the temperature of the Barocap, T_b. For the Phoenix units, in the temperature range anticipated, it was determined that $\partial p/\partial T_b = 5.34$ PaK^{-1}. Temperature is monitored by the Thermocap, T_T. The Barocap and Thermocap are close together and both attached to a PCB (Printed Circuit Board). The actual sources of heat are keep-alive heaters for the unit and a heat sink associated with the lidar but for our simple analysis it is

assumed that the PCB is the source or sink of heat as far as both the Barocap and Thermocap are concerned, and that the temperature of the PCB in the vicinity of these components is uniform. However the Barocap sensor head has a weaker thermal contact to the pressure sensors PCB than the Thermocap and, because of this, temperature differences can occur between these three components (the board, the Barocap and the Thermocap) if temperature rises or falls. During rising temperature the Barocap stays a little bit colder than the Thermocap and during falling temperature the Barocap stays a little bit warmer. The critical point is that the Thermocap temperature, T_T, is different from the required Barocap temperature T_b and a correction is needed to estimate a more accurate estimate of T_b.

We are advised that the Thermocap temperature sensor heads have a strong thermal coupling to the PCB and that the temperature measured by the Thermocaps is practically the same as the temperature of the PCB. The thermal contact between the PCB and the Barocap sensor heads is much weaker. In Martian pressures the effect of heat transfer by gas convection is negligible and only heat transfer by conduction needs to be taken into account when modelling the temperatures of the Barocap.

We assume that temperature change rate of the Barocap is proportional to the difference between the Barocap temperature T_b and PCB temperature T_{PCB}., and that $T_{PCB} = T_T$. Thus

$$\frac{dT_b}{dt} = \frac{1}{\lambda}(T_T - T_b) \qquad (1)$$

where λ is the time constant of the temperature adjustment. The value of λ can be determined from pressure chamber tests. The value is 78.7 s. An alternative analysis of the test data led to a value of 89.15 s but both give acceptable results for the pressure correction. Analyses were also made with T_{PCB} not equal to T_T but this did not make significant differences.

The solution of (1) can be written as

$$T_b(t) = e^{-(t-t_0)/\lambda} T_b(t_0) + \frac{1}{\lambda}\int_{t_0}^{t} e^{-(t-\tau)/\lambda} T_T(\tau) d\tau \qquad (2)$$

Using (2) the Barocap temperature at time point t can be calculated if the $T_T(t)$ and Barocap temperature at some earlier time point t_0 is known. Note that as $t-t_0$ increases dependence on the initial temperature $T_b(t_0)$ decays rapidly and there is minimal dependence on the initial condition.

Pressure data, calculated using the temperature T_T were collected at 0.5 Hz and transmitted to Earth. These pressures are generally reliable to within about 1 Pa but show anomalous oscillations, of amplitude 0.5 Pa on time scales of order 1 hour as heaters are turned on and off, especially at night.

Although 0.5 Hz pressure data were transferred to Earth throughout the mission, this was not true of T_T data. Due to data transfer capacity considerations it had been decided to only transfer sample "housekeeping" data associated with the pressure sensing system once every 512 s on a regular basis. We need 2-s data in order to properly integrate Equation (2).

In order to obtain a 2-s data set for T_T, a natural cubic spline fitted to the 512 s housekeeping data was adopted. This scheme also adds additional, "artificial", T_T points at locations where there is a rapid change in dT_T/dt and includes these in the construction of the spline.

This algorithm labels some data as “banned” and these are omitted from the archived data set (set to “-1” within the data product). The omissions are for periods when T_p data are suspect and are based on an expectation that pressures will not depart by more than 0.25 Pa from a linear fit across a period (usually a few 512-s blocks) of suspect T_p data. This loss of data will not be very important for studies of diurnal and seasonal pressure variations. For studies of short term (of order 30 s) dips in pressure associated with vortex passages it may be best to use the original, uncorrected data. The pressure oscillations associated with PCB heating have time scales of order 1000s rather than 30 s.

Each RDR is comprised or one record with tabulated data (.TAB) provided in the following columns:

“RMC” – Pressure Corrected, High Resolution					
Temporal	*Pressure*	*Temp 250*	*Temp 500*	*Temp 1000*	*Temp Ref*
Duration	Value				

5.7 MET Ancillary Pressure Sensor Temperature Data Level 3 RDR (RMA)

This RDR product is in support of the pressure correction (RMC) product, and provides housekeeping values of the pressure sensor-head temperature, measured once every 512 seconds.

Each RDR is comprised of one record with tabulated data (.TAB) provided in the following columns:

“RMA” – Pressure Sensor Temperature		
Temporal		*Internal Temperature of Pressure Sensor*
Duration	Frame Count	Value

