TREK-USER-002

TREK TELEMETRY TUTORIAL

November 2012

TABLE OF CONTENTS

TABLES

<u>TABLE</u> <u>PAGE</u>

FIGURES

1. Introduction

Telemetry is binary data sent from a remote source to be processed by a computer. Telescience Resource Kit (TReK) can process the telemetry for you and provide you with the tools you need to get the processed data. This tutorial describes telemetry data as it relates to TReK. You will be introduced to many terms and concepts related to telemetry. Some of the terms and concepts are generic. Others are specific to TReK, the Payload Operations Integration Center (POIC), or both. You will also learn enough about telemetry processing to see how it relates to some of the TReK products. This discussion will be slanted towards remote users for International Space Station (ISS), but most of the concepts would apply to other projects.

The document is divided into the following sections:

Section 1 Introduction
You're reading it now.

Section 2 Telemetry Basics
This section describes some of the basic terms and concepts related to telemetry. These topics will include items specific to TReK and topics common between TReK and the Payload Operations Integration Center (POIC).

Section 3 Getting the Packet to TReK
This section covers how packets get to TReK, how TReK identifies a packet, and what TReK will do with the packet once it is identified.

Section 4 Processing the Data
This section covers the different types of processing TReK can perform on the packet once it is identified.

Section 5 Using the Data
The section covers some detailed items about TReK. Knowledge of the items in this section is needed in understanding some of the other user documentation.

Section 6 TReK Telemetry Processing Philosophy
This section covers the TReK philosophy for telemetry processing. There are some features that TReK handles differently than the POIC. Those differences are explained in this section.

Section 7 Questions/Answers
This section contains the topics from the previous sections and additional topics arranged in a question/answer format. This section is intended to be a reference for use after reading the previous sections. Some of the information contained in the previous sections

will not be available in this section. In addition, this section may have items that are not covered in the previous sections. These items are marked with NEW.

For further information on telemetry in the POIC see the following documents:

- MSFC-STD-1274: MSFC HOSC Telemetry Format Standard Volume 2
- HOSC-EHS-065: The EHS Concepts and Scenarios Document
- MSFC-DOC-1949: MSFC HOSC Database Definitions
- SSP-50304: POIC Capabilities Document
- SSP-50305: POIC To Generic User IDD Volume 1

2. Telemetry Basics

Many terms and concepts related to telemetry are shared between TReK and the POIC. This section will cover those items as well as some terms and concepts unique to TReK. Some of the terminology is slightly different for TReK. This section will also correlate terminology for TReK to similar terms in the POIC.

2.1 Packet Contents

The POIC uses the terms packet, subset, and MSID to refer to the items you receive. A packet is just a collection of binary data that is transmitted over a network to your computer for TReK to process.

A subset is a collection of binary data within a packet. You can think of it as a packet within a packet. The only important difference is that you don't transmit a subset by itself.

The final term you will see is MSID. MSID, or Measurement Stimulus Identifier, is a name given to a collection of bits within a packet that has a value. An MSID can be a string, integer, floating point, or time value. The more generic term, parameter, will be used in this document.

A packet can contain subsets and parameters. A subset can contain other subsets and parameters. A parameter is the lowest level in a packet and cannot contain other data. Much of the TReK documentation will refer to packets, subsets, and MSIDs as parameters or measurements. TReK uses these generic terms to mean any piece of the data in a packet.

2.2 Anatomy of a Packet

A packet is divided into two parts as shown in Figure 1. The header varies in length based on the type of packet, and is used in identifying the remainder of the packet. The data zone in the packet contains the data. In general, the header will contain parameters and the data zone will contain subsets and parameters. All of the packets described in the following sections have this same basic layout.

Header	Data Zone

Figure 1 Parts of a Packet

2.3 CCSDS Packets

The Consultative Committee For Space Data Systems (CSSDS) packet is used to hold all of your experiment data. All of the packets and headers described below are used by different computer systems to identify and route data to different places. The CCSDS packet is contained in all of the other packets. The CCSDS packet is composed of a primary header, an optional secondary header, and a user data zone. The primary header contains information used to identify the packet. Your telemetry data (subsets and parameters) will appear in the user data zone of the packet.

CCSDS Primary Header	Optional CCSDS Secondary Header	User Data Zone

Figure 2 CCSDS Packet

2.4 Data Sources and the Packets They Send

You will receive packets from different sources. These packets have different header information that is used in identifying the contents of the packet. Ultimately all data TReK receives will be a CCSDS packet as described in Section 2.3. Figure 3 shows a high level view of the data sources from which TReK can receive data: the POIC, Suitcase Simulator, and the TReK Telemetry Trainer.

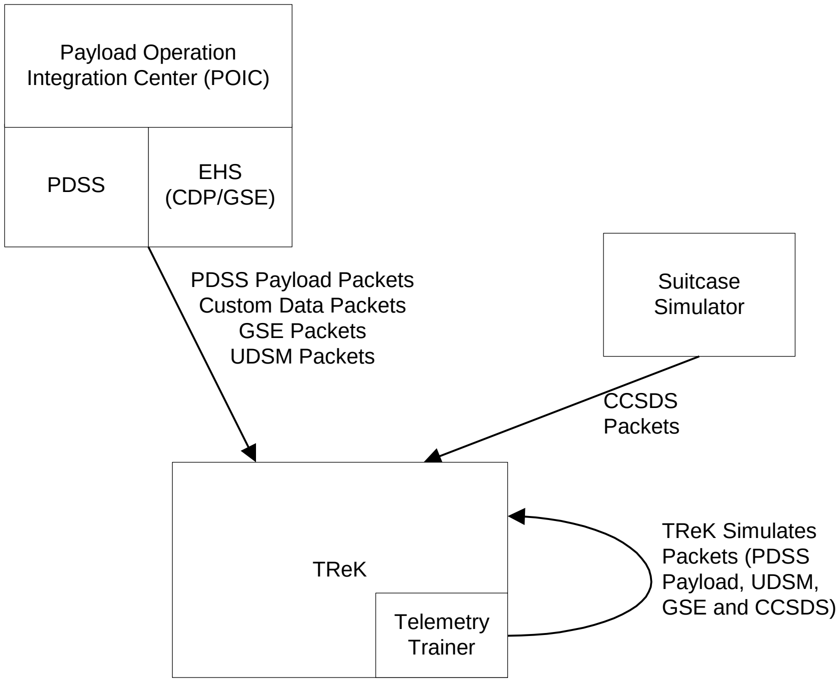

Figure 3 Data Sources

2.4.1 Payload Operations Integration Center

All of the data that you receive from the Payload Operations Integration Center will be EHS Protocol Packets. All of these packets will begin with the same header. This header is called the Primary EHS Protocol Header. You can look in SSP-50305 POIC to Generic User IDD Volume 1 for the details of this header. All you really need to know for now is that the header contains information that will allow TReK to identify the contents of the packet.

EHS Protocol Packets also have a secondary header that is specific to the type of data being sent. The last part of the EHS packet is a CCSDS packet that contains the actual subsets and parameters you need. Figure 4 shows the structure of an EHS Protocol Packet. The Primary EHS Protocol Header always contains the same data. The Secondary EHS Protocol Header will change based on what type of data is being sent in the CCSDS Packet.

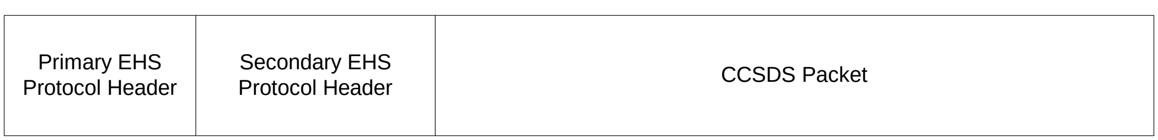

Figure 4 EHS Protocol Packet

The following sections describe each type of EHS Protocol Packet that can be processed by TReK.

2.4.1.1 Payload Data Services System (PDSS) Payload Packets

PDSS receives data from the International Space Station via White Sands Complex. PDSS performs some of its own telemetry processing to determine the contents of the data that was sent and where the data should be forwarded.

International Space Station sends data in CCSDS packets. After determining the contents of the CCSDS packet, PDSS will add the EHS Protocol Headers to the beginning of the CCSDS packet before sending it to remote users. Figure 5 shows the different parts of a PDSS Payload Packet.

Primary EHS Protocol Header	Secondary EHS Protocol Header for PDSS Payload CCSDS Packet	CCSDS Packet

Figure 5 PDSS Payload Packet

The Primary EHS Protocol Header contains information such as the data mode and secondary header protocol type. The secondary header protocol type identifies what follows the Primary EHS Protocol Header, in this case a PDSS Payload Packet Secondary Header and a CCSDS packet. The secondary header contains information specific to a PDSS Payload Packet. After the PDSS Payload Secondary Header is the actual CCSDS packet that was transmitted by International Space Station. The content of a CCSDS Packet is described in Section 2.3.

2.4.1.2 PDSS UDSM Packets

PDSS User Data Summary Message (UDSM) Packets are sent by PDSS to remote users. These packets will be sent for the following reasons.

- There is a scheduled Loss of Signal
- The PDSS operator requests a UDSM to be sent
- It is the end of a user’s data as specified by the payload mission/increment timeline.

These messages contain information about the data quality and quantity for a particular APID. The information includes start and stop times, number of packets transmitted by PDSS, and various errors. This information can be correlated with information from TReK about the number of packets received and processed to identify potential data loss on the network. (See section 2.6 for more information on data transmission and potential data loss.)

Primary EHS Protocol Header	Secondary EHS Protocol Header for PDSS UDSM Packet	CCSDS Packet

Figure 6 PDSS UDSM Packet

2.4.1.3 Custom Data Packets

The POIC can send Custom Data Packets to your TReK computer. You define Custom Data Packets. You tell the POIC what parameters you want and they will be sent in a Custom Data Packet.

The Custom Data Packet has the format shown in Figure 7. The EHS primary header is the same as the PDSS payload packet. There is a Custom Data Packet specific secondary header followed by the data you requested in a CCSDS packet.

The Custom Data Packet can contain raw (unprocessed), converted, calibrated, and limit/expected state sensed data (these terms are explained in Section 4). The user specifies the type of data in a request. For more information on how TReK will process Custom Data Packets, see Section 4.8.

Primary EHS Protocol Header	Secondary EHS Protocol Header for Custom Data Packet	CCSDS Packet

Figure 7 Custom Data Packet

2.4.1.4 Ground Support Equipment Packets

There are two types of Ground Support Equipment (GSE) packets: ground ancillary and user-defined. Ground ancillary GSE packets contain a set of selected onboard core system and payload health and status data downlinked from the ISS. You will receive these packets instead of the PDSS core packets. PDSS core packets are not sent outside of the POIC.

The second type of GSE packet is the user-defined packet. Users can define packets that the POIC can send to user GSE (in this case, TReK). The major difference between GSE packets and Custom Data Packets is that GSE packets are defined in the POIC database. The actual format inside the CCSDS Packet is identical. (For information on how to enter a GSE Packet definition, see POIC documentation.)

The format of a GSE packet is shown in Figure 8. A GSE secondary header follows the EHS primary header. The secondary header is the same for ground ancillary and user-defined GSE packets. The CCSDS packet contains the parameters for the packet. For more information on how TReK processes GSE Packets, see Section 4.8.

Primary EHS Protocol Header	Secondary EHS Protocol Header for GSE Packet	CCSDS Packet

Figure 8 Ground Support Equipment (GSE) Packet

2.4.2 Suitcase Simulator

TReK can be connected to a Suitcase Simulator to interface with your experiment while it is still on the ground. The Suitcase Simulator will generate CCSDS packets (see Section 2.3) that can be processed by TReK.

2.4.3 TReK Telemetry Trainer

You may not always have the POIC or a suitcase simulator to help you test your user products (displays, scripts, and computations). TReK provides a Telemetry Trainer to act as a data source for your TReK system. The TReK Telemetry Trainer can send simulated PDSS payload packets to your TReK platform. These packets are formatted exactly as the PDSS payload packet described in section 2.4.1.1. In addition, Telemetry Trainer can be configured to send other types of packets as well such as CCSDS packets to simulate data from the Suitcase Simulator.

2.5 Identifying the Packet

When you set up your TReK system to process telemetry, you will have to let TReK know some of the attributes of the packet. TReK uses this information to identify the packets it receives. If the packet received does not match any of the ones you have chosen to process, TReK will discard the packet and not process it any further.

Three items make a packet unique to TReK. They are

1. The application process identifier (APID) in the CCSDS Primary Header.
2. The Data Mode in the Primary EHS Protocol Header (e.g., RealTime, Dump1).
3. The Packet Type. This is identified from the secondary protocol header type in the Primary EHS Protocol Header (e.g., PDSS Payload, GSE).

Note: Packets sent from a Suitcase Simulator system do not contain an EHS Protocol Header. For these packets the Data Mode is considered "none" and the packet type is *assumed to be "suitcase simulator". See the Telemetry Processing Users Guide (TREK-USER-003) for more information on configuring TReK to receive and process data.*

2.6 How The Data Gets To You

The telemetry packets described in the previous section travel from the source to TReK via the internet or some other network. Your computer will have an Ethernet network card to connect to the network to receive the data.

Note: For non-simulation tests you do not have to have a network card. See the TReK Telemetry Processing User Guide (TREK-USER-003) and the TReK Telemetry Trainer User Guide (TREK-USER-004) for details on setting up your computer if you don't have a network card. However, you will need a network card to participate in any remote simulations or tests.

Just as TReK needs to know what kind of packet it is to process, TReK must also know how the packet is delivered. There are two network transmission protocols supported by TReK: TCP and UDP. They are described briefly in the following sections. TReK will automatically know which transmission protocol to use based on the packet type to be processed.

2.6.1 Transmission Control Protocol (TCP)

TCP is a connection-oriented protocol. The computer sending the data will actually connect to the computer that it wants to send the data to. Since there is a connection between the two computers all of the data will be delivered in the order it was sent.

Custom Data Packets are sent to TReK using TCP.

2.6.2 User Datagram Protocol (UDP)

UDP is a connectionless-oriented protocol. The computer sending the data doesn't connect to a specific computer, but instead transmits the data to an address. UDP allows for data to be sent at higher rates than TCP. However, since the data is transmitted without a connection, the computer sending the data does not know if the other computer received it. In addition, the data may be received out of order.

PDSS Payload Packets (including those generated by the TReK Telemetry Trainer), PDSS UDSM Packets, GSE Packets, and Suitcase Simulator Packets (CCSDS Packets) are sent using UDP.

TReK provides statistics on all packets it is configured to receive and all ports that are receiving the data. See the Telemetry Processing Users Guide (TREK-USER-003) for details on the statistics TReK maintains.

2.7 Processing The Data

TReK and the POIC can process the data for you in many ways. The terms used are similar and the differences are explained here.

TReK can receive data that is unprocessed (sometimes called raw data) and data that is processed. All data except parameters in the CCSDS data zone of the GSE packets and Custom Data Packets are considered raw. (Processed data is discussed in Section 4.8.) If a raw data pattern has a value, it can be converted to a type that the TReK computer can understand. For example, an integer may have a different binary representation on the International Space Station (ISS) than TReK. TReK translates, or converts, this pattern to the native PC integer type.

Once a value is converted, TReK can also calibrate the value. Calibration is the transformation of a parameter to a desired physical unit or text state code. These calibrated parameters can then be "sensed" for limit, delta, or expected state violations. You can also have TReK monitor these sensed values for violations.

All of this is explained in detail in Section 4.

3. Getting the Packet to TReK

The data that TReK receives is binary data. TReK must determine what this binary data represents and pull the pieces out of the packet. The following sections describe what TReK does when it receives a packet.

3.1 Telemetry Processing: The Big Picture

Before getting into some of the details about how TReK processes data, let's look at an overview of how the packets move through the TReK system.

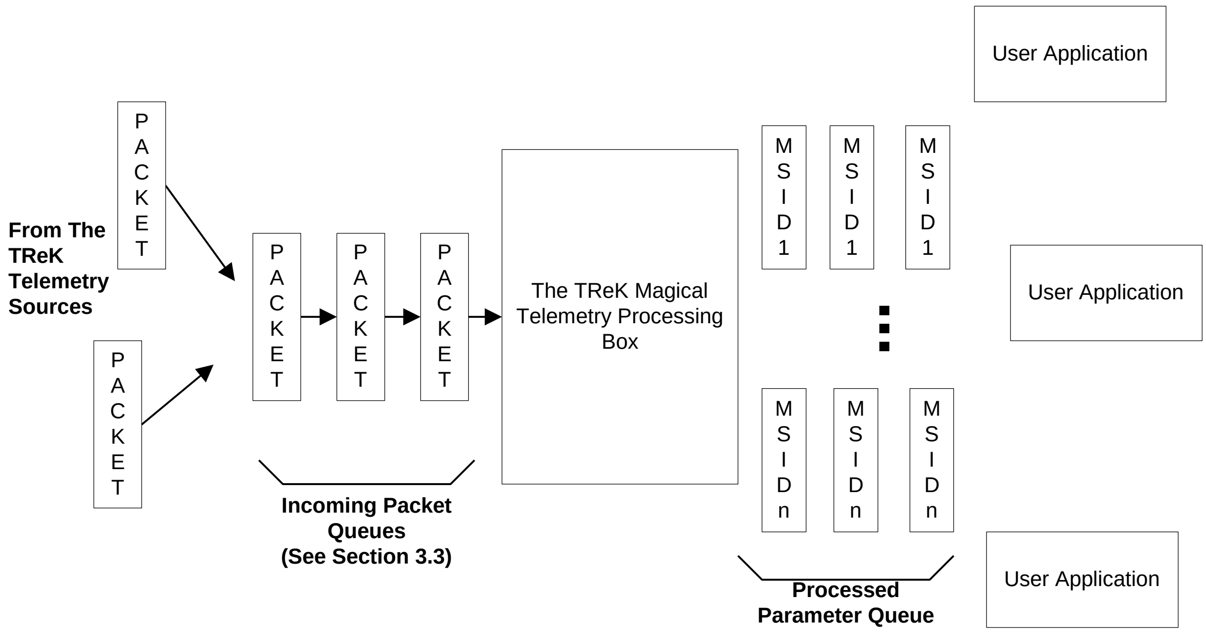

Figure 9 The BIG Picture

Figure 9 shows a simplified high-level picture of the flow of the data through your TReK system. The different data sources send packets to TReK that are queued and processed. You can set the size of the incoming packet queue and other attributes about the incoming packet queue in the TReK Telemetry Processing application (see TReK Telemetry Processing User Guide (TREK-USER-003) for details). The different queues in TReK are also explained further in Section 3.4.

The packets are sent through the TReK Magical Telemetry Processing Box and appear again on the other side. The following sections tell you what happens in the magical box. For now let's just see what comes out of it.

The processed data also comes out in a queue. The different parameters (including packets and subsets) can be retrieved by your user applications. The size of the processed queue can also be set through the TReK Telemetry Processing application.

Well that's a quick overview of the big picture. Now for what goes on inside that "magical" box.

3.2 Identifying the Packet

The first thing TReK must determine is what kind of data it received. You will configure TReK through the Telemetry Processing application to process certain packets. TReK processes the header of the packet to determine what type of packet has been received.

TReK uses three attributes to identify a packet: packet type, data mode, and the application process identifier (APID). The packet type identifies the exact source and type of data. For packets sent with the EHS Protocol Header this is equivalent to the secondary header protocol type found in the Primary EHS Protocol Header (e.g., PDSS Payload). For packets such as those sent from Suitcase Simulator, the packet type identifies the source packet type (e.g., Suitcase Simulator).

The data mode is another attribute found in the Primary EHS Protocol Header. This allows you to receive the same packet from different time segments without getting the data intertwined. There are 16 data modes supported by TReK. (These are the same data modes that are defined in SSP-50305 POIC To Generic User IDD and the ***none*** data mode for Suitcase Simulator packets.) The Real Time data mode identifies packets that are currently being sent from International Space Station experiments. The three Dump data modes identify packets that are being sent from the Communications Outage Recorder (COR) aboard ISS. If the data was recorded on the ground and is being replayed, then the data mode is set to one of the 11 playback channel numbers (Playback 1, Playback 2, etc.). For data without the EHS protocol header (e.g., Suitcase Simulator Packets) the data mode is “none”.

As mentioned before, each packet that TReK receives contains a CCSDS packet. The primary header of the CCSDS packet contains an integer value known as the application process identifier (APID). This value is used by TReK in identifying a packet as part of a unique stream of packets.

Once TReK identifies the packet, TReK can determine if the packet is one that has been selected for processing. If TReK receives a packet that it is not configured to process, the data for that packet is not processed further. Otherwise, the parameters are extracted from the packet as described later.

3.3 Minding Your Packets And Queues

Figure 9 showed you a high level view of how TReK gets data to your applications. Now is a good time to explain more about the different queues on TReK and why they are needed.