APPENDIX A. EDR/RDR PDS LABEL ELEMENTS

Elements	Description	Format	
PDS_VERSION_ID	Represents the version number of the PDS standards document that is valid when a data product label is created. Values for the PDS_version_id are formed by appending the integer for the latest version number to the letters 'PDS'.	string	Source: PDS Location: All
RECORD_TYPE	Indicates the record format of a file. Note: In the PDS, when record_type is used in a detached label file it always describes its corresponding detached data file, not the label file itself. The use of record_type along with other file-related data elements is fully described in the PDS Standards Reference. All data products in this record will use “FIXED_LENGTH”.	string	Source: Static Value Location: All
RECORD_BYTES	The number of bytes in each record (row)	Integer	Source: Calculated Location: All
FILE_RECORDS	Indicates the number of physical file records, including both label records and data records. Note: In the PDS the use of FILE_RECORDS along with other file-related data elements is fully described in the Standards Reference.	Integer	Source: Static Value Location: All
DATA_SET_ID	A unique alphanumeric identifier for a data set or a data product. The DATA_SET_ID value for a given data set or product is constructed according to flight project naming conventions. In most cases the DATA_SET_ID is an abbreviation of the DATA_SET_NAME. "PHX-M-MET-2-PT-EDR-V1.0"	string	Source: PDS Location: All
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. Filename less the extension, e.g. "MS000EMH_00896227783_10C6M0"	string	Source: Calculated Location: All
PRODUCT_TYPE	The PRODUCT_TYPE data element identifies the type or category of a data product within a data set. Examples: EDR, RDR.	string	Source: Static Value Location: All
PRODUCT_VERSION_ ID	Identifies the version of an individual product within a data set.	string	Source: User Parameter

	For MER, PRODUCT_ VERSION_ID includes a Version field that begins with "V" followed by the Version decimal number of the controlling SIS document. Example: "V2.0 D-22850"		Location: All
RELEASE_ID	Unique identifier associated with the release to the public of all or part of a data set. The first release of a data set should have a RELEASE_ID of "0001" When a data set is released incrementally, such as every three months during a mission, the RELEASE_ID is updated each time part of the data set is released.	string	Source: User Parameter Location: All
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. "PHX"	string	Source: Static Value Location: All
INSTRUMENT_HOST_ NAME	Provides the full name of the host on which an instrument is based. "PHOENIX"	string	Source: Static Value Location: All
INSTRUMENT_ID	Provides an abbreviated name or acronym which identifies an instrument "MET"	string	Source: Static Value Location: All
INSTRUMENT_TYPE	Identifies the type of an instrument. "IN SITU METEOROLOGY"	string	Source: Static Value Location: All
LOCAL_TRUE_SOLAR_TIME	Local true solar time, or LTST, is one of two types of solar time used to express the time of day at a point on the surface of a planetary body. LTST is measured relative to the true position of the Sun as seen from a point on the planet's surface.	string	Source: Calculated Location: All
LOCAL_MEAN_SOLAR_TIME	Local mean solar time, or LMST, 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. LMST is calculated relative to the average position of the Sun over a Martian year.	string	Source: Calculated Location: All
MISSION_NAME	Identifies a major planetary mission or project. A given planetary mission may be associated with one or more spacecraft. "PHOENIX"	string	Source: Static Value Location: All
MISSION_PHASE_	Provides the commonly-used identifier of a	string	Source:

NAME	mission phase. e.g. “CRUISE”, “EXTENDED MISSION”, “PRIMARY MISSION”, “ATLO”, “ORT1”, “ORT2”, “TBD”		Operator Supplied Parameter Location: All
START_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) for the START of measurement. Days are measured in rotations of the planet in question from the reference day (which is day zero). Sols are also referenced to planning time: LMST.	integer	Source: Calculated Location: All
END_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) for which the measurement ENDS. Days are measured in rotations of the planet in question from the reference day (which is day zero).	integer	Source: Calculated Location: All
PRODUCER_ INSTITUTION_NAME	Identifies a university, research center, NASA center or other institution associated with the production of a data set. For MET P&T, this is given simply as “MET”	string	Source: Static Value Location: All
PRODUCT_CREATION_TIME	Defines the UTC system format time when a product was created. Formation rule: YYYY-MM-DDThh:mm:ss[.fff]		Source: Calculated Location: All
OPS_TOKEN	(PSI Token) Uniquely identifies a scientific observation within a data set. The value is an 16 bit hex number.	string	Source: PSI Software Tool Location: All
SPACECRAFT_CLOCK_CNT_ PARTITION	Indicates the clock partition active for the SPACECRAFT_CLOCK_START_COUN T and SPACECRAFT_CLOCK_STOP_COUNT elements. “1”	integer	Source: Static Value Location: All
SPACECRAFT_CLOCK_ START_COUNT	Provides the value of the spacecraft clock at the beginning of a time period of interest. Format is dddddddddd.ddd, measured in units of Seconds stored internally as a floating point number.	string	Source: Calculated Location: All
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]Z"	string	Source: Calculated Location: All
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:	string	Source: Calculated Location: All