All of the packets TReK receives and processes are sent asynchronously. While it is possible to know about how many packets should arrive during a specified time frame, you just can’t be guaranteed a steady rate. TReK has implemented queues throughout the system to insure that packets are not lost due to bursts of data, high data rates, or user activity. The size of these queues can be set in the Telemetry Processing application. See the Telemetry Processing Users Guide (TREK-USER-003) for details.

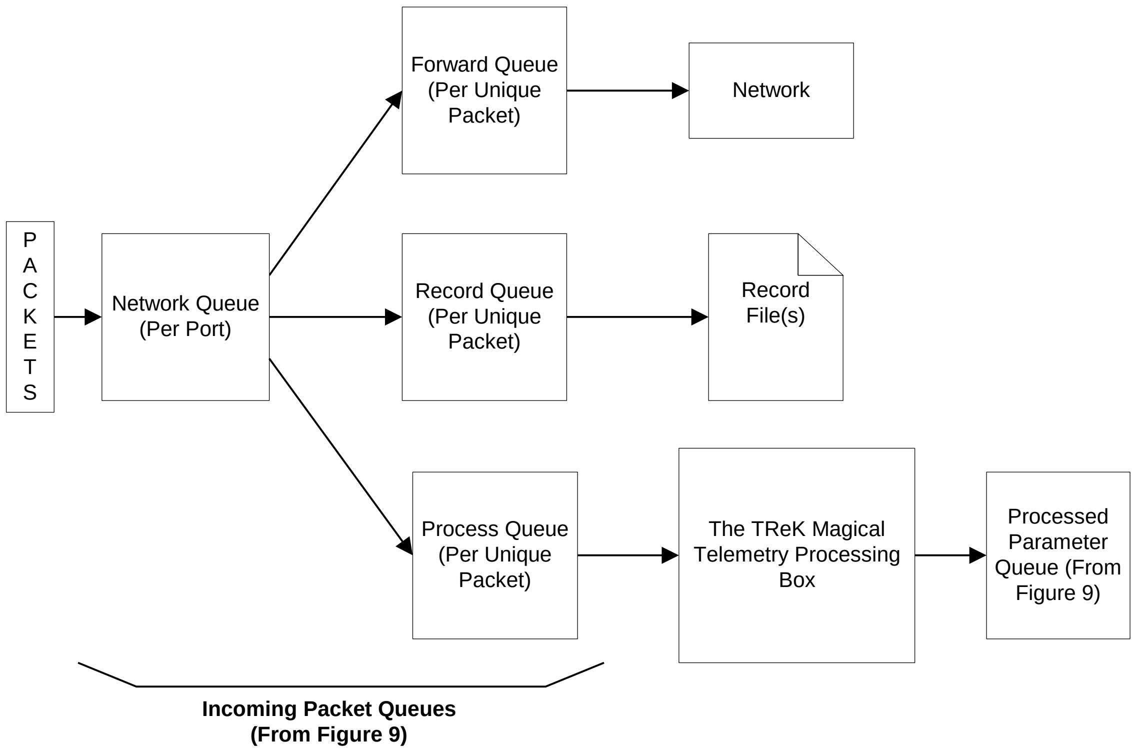

Figure 10 Queue Details

Figure 10 shows all of the queues TReK uses to process your data. When the packet is pulled from the network it is placed in the network packet queue. There is a network packet queue for every port created by the Telemetry Processing application. Once the packet is identified as something TReK should keep the packet is placed in the forward packet queue, record packet queue, or process packet queue as necessary. For example, if you are only recording the data, the packet is only placed in the record packet queue. The forwarding, recording, and processing packet queues are per unique packet stream (APID, data mode, and packet type).

3.4 Forwarding the Packet

Once a packet is identified, the packet can be forwarded to other computers. You have the option of specifying multicast or unicast addresses for forwarding. All data is forwarded using UDP (see Section 2.6.2 for details).

3.5 Recording the Packet

You will also have the option of recording the packets you are receiving. Once a packet is identified as one that TReK should receive, it can be written to the hard disk and you can replay the data at a later time. Be aware that this feature requires enough disk space to save the data. For more information on configuring TReK for processing, recording, and playing back packets, see TReK Telemetry Processing User Guide (TREK-USER-003).

3.6 The Packet Composition Puzzle

The subsets and parameters must be placed in the packet in such a way that software can extract their raw values from the packet. MSFC-STD-1274 Volume 2 defines the different ways a parameter can be placed in a packet. The parameter composition describes how the bits of a parameter are arranged for a sample of a parameter. MSFC-STD-1274 Volume 2 defines three types of parameter composition: typical, multi-syllable, and group. The sample of a parameter can be placed in the packet with one of four sampling composition types: normal, super, counter-dependent, and range dependent.

You can read MSFC-STD-1274 Volume 2 for the details on parameter and sampling composition, but for the purpose of understanding telemetry in TReK you really need to remember only two things. The first is that a parameter may not always appear in each instance of a packet (counter or range dependent sampling). The second is that a parameter may appear more than once in a packet (groups or super sampled). Other than that, the actual layout of a packet doesn't really affect what you can get from the packet.

When TReK first receives the packet it doesn't know anything about it. Once the packet is identified, as described in Section 2.5, TReK can take a binary pattern as shown in Figure 11 and correctly pull out the pieces of data as parameters as shown in Figure 12. These parameters can then be used by your user applications.

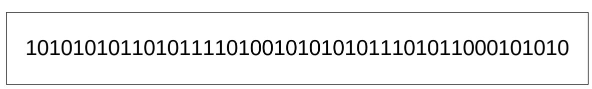

Figure 11 Raw Data in a Packet

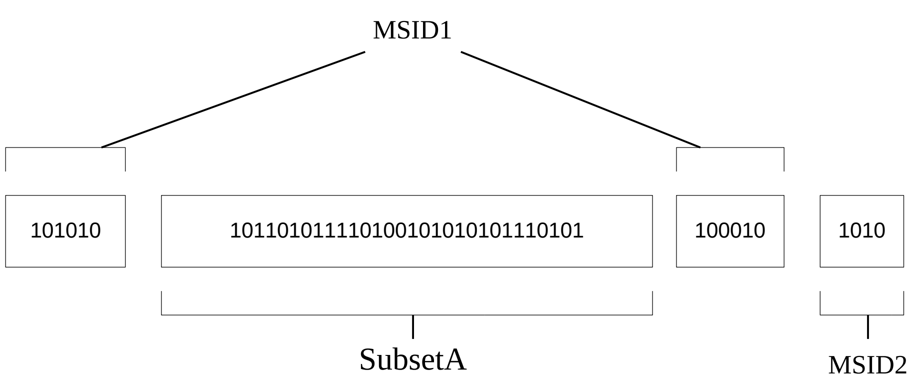

Figure 12 Pieces of the Packet

Figure 12 shows that the raw data is extracted into three parameters. MSID1 has two samples and MSID2 has one sample. SubsetA is extracted and if there is information in the database, can be processed further. (Remember a subset can contain other subsets and parameters.)

3.7 The TReK Database (Part I)

In order for TReK to pull parameters from a packet, the TReK software must know where the parameters are in a packet. The TReK telemetry database contains the information needed to get each parameter from a packet. The information in the TReK database can be built using the database from the POIC. (See TReK Telemetry Database User Guide (TREK-USER-005) for information about how a POIC database is converted into a TReK database.) Each payload team originally entered this information through the Payload Data Library (PDL).

If the packet definition was not entered into PDL, it will not be available for conversion. In this case, the user will have to enter the information into the TReK database using the TReK Telemetry Database application.

The information located in the TReK database is critical for TReK software to be able to extract the data from the packet. Invalid data in the database can prevent TReK from ever identifying packets that are received, pulling the data out correctly, and more.

4. Processing the Data

Once TReK has processed a packet, you can get the data for use with your user products (displays, computations, and scripts). TReK provides a dynamic link library that contains a series of functions you can use to get processed data. See the TReK Telemetry Application Programming Interface (API) Reference Manual (TREK-USER-027) for detailed information on the functions that are available.

This section will describe some of the data that is available from TReK after the data has been processed.

4.1 Raw Data Can Make You Sick

Each parameter that is pulled from a packet is considered raw data. Raw data has no representation other than binary. If that binary pattern represents a value, TReK can give you the value also.

4.2 Conversion

If a raw data pattern has a value, it must be converted to a type that the TReK computer can understand. TReK can convert any of the supported data types from its raw form to a local converted type. Table 1 lists all of the data types supported by TReK and the type of data TReK will convert it to. These data types are the same as are listed in MSFC-STD-1274. Any updates made to that document will be supported by TReK.

Data Type	Converted To
ASCII Character String (SASC)	String
EBCDIC Character String (SEBC)	String
Express ASCII Character String (SEXP)	String
Undefined Byte String (SUND)	Binary Data[1]
Binary Coded Decimal (IBCD)[2]	Unsigned Integer
Discrete Integer (IDIS)	Unsigned Integer
Distended Signed Integer (IDSI)	Signed Integer
Sign and Magnitude Integer (IMAG)	Signed Integer
Undefined Integer (IUND)	Binary Data[1]
Unsigned Integer (IUNS)[3]	Unsigned Integer
Two's Complement Signed Integer (ITWO)	Signed Integer
IEEE Floating Point (FEEE)	Double Precision Floating-Point
IBM Floating Point (FIBM)	Double Precision Floating-Point
INTEL Floating Point (FNTL)	Double Precision Floating-Point
Spacelab Floating Point (FSPL)	Double Precision Floating-Point
VAX Floating Point (FVAX)	Double Precision Floating-Point
Military Floating Point (FMIL)	Double Precision Floating-Point
CCSDS Unsegmented Time (TISS)	TReK Time Type[4]
SSP Data Management System Time (TDMS)	TReK Time Type[4]
ECIO GMT (TECI)	TReK Time Type[4]
ECOS Binary GMT (TECS)	TReK Time Type[4]
EHS Converted Time (TECT)	TReK Time Type[4]
EHS Time Word (TEHS)	TReK Time Type[4]
Earth Receipt Time (TERT)	TReK Time Type[4]
Greenwich Mean Time (TGMT)	TReK Time Type[4]
Orbiter GPC GMT (TGPC)	TReK Time Type[4]
Orbiter OI GMT and MET (TOOI)	TReK Time Type[4]
Time Since Midnight (TTSM)	TReK Time Type[4]
Count Time (TTWO)	TReK Time Type[4]
User Data Summary Time (TUDS)	TReK Time Type[4]

Table 1 TReK Data Types and Converted Types

[1] Some data has no converted value and is always considered raw.
[2] TReK supports up to 10 digit binary coded decimal numbers (40 bits). MSFC-STD-1274 only supports 4 digits (16 bits).
[3] TReK supports 32 bit unsigned integers. MSFC-STD-1274 only supports 31 bits.
[4] The TReK Time Type is explained in section 7.