	YYYY-MM-DDThh:mm:ss[.fff]Z"		
TARGET_NAME	Identifies a target. The target may be a planet, satellite, ring, region, feature, asteroid or comet. See TARGET_TYPE. "MARS",	string	Source: Static Value Location: All
TARGET_TYPE	Identifies the type of a named target. "PLANET"	string	Source: Static Value Location: All
SOFTWARE_NAME	Identifies data processing software such as a program or a program library. "MET-GDS"	string	Source: User Parameter Location: All
SOFTWARE_VERSION_ID	Indicates the version (development level) of a program or a program library. "V1"	string	Source: User Parameter Location: All
PROCESSING_ HISTORY_TEXT	Provides an entry for each processing step and program used in generating a particular data file. e.g. "CODMAC LEVEL 1 TO LEVEL 2"	string	Source: Static Value Location: All
OPS_TOKEN_ACTIVITY	Unique 8 bit hex MET derived telemetry ID, used to track commanded parameters and telemetry within the MET-GDS. Identical to the upper half of OPS_TOKEN.	string	Source: Commanded Value Location: All
PRESSURE_THRESHOLD	MET parameter used with PTAvg (8.53min averages) which triggers the storage of P&T High Resolution (2 sec) data. Occurs when $Pressure_{Max} – Pressure_{Min} > Pressure_{Threshold}$ This value will be -1 for Full Rate Data (no trigger values specified) that has been used to derive EML / RML data products.	Integer	Source: Commanded Value Location: All
INSTRUMENT_MODE_ID	MET Parameter used to distinguish Averaging data Message ("5") with Full Resolution data Message ("3") or Raw ("6")	Integer	Source: Commanded Value Location: All
DETECTOR_ID	MET Parameter used to distinguish one of three pressure sensors ("3", "5" and "6")	Integer	Source: Commanded Value Location: All
TEMPERATUE_THRESHOLD	MET parameter used with PTAvg (8.53min averages) which triggers the storage of P&T High Resolution (2 sec) data. Occurs when $Temp_{Max} – Temp_{Min} > Temp_{Threshold}$. This value will be -1 for Full Rate Data (no trigger values specified) that has been used to derive EML / RML data products..	Integer	Source: Commanded Value Location: All
INTEGRATION_DURATION	MET parameter describing the duration each averaging period spans. For EML and RML data records, this will always be 512 seconds; for EMH, RMH and RMC this will always be 2 seconds	Integer	Source: Static Value Location: All

PERIOD_NUMBER	MET parameter describing the number of 512 or 2 second periods for the data record.	Integer	Source: Commanded Value Location: All