Table 2 lists the converted types and the equivalent types in each of the TReK User COTS products. Some of the data types are not supported by all of the COTS products. Consult the COTS product help for more information on their representation of these data types. You are free to choose other COTS products. Check the COTS documentation for equivalent data types.

Converted Type	Visual C++ Data Type	Visual Basic Data Type
String	char *	String
Binary Data	unsigned char *	Byte
Signed Integer	long	Long
Unsigned Integer	unsigned long	No equivalent type[1]
Double Precision Floating-Point	double	Double
TReK Time Type	TReK_Time	No equivalent type[2]

Table 2 Data Types for COTS Products

[1] Long can be used for unsigned integer data. However, if the data type is a 32-bit Unsigned Integer the result will be negative if the most significant bit is set.
[2] The time can be retrieved as a string, but not the TReK_Time type.

4.2.1 Byte and Word Ordering

One difference between the data types defined in MSFC-STD-1274 and TReK supported data types are that all of the data types for TReK can have different byte and word ordering. TReK has five different types of byte ordering. They are:

1. Big Endian (Network Order)
2. Byte swapped
3. Word swapped (16 bit words)
4. Byte and word swapped
5. Little Endian (Byte Reversal)

An example should help clear things up. Suppose you are receiving a 64-bit IEEE floating-point number. This data type would fit in 8 bytes. The byte ordering for each of the five possibilities is shown in Figure 13. Unless otherwise specified, the byte ordering is always assumed to be the network order. Data stored on the TReK computer is stored as byte reversal. This information is only important when entering information into PDL or the TReK database. If you get raw data from TReK it will have the same byte ordering as it was sent.

	Byte Numbers:							
Network Order:	0	1	2	3	4	5	6	7
ByteSwapped:	1	0	3	2	5	4	7	6
Word Swapped:	2	3	0	1	6	7	4	5
Byte and Word Swapped:	3	2	1	0	7	6	5	4
Byte Reversal:	7	6	5	4	3	2	1	0

Figure 13 64-bit Byte and Word Ordering

For data types that are byte swapped, the length of the data must be a multiple of 16 bits. For data types that are word swapped, the length must be a multiple of 32 bits.

Note: MSFC-STD-1274 identifies the ITWOW (two's complement word swapped integer) and SASCB (byte swapped ASCII string) data types. These are not listed separately in Table 1, but are supported with the general byte ordering described in this section.

4.2.2 Variable Length Data

MSFC-STD-1274 has a data type for variable length ASCII character string for EXPRESS rack user (SEXP). TReK supports variable length data for all string types (SASC, SEBC, SUND, and SEXP). The only requirement is that the string parameter must be the last parameter in a packet.

4.3 Calibration

A converted data value may not be the final value that you need. TReK supports calibration of parameters that have a data type of integer (except Undefined Integer) or floating point. TReK supports three types of calibration: Polynomial Coefficient, Point Pair, and State Code. Each of the calibration types and the data types that can be calibrated are explained in the following sections.

4.3.1 Polynomial Coefficient Calibration

Polynomial Coefficient Calibration can be performed on all integer (except Discrete Integer and Undefined Integer) and all floating point data types. Polynomial Coefficient Calibration takes the converted value of the parameter and substitutes it into the following polynomial equation where x is the converted value:

$$y = ax^9 + bx^8 + cx^7 + dx^6 + ex^5 + fx^4 + gx^3 + hx^2 + ix + j$$

The constants a through j are defined in the TReK database or Telemetry Processing Application. The calibrated value (y) can be retrieved using the TReK API.

4.3.2 Point Pair Calibration

Point Pair calibration can be performed on all integer data types except Discrete Integer and Undefined Integer. Point Pair Calibration calculates the calibrated value using interpolation on a series of linear line segments as shown in Figure 14.

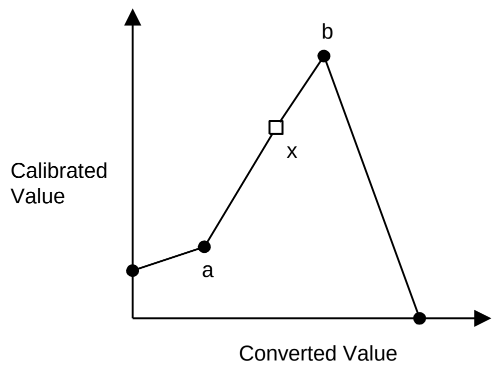

Figure 14 Point Pair Calibration Example

The telemetry software will determine which pair of points the converted value falls between and calculate the calibrated value from the calibrated values associated with the end points. If the converted value of a parameter ($conv_x$) is between points a and b. Then the calibrated value of x (cal_x) is defined in the following equation:

$cal_x = (cal_b – cal_a)(conv_x – conv_a)/(conv_b – conv_a) + cal_a$

Note: If the converted value is less than the minimum converted point pair, the calibrated value will be set to the calibrated value for the minimum converted value. If the converted value is greater than the maximum converted point pair, the calibrated value will be set to the calibrated value for the maximum converted value.

4.3.3 State Code Calibration

State Code Calibration can only be performed on Discrete Integers. The bit pattern of the discrete integer is interpreted as an unsigned integer. This value is checked against the converted ranges provided by the TReK database or the Telemetry Processing application. There is a string value associated with each range. The string associated with the range that the converted value is within is the calibrated value for the parameter. For example, a one bit discrete integer can have two values: 0 and 1. The value 0 can be assigned the state code of “OFF” and 1 can be assigned the state code “ON”.

4.4 Sensing

When you get a parameter’s converted or calibrated value from TReK, you can also have TReK check the data for any errors you have defined. These errors are either limits that the value of a parameter should not exceed or a check to see if the parameter’s value is in an expected state.

4.4.1 Limit Sensing

Limit sensing is the comparison of a value with a set of limits to determine if the value of the parameter is within the limit range. Since the comparison is to a set of limits, the value of the parameter must be convertible to an integer or floating point number. If calibration is defined for a parameter, then the calibrated value is used when checking the limits. If there is no calibration defined in the database, then the converted value is used in the comparison.

There are two sets of limits that are checked. The first check is for a value exceeding the warning limits. Both the high and low limits are checked. If the value is outside the warning range, then the parameter is tagged as out of warning limits. If the value is within the warning limits, then the value of the parameter is checked against the caution limits. If the value is outside of the caution range, then the parameter is tagged as out of caution limits. In addition to the range check, a delta check is performed. The value is compared to the value from the previous packet of data. If the difference in these values is greater than the delta value defined in the database, then the parameter is tagged as having exceeded the delta value. If none of the above checks fails, then the parameter is marked as good data within limits.

The example below shows a set of caution and warning limits and how telemetry values would be evaluated based on these limits.

Warning High Limit: 85
Caution High Limit: 80
Caution Low Limit: 45
Warning Low Limit: 40

- If the calibrated value is 50, then the parameter is within limits
- If the calibrated value is 42, then the parameter has a caution low limit violation
- If the calibrated value is 90, then the parameter has a warning high limit violation

The next example shows 3 consecutive values of a parameter from a packet. If the delta value is 5, then the following would occur:

Value = 5, no delta error since a previous value isn’t available
Value = 7, delta = 2, no error
Value = 13, delta = 6, delta error violation

4.4.2 Expected State Sensing

Expected state sensing is similar to limit sensing. It is the comparison of a state (actually a string) to an expected value. If the value of the parameter is not the same as the expected state, then the data is tagged as having been out of the expected state. Otherwise, the data is tagged as good.

Unlike limit sensing, expected state sensing can only be performed on calibrated data that generates a string (state code calibration). Expected state sensing cannot be performed on converted data.

The example below shows the possible expected states for a parameter and how telemetry values would be evaluated based on the expected state.

Expected State:	On
Other States:	Off
	Paused

- If the calibrated value is On, then the data is in the expected state
- If the calibrated value is Paused, then the data is out of expected state

4.5 Monitoring

One way to keep up with the sensing violations is to monitor a parameter. Monitoring a parameter involves TReK checking every sample of a parameter for violations. A tolerance can be set to flag only the n^{th} consecutive sample in violation. If the tolerance is exceeded then a message will be written to notify the user of the violation. The message consists of a time stamp, a TReK defined message, and an optional user defined message. The TReK defined message contains the parameter name, the type of violation (e.g., Expected State), and the value that caused the message to be generated.

4.5.1 Limit Monitoring

Limit monitoring is analogous to limit sensing. When one of the caution or warning limits occurs more than the tolerance allows a message will be generated. Once a message is generated another message will not occur until the limits change state. For example, if a message is generated for a caution low violation, another message won't be generated until enough consecutive samples put it into another state. If a parameter goes back within limits or goes to a warning low violation, a message will be generated.

Unlike caution and warning limits, a delta limit monitoring message is generated for every violation (implied tolerance of zero). It is possible to receive two messages from a single sample: a delta message and a limit message.

4.5.2 Expected State Monitoring

Expected state monitoring is analogous to expected state sensing. If the expected state is different from the actual state for more than the tolerance allows a message will be generated.

4.6 Switching

TReK allows you to define multiple sets of calibration and sensing information. Each parameter will have a default set that can be used, but there is also a way for your telemetry to tell TReK which set to use. The concept of switching between sets in TReK is similar to that of the POIC. A parameter's value can be used to switch between different calibration and sensing sets. There are two types of switching on TReK: range and state code.

4.6.1 Range Switching

Range switching requires switch parameter values that evaluate to a number. If a calibrated value is available then the calibrated value of the switch parameter is used. If no calibration is defined then the converted value is used. TReK checks the value against the ranges that are defined. The set number that corresponds to the range is used for the calibration or sensing. If the switch parameter's value does not fall with one of the ranges, then the default set number is used.

4.6.2 State Code Switching

Unlike range switching, state code switching requires that the value be calibrated. The calibrated value is a string (state code calibration). The switch parameter's state code is checked to find which set to use. If the switch parameter's state code does not exist, then the default set number is used.

4.7 The TReK Database (Part II)

Just as there is information in the database that allows TReK to pull parameters from a packet, there is information on how to interpret and process the parameters within a packet. The database will tell the TReK telemetry software what the data type of a parameter is and the length of the parameter, so that TReK can convert the parameter to the local computer native data type.

In addition, the calibration, sensing, monitoring, and switching information is contained in the database. When you initialize a packet for processing, TReK will get all of this information out of the database in order to correctly convert, calibrate, sense, and monitor your data. Any changes made to the database can also affect the conversion, calibration, and sensing of your data. It is possible to make changes that could cause TReK to incorrectly process your data, so be careful when you change the database.