APPENDIX B. SAMPLE LABEL FILE

EML.LBL

```
PDS_VERSION_ID                  = PDS3

/* FILE DATA ELEMENTS */

RECORD_TYPE                     = FIXED_LENGTH
RECORD_BYTES                    = 353
FILE_RECORDS                    = 154



/* POINTERS TO DATA OBJECTS */

^TABLE                          = "MS003EML_00896479378_10E0M0.TAB"


/* IDENTIFICATION DATA ELEMENTS */

DATA_SET_ID                     = "PHX-M-MET-2-PT-EDR-V1.0"
PRODUCT_ID                      = "MS003EML_00896479378_10E0M0"
PRODUCT_TYPE                    = EDR
PRODUCT_VERSION_ID              = "V1.1 D-33236"
RELEASE_ID                      = "0001"
INSTRUMENT_HOST_ID              = PHX
INSTRUMENT_HOST_NAME            = "PHOENIX LANDER"
INSTRUMENT_ID                   = MET
INSTRUMENT_TYPE                 = "IN SITU METEOROLOGY"
LOCAL_TRUE_SOLAR_TIME           = "11:25:27"
LOCAL_MEAN_SOLAR_TIME           = "11:02:15"
MISSION_NAME                    = "PHOENIX"
MISSION_PHASE_NAME              = "EXTENDED MISSION"
PLANET_DAY_NUMBER               = 3
PRODUCER_INSTITUTION_NAME       = "YORK UNIVERSITY"
PRODUCT_CREATION_TIME           = 2008-06-05T09:07:29.255
OPS_TOKEN                       = "16#10E00000#"
SPACECRAFT_CLOCK_CNT_PARTITION  = 091
SPACECRAFT_CLOCK_START_COUNT    = "896479377.648"
START_TIME                      = 2008-08-27T06:10:32.777
STOP_TIME                       = 2008-08-28T07:12:22.777
TARGET_NAME                     = MARS
TARGET_TYPE                     = PLANET

/* HISTORY DATA ELEMENTS */

SOFTWARE_NAME                   = "MET-GDS"
SOFTWARE_VERSION_ID             = "3.0.5"
PROCESSING_HISTORY_TEXT         = "CODMAC LEVEL 0 TO LEVEL 1"

/* COMMANDED PARAMETERS */

OPS_TOKEN_ACTIVITY              = "16#10E0#"
INSTRUMENT_MODE_ID              = 5
```

```
DETECTOR_ID                      = 3
PRESSURE_THRESHOLD               = 0
TEMPERATURE_THRESHOLD            = 0
PERIOD_DURATION                  = 512
PERIOD_NUMBER                    = 154


/* TABLE DATA ELEMENTS */

OBJECT                           = TABLE
INTERCHANGE_FORMAT               = ASCII
  COLUMNS                        = 22
  ROWS                           = 154
  ROW_BYTES                      = 353

  OBJECT                         = COLUMN
    COLUMN_NUMBER                = 1
    NAME                         = "FRAME_COUNT"
    DATA_TYPE                    = ASCII_INTEGER
    START_BYTE                   = 1
    BYTES                        = 15
    UNIT                         = "DN"
    DESCRIPTION                  = "The absolute number of frames since
                                    instrument power up."
  END_OBJECT                     = COLUMN

  OBJECT                         = COLUMN
    COLUMN_NUMBER                = 2
    NAME                         = "AVERAGE_PRESSURE"
    DATA_TYPE                    = ASCII_REAL
    START_BYTE                   = 17
    BYTES                        = 15
    UNIT                         = "PASCAL"
    DESCRIPTION                  = "The average pressure over each 512
                                    second period."
  END_OBJECT                     = COLUMN

  OBJECT                         = COLUMN
    COLUMN_NUMBER                = 3
    NAME                         = "STANDARD_DEVIATION_PRESSURE"
    DATA_TYPE                    = ASCII_REAL
    START_BYTE                   = 33
    BYTES                        = 15
    UNIT                         = "PASCAL"
    DESCRIPTION                  = "The pressure standard deviation
                                    over each 512 second period."
  END_OBJECT                     = COLUMN

  OBJECT                         = COLUMN
    COLUMN_NUMBER                = 4
    NAME                         = "MINIMUM_PRESSURE"
    DATA_TYPE                    = ASCII_REAL
    START_BYTE                   = 49
    BYTES                        = 15
    UNIT                         = "PASCAL"
    DESCRIPTION                  = "The minimum pressure over each 512
                                    second period."
```

```
END_OBJECT                    = COLUMN

OBJECT                        = COLUMN
  COLUMN_NUMBER               = 5
  NAME                        = "MAXIMUM_PRESSURE"
  DATA_TYPE                   = ASCII_REAL
  START_BYTE                  = 65
  BYTES                       = 15
  UNIT                        = "PASCAL"
  DESCRIPTION                 = "The maximum pressure over each 512
                                 second period."
END_OBJECT                    = COLUMN

OBJECT                        = COLUMN
  COLUMN_NUMBER               = 6
  NAME                        = "250_AVERAGE_TEMPERATURE"
  DATA_TYPE                   = ASCII_INTEGER
  START_BYTE                  = 81
  BYTES                       = 15
  UNIT                        = "DN"
  DESCRIPTION                 = "The average temperature difference
                                 wrt REF_TEMP over each 512 second
                                 period for
                                sensor at 250mm."
END_OBJECT                    = COLUMN

OBJECT                        = COLUMN
  COLUMN_NUMBER               = 7
  NAME                        = "250_STANDARD_DEVIATION_TEMPERATURE"
  DATA_TYPE                   = ASCII_INTEGER
  START_BYTE                  = 97
  BYTES                       = 15
  UNIT                        = "DN"
  DESCRIPTION                 = "The temperature standard deviation
                                 over each 512 second period for
                                 sensor at
                                250mm."
END_OBJECT                    = COLUMN

OBJECT                        = COLUMN
  COLUMN_NUMBER               = 8
  NAME                        = "250_MINIMUM_TEMPERATURE"
  DATA_TYPE                   = ASCII_INTEGER
  START_BYTE                  = 113
  BYTES                       = 15
  UNIT                        = "DN"
  DESCRIPTION                 = "The minimum temperature over each
                                 512 second period for sensor at
                                 250mm."
END_OBJECT                    = COLUMN

OBJECT                        = COLUMN
  COLUMN_NUMBER               = 9
  NAME                        = "250_MAXIMUM_TEMPERATURE"
  DATA_TYPE                   = ASCII_INTEGER
  START_BYTE                  = 129
  BYTES                       = 15
```

```
    UNIT                          = "DN"
    DESCRIPTION                   = "The maximum temperature over each
                                     512 second period for sensor at
                                     250mm."
  END_OBJECT                      = COLUMN

  OBJECT                          = COLUMN
    COLUMN_NUMBER                 = 10
    NAME                          = "500_AVERAGE_TEMPERATURE"
    DATA_TYPE                     = ASCII_INTEGER
    START_BYTE                    = 145
    BYTES                         = 15
    UNIT                          = "DN"
    DESCRIPTION                   = "The average temperature difference
                                     wrt REF_TEMP over each 512 second
                                     period for
                                    sensor at 500mm."
  END_OBJECT                      = COLUMN

  OBJECT                          = COLUMN
    COLUMN_NUMBER                 = 11
    NAME                          = "500_STANDARD_DEVIATION_TEMPERATURE"
    DATA_TYPE                     = ASCII_INTEGER
    START_BYTE                    = 161
    BYTES                         = 15
    UNIT                          = "DN"
    DESCRIPTION                   = "The temperature standard deviation
                                     over each 512 second period for
                                     sensor at
                                    500mm."
  END_OBJECT                      = COLUMN

  OBJECT                          = COLUMN
    COLUMN_NUMBER                 = 12
    NAME                          = "500_MINIMUM_TEMPERATURE"
    DATA_TYPE                     = ASCII_INTEGER
    START_BYTE                    = 177
    BYTES                         = 15
    UNIT                          = "DN"
    DESCRIPTION                   = "The minimum temperature over each
                                     512 second period for sensor at
                                     500mm."
  END_OBJECT                      = COLUMN

  OBJECT                          = COLUMN
    COLUMN_NUMBER                 = 13
    NAME                          = "500_MAXIMUM_TEMPERATURE"
    DATA_TYPE                     = ASCII_INTEGER
    START_BYTE                    = 193
    BYTES                         = 15
    UNIT                          = "DN"
    DESCRIPTION                   = "The maximum temperature over each
                                     512 second period for sensor at
                                     500mm."
  END_OBJECT                      = COLUMN

  OBJECT                          = COLUMN
```

```
    COLUMN_NUMBER                = 14
    NAME                         = "1000_AVERAGE_TEMPERATURE"
    DATA_TYPE                    = ASCII_INTEGER
    START_BYTE                   = 209
    BYTES                        = 15
    UNIT                         = "DN"
    DESCRIPTION                  = "The average temperature difference
                                    wrt REF_TEMP over each 512 second
                                    period
                                   for sensor at 1000mm."
  END_OBJECT                     = COLUMN

  OBJECT                         = COLUMN
    COLUMN_NUMBER                = 15
    NAME                         =
"1000_STANDARD_DEVIATION_TEMPERATURE"
    DATA_TYPE                    = ASCII_INTEGER
    START_BYTE                   = 225
    BYTES                        = 15
    UNIT                         = "DN"
    DESCRIPTION                  = "The temperature standard deviation
                                    over each 512 second period for
                                    sensor at
                                   1000mm."
  END_OBJECT                     = COLUMN

  OBJECT                         = COLUMN
    COLUMN_NUMBER                = 16
    NAME                         = "1000_MINIMUM_TEMPERATURE"
    DATA_TYPE                    = ASCII_INTEGER
    START_BYTE                   = 241
    BYTES                        = 15
    UNIT                         = "DN"
    DESCRIPTION                  = "The minimum temperature over each
                                    512 second period for sensor at
                                    1000mm."
  END_OBJECT                     = COLUMN

  OBJECT                         = COLUMN
    COLUMN_NUMBER                = 17
    NAME                         = "1000_MAXIMUM_TEMPERATURE"
    DATA_TYPE                    = ASCII_INTEGER
    START_BYTE                   = 257
    BYTES                        = 15
    UNIT                         = "DN"
    DESCRIPTION                  = "The maximum temperature over each
                                    512 second period for sensor at
                                    1000mm."
  END_OBJECT                     = COLUMN

  OBJECT                         = COLUMN
    COLUMN_NUMBER                = 18
    NAME                         = "REFERENCE_AVERAGE_TEMPERATURE"
    DATA_TYPE                    = ASCII_INTEGER
    START_BYTE                   = 273
    BYTES                        = 15
    UNIT                         = "DN"
```

```
    DESCRIPTION                    = "The average absolute temperature
                                      over each 512 second period  for
                                      reference PRT
                                     sensor."
  END_OBJECT                       = COLUMN

  OBJECT                           = COLUMN
    COLUMN_NUMBER                  = 19
    NAME                           =
"REFERENCE_STANDARD_DEVIATION_TEMPERATURE"
    DATA_TYPE                      = ASCII_INTEGER
    START_BYTE                     = 289
    BYTES                          = 15
    UNIT                           = "DN"
    DESCRIPTION                    = "The temperature standard deviation
                                      over each 512 second period for
                                      reference
                                     PRT sensor."
  END_OBJECT                       = COLUMN

  OBJECT                           = COLUMN
    COLUMN_NUMBER                  = 20
    NAME                           = "REFERENCE_MINIMUM_TEMPERATURE"
    DATA_TYPE                      = ASCII_INTEGER
    START_BYTE                     = 305
    BYTES                          = 15
    UNIT                           = "DN"
    DESCRIPTION                    = "The minimum absolute temperature
                                      over each 512 second period for
                                      reference PRT
                                     sensor."
  END_OBJECT                       = COLUMN

  OBJECT                           = COLUMN
    COLUMN_NUMBER                  = 21
    NAME                           = "REFERENCE_MAXIMUM_TEMPERATURE"
    DATA_TYPE                      = ASCII_INTEGER
    START_BYTE                     = 321
    BYTES                          = 15
    UNIT                           = "DN"
    DESCRIPTION                    = "The maximum absolute temperature
                                      over each 512 second period for
                                      reference PRT
                                     sensor."
  END_OBJECT                       = COLUMN

  OBJECT                           = COLUMN
    COLUMN_NUMBER                  = 22
    NAME                           = "EVENT_TRIGGER"
    DATA_TYPE                      = ASCII_INTEGER
    START_BYTE                     = 337
    BYTES                          = 15
    UNIT                           = "DN"
    DESCRIPTION                    = "0 = No event, 1 = Temperature Event
                                      Sensor 1,  2 = Temperature Event
                                      Sensor 2,
                                     3 = Temperature Event Sensor 3, 4 =
```

```
                                   Pressure Event, 9 = Full Rate Data
                                   (No Triggers Specified.)"
  END_OBJECT                     = COLUMN

END_OBJECT                       = TABLE

END
```