4.8 Preprocessed Data (GSE and CDP) on TReK

As was mentioned earlier, TReK considers GSE Packets and Custom Data Packets to have preprocessed data. The format of the data in the CCSDS packet data zone contains more information than just the value of the parameter. For each sample of a parameter there is a status character and an overall status that TReK must process. The overall status lets TReK know if there are any problems associated with the parameter when it was processed by the POIC. By default TReK does not process parameters that contain errors. However, you can change this behavior in the Telemetry Processing application to allow selected or all errors through. If you choose to allow errors, the TReK Application Programming Interface (API) provides a function that you can use to retrieve the error (overall status) associated with a particular parameter.

For more information on the errors that can occur in a GSE packet or Custom Data Packet, see the Payload to Generic User IDD (SSP-50305). For more information on the GetOverallStatus function in the TReK API, see the TReK Telemetry API Reference Manual (TREK-USER-027). For more information on how to configure TReK to allow some or all errors for GSE packets or Custom Data Packet, see the Telemetry Processing Users Guide (TREK-USER-003). Some additional information on customizing this capability is also available in Section 4.9.

Section 5.6 discusses what happens to the status character that is sent for the GSE or CDP parameters.

4.9 Error Control for Preprocessed Parameters

Even though TReK doesn’t allow any errors through for GSE and Custom Data Packet parameters by default, you can change this behavior. Just as you can specify what calibrators or sensors are used for a parameter, you can specify an error control object for each parameter. When you set up your TReK system to process a GSE or Custom Data Packet, a default error control object is created and used for each parameter in the CCSDS data zone of the GSE or Custom Data Packet. You can modify the default error control to allow errors or create a new error control object. For more information on creating these objects and configuring parameters, see the Telemetry Processing Users Guide (TREK-USER-003).

5. Using the Data

This section covers some topics associated with telemetry processing and TReK. The topics in the following sections will help you understand some of the other TReK documentation and concepts.

5.1 Identifying a Parameter

Earlier you learned about what makes a packet unique and how TReK identifies the packet. To uniquely identify a parameter, you are required to identify the parameter name (MSID, subset ID, or packet ID) along with the data mode and packet type. This is the minimum amount of information that TReK needs to identify a parameter. In addition, you can optionally request data from a specific packet stream (identified by APID). This will allow you to get a parameter’s value from a specific packet stream.

5.2 Newest vs. Next Parameter

TReK allows you to get the newest or next parameter value for all types of data (converted, calibrated, etc.). If you want to get only the latest value (i.e., the value from the last packet processed by TReK), you will want to use one of the GetNewest…Value functions provided by the TReK API. If you want to get the value of the parameter from every packet that arrives on TReK you will use one of the GetNext…Value functions provided by the API. For more information on the functions available in the TReK API see the TReK Telemetry API Reference Manual (TREK-USER-027).

Section 3.1 described a processed parameter queue. Figure 15 shows how the newest and next parameters are related to the processed data queue.

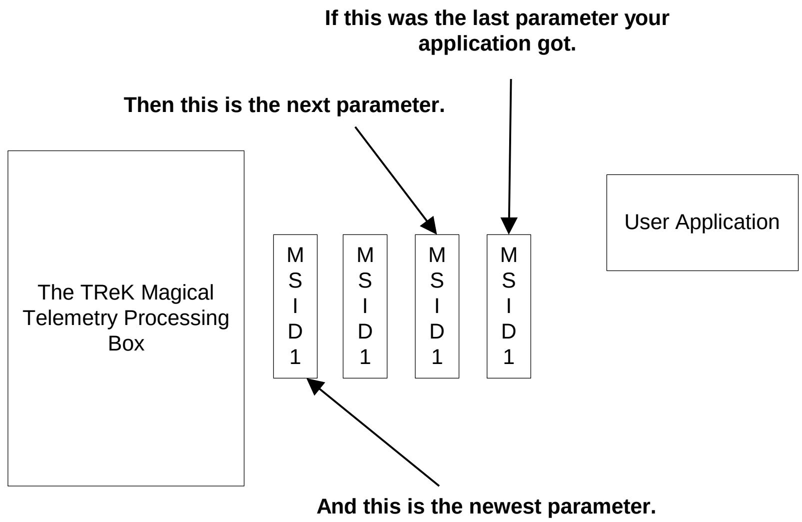

Figure 15 Processed Parameter Queue

5.3 The TReK Telemetry Token

The TReK telemetry token is a value that the TReK telemetry processing system uses to determine which parameter to give you next and whether or not you have already received that instance of the parameter. If you are asking for the next value of a parameter, TReK uses the token to determine the last value you received. If more data has arrived since you last requested this parameter, TReK will return the one received immediately after the last one you got. If no new data has arrived, TReK will return the old value and set the TReK freshness status character to stale (see section 5.6 for details). If you are asking for the newest value of a parameter, TReK just uses the token to determine whether or not the data is stale.

In addition to the above uses, TReK will use the token internally to the telemetry processing system to perform switching (see section 7 for details) and sensing. You only have to initialize the value of the token. As matter of fact, changing the value of the token afterwards can cause unexpected results.

5.4 Getting Multiple Samples of a Parameter

As was mentioned before, some parameters may appear in a packet multiple times. For example, in Figure 16 parameter A appears three times and parameters B and C appear once each.

Header	MSID A	MSID A	MSID A	MSID B	MSID C

Figure 16 Packet With Multiple Parameters

TReK allows you to request multiple samples of a parameter from a packet. If you request more samples of a parameter than are available at that time from TReK, you will get the number of samples available and will be notified that there were fewer samples available than you requested. If you ask for fewer samples than are available, you will not know that more data was available. So be sure to ask for what you want.

You can use the TReK Telemetry Database application to query the database to find out the maximum number of samples for a parameter that could ever appear in a packet. See the TReK Telemetry Database User Guide (TREK-USER-005) for more information. You can also get this information from the Telemetry Processing application. See the Telemetry Processing User Guide (TREK-USER-003) for more information.

In addition to using the database, there is a call available in the TReK Application Programming Interface (API) that allows you to get information about a parameter. This information includes the maximum number of samples for a parameter. See the TReK Telemetry Application Programming Interface Reference Manual (TREK-USER-027) for more information on the GetParameterInformation function.

5.5 Pseudo Telemetry Parameters Are People Too

If you need to share data from one user product with other user products, you can use the pseudo telemetry parameter capability in TReK. For example, if you perform your own calibration of data you may want to share it with other applications. Pseudo telemetry parameters can be created by any application. Once they are created, any other application can get the value of the pseudo telemetry parameter. This mechanism provides a good means of sharing data between user products.

Unlike telemetry parameters, pseudo telemetry parameters do not exist in the database. They are created by a user application, such as a computation. Please see TREK-USER-

027 for detailed information about the pseudo telemetry functions and how to use them. The How To Process Your Own Data Tutorial (TREK-USER-010) also has an example of creating and using pseudo telemetry parameters.

5.6 The TReK Status String

The status string is a TReK specific feature of telemetry processing. If you are familiar with the POIC then you may know that the POIC provides a single character to indicate the status of a parameter (e.g., 'H' is used to indicate that the data has exceeded the warning high limit). If a parameter is out of range and has an old value (stale data), you will be informed of only one of these conditions based on a priority scheme implemented within the POIC.

TReK has taken the concept of a status character and expanded it into a status string. This minimizes the possible conflicts, which are resolved by assigning a priority to each status character. Each of the characters within the string indicates a status. There are three values in the TReK status string. By default each of these characters is separated by a '/' character. This separator character can be changed in the Telemetry Processing application. See the TReK Telemetry Processing User Guide (TREK-USER-003) for details.

The first character is the POIC status character received with any data processed by the POIC (e.g., Custom Data Packets). This status character is not modified by TReK, but is placed directly in the status string. If you are receiving data from a source other than the POIC, this character will always be blank. (A blank is considered good data.) See SSP-50305 POIC To Generic User IDD for all POIC status characters.

The second non-separator character in the TReK status string is the TReK freshness character. This character indicates whether or not this value has already been retrieved by the calling application. If the value has been retrieved before, then the TReK freshness character is set to 'S' to indicate stale data. Otherwise, the TReK freshness character is set to blank which indicates that this is the first time this instance of the data value has been retrieved by that particular application.

The final non-separator character in the TReK status string is the TReK processing status character. This character indicates the status of processing on TReK for the requested parameter. See Table 3 for a list of all of the TReK status characters. The last three columns of the table indicate whether the status character can appear in the POIC, TReK Freshness, or TReK Processing statuses.

Status Character	Definition	EHS[1]	TReK Freshness	TReK Processing
‘H’	Warning High Limit Exceeded	√		√
‘L’	Warning Low Limit Exceeded	√		√
‘>’	Caution High Limit Exceeded	√		√
‘<’	Caution Low Limit Exceeded	√		√
‘S’	Stale data. This is the same data you got back last time.	√	√	
‘E’	Out of expected state.	√		√
‘ ‘ (Blank)	Good data	√	√	√
‘l’	Sensing information not available.			√
‘b’	Delta Limit Exceeded			√
‘y’	Delta check could not be performed on parameter because a sample from the previous packet could not be retrieved.			√
‘&’	Calibration information not available.			√
‘u’	Warning High and Delta Error Violation			√
‘q’	Warning Low and Delta Error Violation			√
‘s’	Caution High and Delta Error Violation.			√
‘r’	Caution Low and Delta Error Violation.			√
‘c’	Calibration Error. (Most likely the default set number does not exist.)	√		√
‘g’	Sensing Error.			√

Table 3 TReK Status Characters

[1] For a complete list of POIC status characters and their priority, see SSP-50305 POIC to Generic User IDD. This column is only applicable for GSE or CDP data, and is otherwise blank.

6. TReK Telemetry Processing Philosophy

Before going any further it is important to explain a major difference in Telemetry Processing between TReK and the POIC. The calibration, sensing, monitoring, and switching information described in Section 4 can be shared in TReK. The sharing of information can make it much easier to make mass changes.

6.1 POIC (One-to-One Correlation)

The POIC requires that each MSID (parameter) have its own calibration, sensing, monitoring (POIC uses the term exception monitoring), and switching defined. What ends up happening is that there is a one-to-one correlation for a parameter and its calibration, etc. as is shown in Figure 17.

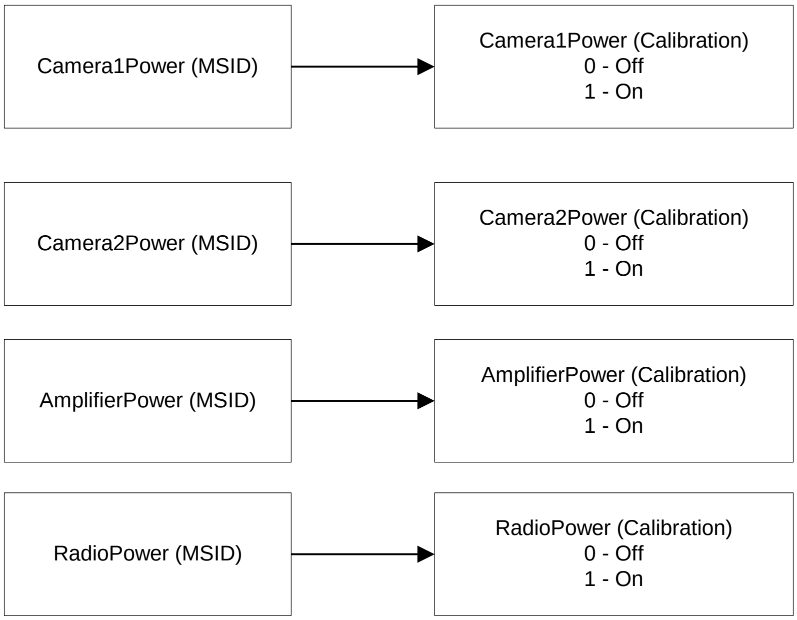

Figure 17 POIC MSIDs and Calibration Correlation

6.2 TReK (Sharing Data)

The name for a parameter, its calibration, sensing, monitoring, and switching information is always the same in the POIC. This makes it easy to correlate the calibration to a particular measurement, but is limiting since data may have to be defined multiple times. TReK allows you to share the calibration, sensing, monitoring, and switching information among as many parameters as you like. Figure 17 shows four parameters, each having separate calibration information. However, the information for each is really the same. Figure 18 shows how the same information can be represented in TReK telemetry processing.

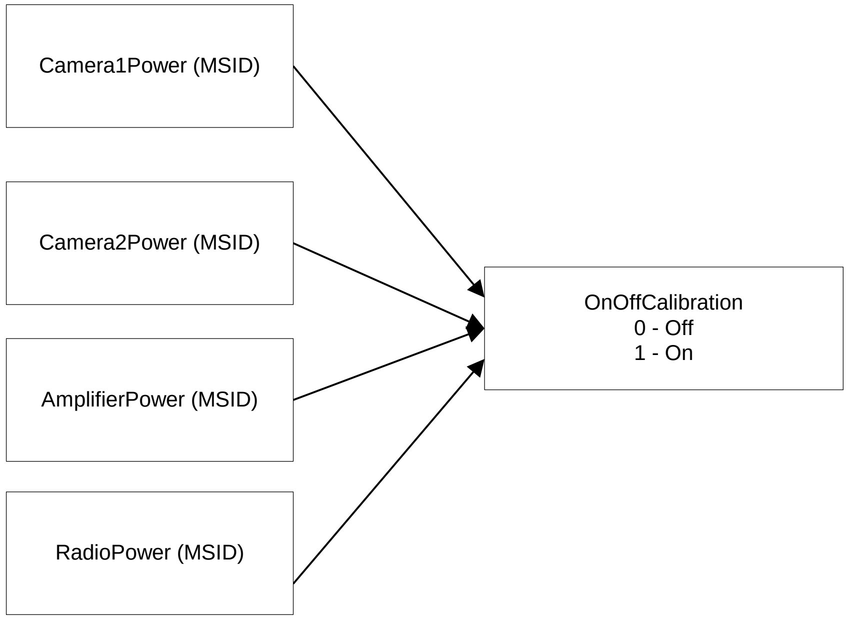

Figure 18 TReK Sharing Calibration

Now the four parameters each share common information for calibration. If the calibration information needed to be changed so that 0 represented On and 1 represented Off, then it would only have to be changed in one place.

It is still possible to have the same one-to-one correlation that the POIC has. You can define separate calibration information for each of the parameters if you wish. You could use the same name as the parameter or a different one. When a POIC database is converted to a TReK database, TReK preserves the one-to-one correlation that existed in the POIC database. You will have to manually change the TReK database to share items that originated from a POIC database.

6.3 TReK Terminology

TReK has a few terms that need explaining at this point. When you go through the Telemetry Processing application you will see the terms Calibrator, Sensor, Monitor, and Switch. These terms correspond to the calibration, sensing, monitoring, and switch information respectively. For example, if you wanted to define some new calibration information you could use the Telemetry Processing application to create a new calibrator. This calibrator could then be used with zero or more parameters. These generic terms will be used throughout the TReK documentation.

7. Questions/Answers

The previous sections have given an end-to-end telemetry processing overview. This section is formatted as questions and answers. If this is the first time the information has appeared in the document, then the question will be marked with 

.

What is a raw value?

The raw value of a parameter is just binary data. This data cannot necessarily be converted by TReK to a form that can be interpreted by the computer. Sometimes this is because the raw data is not intended to be interpreted by the computer. Other times TReK may not support the conversion of the type of data that is represented by the raw value. In the second case, you can write your own processing code to convert the parameter to one of the converted types that TReK supports. For details on converting your own data, see TREK-USER-010 TReK How To Process Your Own Data Tutorial.

What is conversion?

The data that is received by TReK is in its raw form. In order for the computer to be able to interpret the data correctly the data must be converted into a form that the computer will understand. A parameter could represent a floating-point number in its raw form. The TReK telemetry processing system will convert the binary pattern into the internal representation of a floating-point number for the computer for known data types (see Table 1 for supported data types).

What is calibration?

Calibration is best described using an example. Your experiment may have a temperature sensor. This sensor's value is transmitted in your CCSDS packet and is converted by TReK. However, this value needs to be calibrated to interpret the data as degrees Celsius instead of the converted value of degrees Kelvin. The calibration data you entered through PDL, directly into the TReK database, or directly into the Telemetry Processing application is used to perform this calculation.

What is Limit Sensing?

Limit sensing is the comparison of a value with a set of limits to determine if the value of the parameter is within the limit range. Since the comparison is to a set of limits, the value of the parameter must be convertible to an integer or floating point number. If calibration is defined for a parameter, then the calibrated value is used when checking the limits. If there is no calibration defined in the database, then the converted value is used in the comparison.

There are two sets of limits that are checked. The first check is for a value exceeding the warning limits. Both the high and low limits are checked. If the value is outside the warning range, then the parameter is tagged as "out of warning limits." If the value is within the warning limits, then the value of the parameter is checked against the caution limits. If the value is outside of the caution range, then the parameter is tagged as "out of caution limits." In addition to the range check, a delta check is performed. The value is compared to the value from the previous packet of data. If the difference in these values is greater than the delta value defined in the database, then the parameter is tagged as having exceeded the delta value. If none of the above checks fails, then the parameter is marked as good data within limits.

What is expected state sensing?

Expected state sensing is similar to limit sensing. It is the comparison of a state (actually a string) to an expected value. If the value of the parameter is not the same as the expected state, then the data is tagged as having been out of the expected state. Otherwise, the data is tagged as good.

Unlike limit sensing, expected state sensing can only be performed on calibrated data that generates a string (state code calibration). Expected state sensing cannot be performed on converted data.

What is switching?

You can define multiple sets of calibration and sensing information in the TReK database or through the Telemetry Processing application. In order to use these different sets of data you can define a parameter that can be used to switch between the sets of data. The parameter is known as a switch parameter. The value of the switch parameter is checked to see which calibration or sensing set should be used.

What is the TReK time type? NEW

The TReK time type is a special type used for the conversion of time data on TReK. It is composed of the ANSI C time structure (struct tm) and other fields specific to TReK. The C structure definition of the TReK time type is:

```
typedef struct
{
      struct tm tm_time;
      unsigned long milliseconds;
      unsigned long microseconds;
      unsigned short new_data;
      unsigned short hold;
      unsigned short sign;
} TReK_Time;
```

The tm_time value can be used with many different C functions (see online help for any C or C++ development tools for the different functions). The new_data, hold, and sign values are specific to the EHS Time Word (TEHS) data type.

The TReK_Time type is available in the trek.h file included in the TReK installation directory. See the TReK Telemetry Application Programming Interface (API) Reference Manual (TREK-USER-027) for detailed information on the functions that are available for retrieving time.

What is the status string? How does it relate to the EHS status characters?

The status string is a TReK specific feature of telemetry processing. If you are familiar with the POIC then you may know that the POIC provides a single character to indicate the status of a parameter (e.g., ‘H’ is used to indicate that the data has exceeded the warning high limit). If a parameter is out of range and has an old value (stale data), you will be informed of only one of these conditions based on a priority scheme implemented within the POIC.

TReK has taken the concept of a status character and expanded it into a status string. This minimizes the possible conflicts, which are resolved by assigning a priority to each status character. Each of the characters within the string indicates a status. There are three values in the TReK status string. By default each of these characters is separated by a ‘/’ character. This separator character can be changed in the Telemetry Processing application. See the TReK Telemetry Processing User Guide (TREK-USER-003) for details.

The first character is the POIC status character received with any data processed by the POIC (e.g., Custom Data Packets). This status character is not modified by TReK, but is placed directly in the status string. If you are receiving data from a source other than the POIC, this character will always be blank. (A blank is considered good data.) See SSP-50305 POIC To Generic User IDD for all POIC status characters.

The second non-separator character in the TReK status string is the TReK freshness character. This character indicates whether or not this value has already been retrieved by the calling application. If the value has been retrieved before, then the TReK freshness character is set to ‘S’ to indicate stale data. Otherwise, the TReK freshness character is set to blank which indicates that this is the first time this instance of the data value has been retrieved.

The final non-separator character in the TReK status string is the TReK processing status character. This character indicates the status of processing on TReK for the requested parameter. See Table 3 for a list of all of the TReK status characters. The last three columns of the table indicate whether the status character can appear in the POIC, TReK Freshness, or TReK Processing statuses.

What does the status string look like? NEW

The status string is a NULL terminated string. Assume you request a PDSS parameter’s calibrated value and request to perform limit sensing on the parameter and “ / /H” was returned as the status string. The first character is blank (good data) since this data is not from a Custom Data Packet or GSE Packet. The second character is blank which indicates this is the first time you retrieved the data value (i.e., the value is not stale). The third character is ‘H’ which indicates that the parameter has exceeded the warning high limit. The ‘/’ is a separator character and can be set in the Telemetry Processing application.

What are pseudo telemetry parameters?
A pseudo telemetry parameter is a value that is calculated by an application and shared with other applications through the TReK telemetry processing system. The pseudo telemetry parameter can be calculated from telemetry data or other data.

Unlike telemetry parameters, pseudo telemetry parameters do not exist in the database. They are created by a user application, such as a computation.