APPENDIX C. EDR/RDR DATA FILE ELEMENTS

Column	Description	Format	Notes*
FRAME_COUNT	Absolute Number of Frames Since Instrument Power Up (1 Frame (DN) = 2 sec)	ASCII_ INTEGER	Source: Telemetry Location: EML, EMH, RMA Unit: DN
DURATION	The number of Earth Seconds since START_TIME	ASCII_ REAL	Source: Calculation Location: RML, RMH, RMA, RMC Unit: Seconds
AVERAGE_PRESSURE	The average Pressure over a 256 frame, 512 second, averaging period.	ASCII_ REAL	Source: Telemetry Location: EML, RML Unit: Pascal
STANDARD_ DEVIATION_ PRESSURE	The Standard Deviation of the Pressure over a 256 frame, 512 second, averaging period.	ASCII_ REAL	Source: Telemetry Location: EML, RML Unit: Pascal
MINIMUM_PRESSURE	Minimum pressure observed over a 256 frame, 512 second, averaging period.	ASCII_ REAL	Source: Telemetry Location: EML, RML Unit: Pascal
MAXIMUM_PRESSURE	Maximum pressure observed over a 256 frame, 512 second, averaging period.	ASCII_ REAL	Source: Telemetry Location: EML, RML Unit: Pascal
250_AVERAGE_ TEMPERATURE	Average temperature wrt PRT for thermocouple at 250mm above deck over a 256 frame, 512 second, averaging period.	ASCII_ INTEGER, *ASCII_ REAL*	Source: Telemetry Location: EML, *RML* Units: DN, *Kelvin*
250_STANDARD_ DEVIATION_ TEMPERATURE	Standard Deviation in thermocouple at 250mm over a 256 frame, 512 second, averaging period.	ASCII_ INTEGER, *ASCII_ REAL*	Source: Telemetry Location: EML, *RML* Units: DN, *Kelvin*
250_MINIMUM_ TEMPERATURE	Maximum thermocouple at 250mm above deck wrt PRT observed over a 256 frame, 512 second averaging period.	ASCII_ INTEGER, *ASCII_ REAL*	Source: Telemetry Location: EML, *RML* Units: DN, *Kelvin*
250_MAXIMUM_ TEMPERATURE	Minimum thermocouple at 250mm above deck wrt PRT observed over a	ASCII_ INTEGER,	Source: Telemetry