What is the difference between Newest and Next?
The newest data in TReK is the last value processed for that parameter (see Figure 15). You may want to build a display that only shows the latest values for a parameter. In this case, you should use the newest value functions.

The next data in TReK is the value immediately after the one you last received. If you hadn't requested the data before, you would get the oldest data available. By using the next value functions you can get every instance of a parameter that is processed by TReK.

What is the TReK telemetry token and why do I need it?
The TReK telemetry token is a value that the TReK telemetry processing system uses to determine which parameter to give you next and whether or not you have received that instance of the parameter before. If you are asking for the Next value of a parameter, TReK uses the token to determine the last value you received. If more data has arrived since you last requested this parameter, TReK will return the one received immediately after the last one you got. If no new data has arrived, TReK will return you the old value and set the TReK freshness status character to stale. If you are asking for the Newest value of a parameter, TReK just uses the token to determine whether or not the data is stale.

In addition to the above uses, TReK will use the token internally to the telemetry processing system to perform switching and sensing. You never have to set the value of the token. As matter of fact, changing the value of the token can cause unexpected results.

How can I get a parameter from a specific packet?
TReK requires three items to uniquely identify a processed parameter: a name (parameter, subset ID, or packet ID), a data mode (e.g., Real Time), and a stream type (e.g., PDSS Payload). If a parameter is contained in multiple packets you will receive data from all of those packets by default. However, if you only want to get the data from a particular packet, then you can specify the APID of the packet when requesting the data value. (See TREK-USER-027 TReK Telemetry Application Programming Interface Reference Manual for the different functions you can use.)

What is data mode and why do I need it?
Data mode is an attribute in the EHS primary header. It is used to indicate the state of the data being received. EHS has defined 15 different data modes that can be sent. It is

possible to be receiving a packet in multiple data modes. You would not want to mix and match data from a real time packet and a playback packet. In order to prevent this from ever occurring, TReK always requires you to specify the data mode when requesting a parameter. The data mode for Suitcase Simulator packets is “none”.

The POIC allows me to set the data mode for a display, computation, etc. Why can’t I do that for TReK?

You can. Since you are building your user products (displays, comps, and scripts) with COTS products you could set up a display to only run for a single data mode. (See the CyclicTwo example in the TReK installation to see an example of a single data mode display.) However, since TReK allows you to specify the data mode for each parameter you request, it is possible to build a user product that has more than one data mode available.

What does it take to uniquely specify a parameter?

There are three items that TReK uses to determine the uniqueness of a parameter. They are parameter name (e.g., MSID name), data mode, and packet type (e.g., PDSS Payload). In addition you can use the application process id (APID) of the CCSDS packet to ask for data from a particular packet. If a particular APID is not requested, then TReK returns the data without regard to APID.

How do the sampling and composition types defined in MSFC-STD-1274 Volume 2 relate to what data I get?

There are really only two things to remember about the sampling and composition types in MSFC-STD-1274 as it relates to your data. The first is that some parameters may appear more than one time within a packet (Super and Group parameters). If a parameter appears more than once, then you can select to retrieve any number of parameters from an instance of a packet. The second thing to remember is that some data does not appear in each instance of a packet (Counter and Range dependent parameters).

How are the data types defined in MSFC-STD-1274 Volume 2 interpreted on my computer?

Each data type in MSFC-STD-1274 is converted to a representation that the TReK computer can understand. All of the data types in MSFC-STD-1274 can be converted to six local computer types. These types are String, Binary, Unsigned Integer, Integer, Double Precision Floating-Point, and Time. Table 1 lists all of the data types found in MSFC-STD-1274 and their converted types. Table 2 lists all of the converted types and the equivalent data types to use in each of the approved user products. Remember that TReK also supports different byte ordering for all data types. See section 4.2 for details.

How many databases can I use with TReK?

TReK does not impose a limit on the number of different databases you can use. You will specify a database for each telemetry stream (APID, data mode, and packet type) you wish to have TReK process. This database can be different for each stream if you wish.

If I get a raw value of a data type defined to have byte and word swapping what do I get?

The data you get will be the exact pattern that was received in the packet. No processing is performed on the data when requested as raw. If the hex pattern was A4B3787D in the packet, then you will get the exact pattern back in a Get…RawValue() call. TReK only considers the byte order when performing conversion and calibration.

How much data does TReK keep? Can I change it?

The Processed Packet Queue shown in Figure 9 can be configured to as much data as you want. TReK defaults to keeping the last 5 packets worth of data. Once the sixth packet arrived, the first packet would no longer be available in memory. You can change this default through the Telemetry Processing application.

Am I guaranteed to receive all my data?

The answer really depends on what type of data you are receiving, or more correctly, what protocol is used to transmit it. If the data is sent using TCP (see section 2.6.1) then a connection between the sending computer and the receiving computer (TReK) is made. This allows the sending computer to know that the receiving computer actually got the data. TCP will prevent packet loss and preserve packet ordering. The only data transmitted via TCP at this time is Custom Data Packets.

PDSS Payload Packets, PDSS UDSM Packets, Suitcase Simulator Packets, and GSE Packets are sent via UDP. Since a connection is not made, the sending computer doesn't know that the receiving computer actually got the packet. In addition, it is possible to receive the data in a different order than it was sent. The advantage of UDP is that this method of transmission also allows for higher data rates since the sending computer doesn't have to wait to see if a packet needs to be retransmitted. For details on the data rates supported by TReK see the TReK Hardware Recommendations and Performance Results Document (TREK-USER-013).

The different queues described in section 3.3 can be increased in size to help prevent data loss with high data rates.

Can I distribute my packets to different machines?

Yes. TReK now has the capability to forward the packets it receives to other computers (the other computers do not have to have TReK installed). Using this capability you could set up a computer with TReK software to receive your PDSS packets, record them to disk, and forward them to a different computer. This "front end" computer would only be responsible for a few activities. The other computers (you can forward to multiple IP addresses) would then do the other processing using TReK software or, if needed, custom-built software. The performance tests described in the TReK Hardware Recommendations and Performance Results Document (TReK-USER-013) describe a single TReK system. Tests for multiple TReK computers working together were not conducted.

When a POIC database is converted into a TReK database does it look for elements that can be shared?

No. The one-to-one correlation of measurements, calibrators, etc. is preserved by TReK during the convert process. The user is free to make changes to the database after the convert to share common items. However, this is not recommended.

How do I know when a packet arrives?

The Telemetry Processing application uses colors to indicate that packets are arriving as expected. You can look in the Telemetry Processing Users Guide (TREK-USER-003) for more information on the colors and their meaning.

You can also use the TReK API function GetPacketArrivalEventName to setup your programs to allow TReK to send you a notification when a packet arrives. See the TReK Telemetry API Reference Manual (TREK-USER-027) for more information.

Can I use multiple network cards with a TReK PC?

Yes. You will have to specify which network card to use when setting up TReK to process data. See the TReK Telemetry Processing Users Guide (TREK-USER-003) for information on how to set TReK up to receive telemetry data.

Are the names in TReK case sensitive?

No. Names for items such as calibrators and sensors are not case-sensitive. The calibrator names “SOMETHING” and “Something” are equivalent. However, you could use the name “something” for both a calibrator and a sensor.

How much data can TReK process? What computer should I buy?

The latest information on TReK performance and hardware/software requirements can be found on the TReK web site (http://payloads.msfc.nasa.gov/trek).

Appendix A Glossary

Note: This Glossary is global to all TReK documentation. All entries listed may not be referenced within this document.

Application Programming Interface (API)	A set of functions used by an application program to provide access to a system's capabilities.
Application Process Identifier (APID)	An 11-bit field in the CCSDS primary packet header that identifies the source-destination pair for ISS packets. The type bit in the primary header tells you whether the APID is a payload or system source-destination.
Calibration	The transformation of a parameter to a desired physical unit or text state code.
Communications Outage Recorder	System that captures and stores payload science, health and status, and ancillary data during TDRSS zone of exclusion.
Consultative Committee for Space Data Systems (CCSDS) format	Data formatted in accordance with recommendations or standards of the CCSDS.
Consultative Committee for Space Data Systems (CCSDS) packet	A source packet comprised of a 6-octet CCSDS defined primary header followed by an optional secondary header and source data, which together may not exceed 65535 octets.
Conversion	Transformation of downlinked spacecraft data types to ground system platform data types.
Custom Data Packet	A packet containing a subset of parameters that can be selected by the user at the time of request.
Cyclic Display Update Mode	A continuous update of parameters for a particular display.
Decommutation (Decom)	Extraction of a parameter from telemetry.
Discrete Values	Telemetry values that have states (e.g., on or off).

Dump	During periods when communications with the spacecraft are unavailable, data is recorded onboard and played back during the next period when communications resume. This data, as it is being recorded onboard, is encoded with an onboard embedded time and is referred to as dump data.
Enhanced HOSC System (EHS)	Upgraded support capabilities of the HOSC systems to provide multi-functional support for multiple projects. It incorporates all systems required to perform data acquisition and distribution, telemetry processing, command services, database services, mission support services, and system monitor and control services.
Exception Monitoring	A background process capable of continuously monitoring selected parameters for Limit or Expected State violations. Violation notification is provided through a text message.
Expected State Sensing	Process of detecting a text state code generator in an off-nominal state.
EXPRESS	An EXPRESS Rack is a standardized payload rack system that transports, stores and supports experiments aboard the International Space Station. EXPRESS stands for EXpedite the PRocessing of Experiments to the Space Station.
File transfer protocol (ftp)	Protocol to deliver file-structured information from one host to another.
Flight ancillary data	A set of selected core system data and payload health and status data collected by the USOS Payload MDM, used by experimenters to interpret payload experiment results.

Grayed out	Refers to a menu item that has been made insensitive, which is visually shown by making the menu text gray rather than black. Items that are grayed out are not currently available.
Greenwich Mean Time (GMT)	The solar time for the meridian passing through Greenwich, England. It is used as a basis for calculating time throughout most of the world.
Ground ancillary data	A set of selected core system data and payload health and status data collected by the POIC, which is used by experimenters to interpret payload experiment results. Ground Ancillary Data can also contain computed parameters (pseudos).
Ground receipt time	Time of packet origination. The time from the IRIG-B time signal received.
Ground Support Equipment (GSE)	GSE refers to equipment that is brought in by the user (i.e. equipment that is not provided by the POIC).
Ground Support Equipment Packet	A CCSDS Packet that contains data extracted from any of the data processed by the Supporting Facility and the format of the packet is defined in the Supporting Facility's telemetry database.
Huntsville Operations Support Center (HOSC)	A facility located at the Marshall Space Flight Center (MSFC) that provides scientists and engineers the tools necessary for monitoring, commanding, and controlling various elements of space vehicle, payload, and science experiments. Support consists of real-time operations planning and analysis, inter- and intra-center ground operations coordination, facility and data system resource planning and scheduling, data systems monitor and control operations, and data flow coordination.