	256 frame, 512 second, averaging period.	*ASCII_ REAL*	Location: EML, *RML* Units: DN, *Kelvin*
500_AVERAGE_ TEMPERATURE	Average temperature wrt PRT for thermocouple at 500mm above deck over a 256 frame, 512 second, averaging period.	ASCII_ INTEGER, *ASCII_ REAL*	Source: Telemetry Location: EML, *RML* Units: DN, *Kelvin*
500_STANDARD_ DEVIATION_ TEMPERATURE	Standard Deviation in thermocouple at 500mm above deck over a 256 frame, 512 second, averaging period.	ASCII_ INTEGER, *ASCII_ REAL*	Source: Telemetry Location: EML, *RML* Units: DN, *Kelvin*
500_MINIMUM_ TEMPERATURE	Minimum thermocouple at 500mm above deck, wrt PRT, observed over a 256 frame, 512 second, averaging period.	ASCII_ INTEGER, *ASCII_ REAL*	Source: Telemetry Location: EML, *RML* Units: DN, *Kelvin*
500_MAXIMUM_ TEMPERATURE	Maximum thermocouple at 500mm above deck, wrt PRT, observed over a 256 frame, 512 second, averaging period.	ASCII_ INTEGER, *ASCII_ REAL*	Source: Telemetry Location: EML, *RML* Units: DN, *Kelvin*
1000_AVERAGE_ TEMPERATURE	Average temperature wrt PRT for thermocouple at 1000mm above deck over a 256 frame, 512 second averaging period..	ASCII_ INTEGER, *ASCII_ REAL*	Source: Telemetry Location: EML, *RML* Units: DN, *Kelvin*
1000_STANDARD_ DEVIATION_ TEMPERATURE	Standard Deviation in thermocouple at 1000mm above deck average over a 256 frame, 512 second averaging period.	ASCII_ INTEGER, *ASCII_ REAL*	Source: Telemetry Location: EML, *RML* Units: DN, *Kelvin*
1000_MINIMUM_ TEMPERATURE	Minimum thermocouple at 1000mm above deck, wrt PRT, observed over a 256 frame, 512 second, averaging period.	ASCII_ INTEGER, *ASCII_ REAL*	Source: Telemetry Location: EML, *RML* Units: DN, *Kelvin*
1000_MAXIMUM_ TEMPERATURE	Maximum thermocouple at 1000mm above deck, wrt PRT, observed over a 256 frame, 512 second, averaging period.	ASCII_ INTEGER, *ASCII_ REAL*	Source: Telemetry Location: EML, *RML* Units: DN, *Kelvin*
REFERENCE_ AVERAGE_ TEMPERATURE	Averaged reference value from the PRT over a 256 frame, 512 second, averaging period.	ASCII_ INTEGER, *ASCII_ REAL*	Source: Telemetry Location: EML, *RML* Units: DN, *Kelvin*
REFERENCE_ STANDARD_ DEVIATION_ TEMPERATURE	PRT block Standard Deviation over a 256 frame, 512 second, averaging period.	ASCII_ INTEGER, *ASCII_ REAL*	Source: Telemetry Location: EML, *RML* Units: DN, *Kelvin*
REFERENCE_ MINIMUM_	Minimum PRT value over a 256 frame, 512 second, averaging period.	ASCII_ INTEGER,	Source: Telemetry

TEMPERATURE		*ASCII_ REAL*	Location: EML, *RML* Units: DN, *Kelvin*
REFERENCE_ MAXIMUM_ TEMPERATURE	Maximum PRT value over a 256 frame, 512 second, averaging period.	ASCII_ INTEGER, *ASCII_ REAL*	Source: Telemetry Location: EML, *RML* Units: DN, *Kelvin*
EVENT_TRIGGER	Details if an event triggered the High Resolution data acquisition, and which sensor triggered: 0 = No Event; 1 = Temperature 1 Event; 2 = Temperature 2 Event; 3 = Temperature 3 Event, 4 = Pressure Event. 9 = Full Rate Data, No Trigger Values specified.	ASCII_ INTEGER	Source: Telemetry of High-Res Data Location: EML, RML Unit: N/A
PRESSURE	Pressure value for 2 second Frame Period Duration = 2 second Frame Duration	ASCII_ REAL	Source: Telemetry Location: EMH, RMH, *RMC* Unit: Pascal
250_TEMPERATURE	Temperature value for thermocouple ~250 mm above deck wrt PRT over a 2 second Frame Period Duration = 2 second Frame Duration	ASCII_ INTEGER, *ASCII_ REAL*	Source: Telemetry Location: EMH, *RMH, RMC* Units: DN, *Kelvin*
500_TEMPERATURE	Temperature value for thermocouple ~500 mm above deck wrt PRT over a 2 second Frame Period Duration = 2 second Frame Duration	ASCII_ INTEGER, *ASCII_ REAL*	Source: Telemetry Location: EMH, *RMH, RMC* Units: DN, *Kelvin*
1000_TEMPERATURE	Temperature value for thermocouple ~1000 mm above deck wrt PRT over a 2 second Frame Period Duration = 2 second Frame Duration	ASCII_ INTEGER, *ASCII_ REAL*	Source: Telemetry Location: EMH, *RMH, RMC* Units: DN, *Kelvin*
REFERENCE_ TEMPERATURE	Temperature value for the PRT over a 2 second Frame Period Duration = 2 second Frame Duration	ASCII_ INTEGER, *ASCII_ REAL*	Source: Telemetry Location: EMH, *RMH, RMC* Units: DN, *Kelvin*
SENSOR_ TEMPERATURE	Internal Temperature of the pressure sensor, as measured from housekeeping data	*ASCII_ REAL*	Source: Housekeeping Location: RMA Units: *Kelvin*

*Roman and Italic Data “Locations” correlate with Roman and Italic “Units”

APPENDIX D. EDR/RDR FILE NAMING CONVENTION

The file naming scheme defined for the MET complies with the filenaming conventions for Phoenix EDRs and RDRs which in turn complies with the PDS Level II 27.3 filename standards.

Each product name is uniquely identifiable throughout the mission.

A template for MET filenames is shown below:

Position	Name	Description/value
1	Instrument	**S** SSI **R** RAC **T** TEGA **A** RA **O** MECA-OM **P** MECA-TECP **F** MECA-AFM **W** MECA-WCL **M** MET-P&T **L** MET-LIDAR **D** MARDI **E** ASE
2	Source/Epic	Spacecraft **S** Surface, flight model **T** Test-bed **C** Cruise, flight model
3-5	SOL	Solar days since first full day on Mars. Landing day is Sol zero. If Source/Epic is T, day of year should be used (ERT or SCET). For cruise phase, always set to "_C_".
6-8	Product Type	Lidar: ELP, RLP – **L**idar 532 nm **P**hoton Counting (**E**DR/**R**DR) ELA, RLA – **L**idar 532/1064 nm **A**nalog (**E**DR/**R**DR) ELS, RLS – **L**idar **S**upplemental Data (**E**DR/**R**DR) P&T: EML, RML – **MET** P&T **L**ow-Res (512sec) (**E**DR/**R**DR) EMH, RMH – **MET** P&T **H**i-Res (2sec) (**E**DR/**R**DR) RMC – **MET** P&T Pressure **C**orrection (**R**DR) RMA – **MET** P&T Pressure **A**ncillary (**R**DR)
9	Reserved	Filler "_"
10-20	SCLK	Spacecraft Clock (zero padded)
21	Reserved	Filler "_"
22-25	Ops Token	Last 4 digits of PSI Token (Instrument + Command IDs = MET ID)
26	Producer (Reserved)	M – MET Team
27	Version	Version number, 0-9,A-Z (36 total)
28	Period	Always set to "." (ASCII period)
29-31	File Extension	PDS file extensions. **LBL** Label File **TAB** Data Table

APPENDIX E. PHOENIX TOKEN CONVENTION

A template for Token Conventions is shown below (All values given in Hex format):

Position	***Name***	***Description/value***
1-4	Ops Token	Assigned value for each unique PSI Activity
5-8	Command ID	Payload specified ID, for MET: Not Used

Ops Token presently identified as positions 1-4, to be used as MET-GDS "Telemetry ID."

APPENDIX F. MARS24 ALGORITHM

(Updated May 20, 2008)

The calculations made by Mars24 to determine the time for a given location on Mars are primarily based on Allison and McEwen (2000) (henceforth AM2000). We also refer to Allison (1997) (henceforth A1997), of which AM2000 was a thorough update. However, some typographical errors appeared in the published version of AM2000, and some calculations have been revised since publication of that paper because of the availability of new data. (These updates may appear in a paper in preparation by Allison and Ferrier.)

Consequently, we provide here step-by-step documentation of the equations employed by Mars24 for users who wish to implement their own Martian timekeeping applications. At the end of this presentation, we also provide worked examples for verification of intermediate results.
I. Equations

A. Determine Days Since J2000 Epoch

Our Mars time calculations will use the parameter ΔtJ2000, the elapsed time in days since the J2000 epoch, i.e., 12:00 on Jan. 1, 2000 (TT). The following describes how we get there from a Java call which returns the system time. If one has an alternative scheme for obtaining ΔtJ2000, then these steps can be skipped.

A-1. Get a starting Earth time.

Mars24 is written in Java, so we use the System.currentTimeMillis() method to find out the number of milliseconds, millis, that have elapsed since 00:00:00 on Jan. 1, 1970 (i.e., the Unix epoch).

Unfortunately, most if not all Java implementations (or the operating systems on which they run) do not keep track of leap seconds, i.e., they keep time in UT rather than UTC. Consequently, when Mars24 calls System.currentTimeMillis(), it assumes that the value returned uses UT rather than UTC.

Only A-2 in the following steps explicitly uses millis. However, there are some displays in Mars24 which also use other readings based on the value of millis, so there is a possibility that if Mars24 is used on a computer and Java implementation which do keep track of leap seconds, display errors could result.

A-2. Convert millis to Julian Date (UT).

Although there's plenty of sample code available on-line which demonstrates how to convert a Gregorian calendar date to a Julian Date, we simply use the offset from a known, recent Julian Date. Again, we use the Unix epoch, 00:00:00 on Jan. 1, 1970.

JDUT = 2440587.5 + (millis / 8.64×107 ms/day)

A-3. Determine time offset from J2000 epoch (UT).

This step is optional; we only need to make this calculation if the date is before Jan. 1, 1972. Determine the elapsed time in Julian centuries since 12:00 on Jan. 1, 2000 (UT).

T = (JDUT - 2451545.0) / 36525.

A-4. Determine UTC to TT conversion. (Replaces AM2000, eq. 27)

Terrestrial Time (TT) advances at constant rate, as does UTC, but no leap seconds are inserted into it and so it gradually gets further ahead of UTC. The best way to determine the difference between TT and UTC is to consult a table of leap seconds. Alternatively, one could try to use an empirical formula.

In Mars24 we, oddly enough, use both methods. We use the USNO table for dates after Jan. 1, 1972, and a formula for dates prior to then. In consulting the USNO table, however, it is important to note that the table provides values for the TAI-UTC difference, where TAI is International Atomic Time. To obtain the TT-UTC difference, add 32.184 seconds to the value of TAI-UTC. For example, the USNO table indicates that on Jan. 1, 2006, the TAI-UTC value is 33.0 seconds, and thus, the value for TT-UTC on that date (and until the next date on which another leap second is added to the clock) would be 33.0s + 32.184s = 35.184s.

The formula applied for dates prior to Jan. 1, 1972, is similar to AM2000, eq. 27, but has been revised and includes additional terms:

$TT - UTC = 64.184s + 59\ s \times T - 51.2\ s \times T2 - 67.1\ s \times T3 - 16.4\ s \times T4$

(Note: Mars24 uses the USNO table which includes the leap second added Jan. 1, 2006. Obviously, then, it does not allow for any leap seconds which might be subsequently added. Bulletin C 33 from the IERS Earth Orientation Centre indicates this will not occur any earlier than Jan. 1, 2009.)

A-5. Determine Julian Date (TT).

$JDTT = JDUT + [(TT - UTC) / 86400\ s{\cdot}day-1]$

A-6. Determine time offset from J2000 epoch (TT). (AM2000, eq. 15)

$\Delta tJ2000 = JDTT - 2451545.0$

B. Determine Mars Parameters of Date

Now we turn our attention to Mars, first determining some orbital paramaters.

B-1. Determine Mars mean anomaly. (AM2000, eq. 16)

$M = 19.3870° + 0.52402075° \Delta tJ2000$

B-2. Determine angle of Fiction Mean Sun. (AM2000, eq. 17)

$\alpha FMS = 270.3863° + 0.52403840° \Delta tJ2000$

B-3. Determine perturbers. (AM2000, eq. 18)

$PBS = \Sigma(i=1,7)\ Ai \cos [(0.985626° \Delta tJ2000 / \tau i) + \varphi i]$

where 0.985626° = 360° / 365.25, and

i	Ai	τi	φi
1	0.0071	2.2353	49.409
2	0.0057	2.7543	168.173
3	0.0039	1.1177	191.837
4	0.0037	15.7866	21.736
5	0.0021	2.1354	15.704
6	0.0020	2.4694	95.528
7	0.0018	32.8493	49.095

B-4. Determine Equation of Center. (Bracketed term in AM2000, eqs. 19 and 20)

The equation of center is the true anomaly minus mean anomaly.

$\nu - M = (10.691° + 3.0° \times 10^{-7}\ \Delta t_{J2000}) \sin M + 0.623° \sin 2M + 0.050° \sin 3M + 0.005° \sin 4M + 0.0005° \sin 5M + PBS$

B-5. Determine areocentric solar longitude. (AM2000, eq. 19)

$L_s = \alpha_{FMS} + (\nu - M)$

C. Determine Mars Time

C-1. Determine Equation of Time. (AM2000, eq. 20)

$EOT = 2.861° \sin 2L_s - 0.071° \sin 4L_s + 0.002° \sin 6L_s - (\nu - M)$

The above result for EOT is in degrees. Multiply by (24 h / 360°) = (1 h / 15°) to obtain the result in hours.

C-2. Determine Coordinated Mars Time. (AM2000, eq. 22, modified)

This is the mean solar time at Mars' prime meridian.

$MTC = \text{mod}24\ \{ 24\ h \times ([(JD_{TT} - 2451549.5) / 1.027491252] + 44796.0 - 0.00096) \}$

The function modX indicates a re-setting of the function parameter, a cyclical value, to a value between 0 and X. In this case, we apply mod24 to indicate that values outside the range 0-24 should be re-set to be within that range, e.g., mod24 (30) = 6.

C-3. Determine Local Mean Solar Time.

The Local Mean Solar Time for a given planetographic longitude, Λ, in degrees west, is easily determined by offsetting from the mean solar time on the prime meridian.

$LMST = MTC - \Lambda\ (24\ h / 360°) = MTC - \Lambda\ (1\ h / 15°)$

C-4. Determine Local True Solar Time. (AM2000, eq. 23)

$LTST = LMST + EOT\ (24\ h / 360°) = LMST + EOT\ (1\ h / 15°)$