IMAQ ASCII	A packet type that was added to TReK to support a very specific application related to NASA's Return to Flight activities. It is not applicable to ISS. It is used to interface with an infrared camera that communicates via ASCII data.
Limit Sensing	Process of detecting caution and warning conditions for a parameter with a numerical value.
Line Outage Recorder Playback	A capability provided by White Sands Complex (WSC) to play back tapes generated at WSC during ground system communication outages.
Measurement Stimulus Identifier (MSID)	Equivalent to a parameter.
Monitoring	A parameter value is checked for sensing violations. A message is generated if the value is out of limits or out of an expected state.
Parameter	TReK uses the generic term parameter to mean any piece of data within a packet. Sometimes called a measurement or MSID in POIC terminology.
Payload Data Library (PDL)	An application that provides the interface for the user to specify which capabilities and requirements are needed to command and control his payload.
Payload Data Services Systems (PDSS)	The data distribution system for ISS. Able to route data based upon user to any of a number of destinations.
Payload Health and Status Data	Information originating at a payload that reveals the payload's operational condition, resource usage, and its safety/anomaly conditions that could result in damage to the payload, its environment or the crew.
Payload Operations Integration Center (POIC)	Manages the execution of on-orbit ISS payloads and payload support systems in coordination/unison with distributed International Partner Payload Control Centers, Telescience Support Centers (TSC's) and payload-unique remote facilities.

Payload Rack Checkout Unit (PRCU)	The Payload Rack Checkout Unit is used to verify payload to International Space Station interfaces for U.S. Payloads.
Playback	Data retrieved from some recording medium and transmitted to one or more users.
Pseudo Telemetry (pseudo data)	Values that are created from calculations instead of directly transported telemetry data. This pseudo data can be created from computations or scripts and can be displayed on the local PC.
Remotely Generated Command	A command sent by a remote user whose content is in a raw bit pattern format. The commands differ from predefined or modifiable commands in that the content is not stored in the POIC Project Command Database (PCDB).
Science data	Sensor or computational data generated by payloads for the purpose of conducting scientific experiments.
Subset	A collection of parameters from the total parameter set that is bounded as an integer number of octets but does not constitute the packet itself. A mini-packet.
Super sampled	A parameter is super sampled if it occurs more than once in a packet.
Swap Type	A flag in the Parameter Table of the TReK database that indicates if the specified datatype is byte swapped (B), word swapped (W), byte and word swapped (X), byte reversal (R), word reversal (V) or has no swapping (N).
Switching	A parameter's value can be used to switch between different calibration and sensing sets. There are two types of switching on TReK: range and state code.

Transmission Control Protocol (TCP)	TCP is a connection-oriented protocol that guarantees delivery of data.
Transmission Control Protocol (TCP) Client	A TCP Client initiates the TCP connection to connect to the other party.
Transmission Control Protocol (TCP) Server	A TCP Server waits for (and accepts connections from) the other party.
Telemetry	Transmission of data collected form a source in space to a ground support facility. Telemetry is downlink only.
Telescience Support Center (TSC)	A TSC is a NASA funded facility that provides the capability to plan and operate on-orbit facility class payloads and experiments, other payloads and experiments, and instruments.
User Application	Any end-user developed software program that uses the TReK Application Programming Interface software. Used synonymously with User Product.
User Data Summary Message (UDSM)	Packet type sent by PDSS that contains information on the number of packets sent during a given time frame for a PDSS Payload packet. For details on UDSM packets, see the POIC to Generic User IDD (SSP-50305).
Uplink format	The bit pattern of the command or file uplinked.
User Datagram Protocol (UDP)	UDP is a connection-less oriented protocol that does not guarantee delivery of data. In the TCP/IP protocol suite, the UDP provides the primary mechanism that application programs use to send datagrams to other application programs. In addition to the data sent, each UDP message contains both a destination port number and a fully qualified source and destination addresses making it possible for the UDP software on the destination to deliver the message to the correct recipient process and for the recipient process to send a reply.

User Product	Any end-user developed software program that uses the TReK Application Programming Interface software. Used synonymously with User Application.
Web	Term used to indicate access via HTTP protocol; also referred to as the World Wide Web (WWW).

Appendix B Acronyms

Note: This acronym list is global to all TReK documentation. Some acronyms listed may not be referenced within this document.

AOS	Acquisition of Signal
API	Application Programming Interface
APID	Application Process Identifier
ASCII	American Standard Code for Information Interchange
CAR	Command Acceptance Response
CAR1	First Command Acceptance Response
CAR2	Second Command Acceptance Response
CCSDS	Consultative Committee for Space Data Systems
CDB	Command Database
CDP	Custom Data Packet
COR	Communication Outage Recorder
COTS	Commercial-off-the-shelf
CRR	Command Reaction Response
DSM	Data Storage Manager
EHS	Enhanced Huntsville Operations Support Center (HOSC)
ERIS	EHS Remote Interface System
ERR	EHS Receipt Response
EXPRESS	Expediting the Process of Experiments to the Space Station
ES	Expected State
FAQ	Frequently Asked Question
FDP	Functionally Distributed Processor
FSV	Flight System Verifier
FSV1	First Flight System Verifier
FSV2	Second Flight System Verifier
FPD	Flight Projects Directorate
FTP	File Transfer Protocol
GMT	Greenwich Mean Time
GRT	Ground Receipt Time
GSE	Ground Support Equipment
HOSC	Huntsville Operations Support Center
ICD	Interface Control Document
IMAQ ASCII	Image Acquisition ASCII
IP	Internet Protocol
ISS	International Space Station
LDP	Logical Data Path
LES	Limit/Expected State
LOR	Line Outage Recorder
LOS	Loss of Signal
MCC-H	Mission Control Center – Houston
MOP	Mission, Operational Support Mode, and Project

MSFC	Marshall Space Flight Center
MSID	Measurement Stimulus Identifier
NASA	National Aeronautics and Space Administration
OCDB	Operational Command Database
OS	Operating System
PC	Personal Computer, also Polynomial Coefficient
PCDB	POIC Project Command Database
PDL	Payload Data Library
PDSS	Payload Data Services System
PGUIDD	POIC to Generic User Interface Definition Document
POIC	Payload Operations Integration Center
PP	Point Pair
PRCU	Payload Rack Checkout Unit
PSIV	Payload Software Integration and Verification
RPSM	Retrieval Processing Summary Message
SC	State Code
SCS	Suitcase Simulator
SSP	Space Station Program
SSCC	Space Station Control Center
SSPF	Space Station Processing Facility
TCP	Transmission Control Protocol
TReK	Telescience Resource Kit
TRR	TReK Receipt Response
TSC	Telescience Support Center
UDP	User Datagram Protocol
UDSM	User Data Summary Message
URL	Uniform Resource Locator
USOS	United States On-Orbit Segment
VCDU	Virtual Channel Data Unit
VCR	Video Cassette Recorder
VPN	Virtual Private Network

Appendix C Detailed Status Character Explanation

All of the status characters from Table 1 are shown below along with a detailed explanation on possible causes.

Status Character: 'H'
Definition: Warning High Limit Exceeded
Explanation: This parameter is being (limit) sensed and the value received is greater than the warning high limit allowed by the sensor defined for this parameter. For limit sensing the calibrated value is used if calibration is defined. If the calibrated value is not available then the converted value is used. You can see which sensor is being used and whether or not calibration is defined on the Measurements dialog in the Telemetry Processing application or by using the GetParameterInformation function in the TReK API.

Status Character: 'L'
Definition: Warning Low Limit Exceeded
Explanation: This parameter is being (limit) sensed and the value received is less than the warning low limit allowed by the sensor defined for this parameter. For limit sensing the calibrated value is used if calibration is defined. If the calibrated value is not available then the converted value is used. You can see which sensor is being used and whether or not calibration is defined on the Measurements dialog in the Telemetry Processing application or by using the GetParameterInformation function in the TReK API.

Status Character: '>'
Definition: Caution High Limit Exceeded
Explanation: This parameter is being (limit) sensed and the value received is greater than the caution high limit allowed by the sensor defined for this parameter, but is less than the warning high limit. For limit sensing the calibrated value is used if calibration is defined. If the calibrated value is not available then the converted value is used. You can see which sensor is being used and whether or not calibration is defined on the Measurements dialog in the Telemetry Processing application or by using the GetParameterInformation function in the TReK API.

Status Character: ‘<’
Definition: Caution Low Limit Exceeded
Explanation: This parameter is being (limit) sensed and the value received is less than the caution low limit allowed by the sensor defined for this parameter, but greater than the warning low limit. For limit sensing the calibrated value is used if calibration is defined. If the calibrated value is not available then the converted value is used. You can see which sensor is being used and whether or not calibration is defined on the Measurements dialog in the Telemetry Processing application or by using the GetParameterInformation function in the TReK API.

Status Character: ‘S’
Definition: Stale data
Explanation: The data returned from this TReK API call has been returned before. This check is performed by the TReK telemetry processing software based on the token value. Using the same token for multiple parameters can cause a stale status character to be returned when the data is not actually stale.

Status Character: ‘E’
Definition: Out of expected state
Explanation: The parameter is being (expected state) sensed and the calibrated value is not the expected string. You can see which calibrator and sensor is being used on the Measurements dialog in the Telemetry Processing application or by using the GetParameterInformation function in the TReK API.

Status Character: ‘ ‘ (blank)
Definition: Good data
Explanation: The ideal status character to see. Basically means that all of the conditions that causes other status characters are not true.

Status Character: ‘l’
Definition: Sensing information not available
Explanation: You have asked that the data be sensed (based on the les_sense flag in a TReK API function call). However, there is no sensing information defined. The likely cause is that you forgot to add a sensor to this parameter. You can create a sensor from the Sensors dialog in the Telemetry Processing application and use the Measurements dialog to associate that sensor with the parameter.

Status Character: ‘b’
Definition: Delta Limit Exceeded
Explanation: The parameter is being (limit) sensed and the delta limit defined in the sensor is being exceeded. This flag probably indicates that a value in your telemetry stream is changing too quickly.

Status Character: ‘y’
Definition: Could not perform delta check
Explanation: TReK tried to perform a delta check but could not. More than likely this occurred because a sample from a previous packet could not be found. This will occur on the very first packet you receive and could also indicate that all of the packets being sent are not making it to your computer. See the Telemetry Processing Statistics dialog in the Telemetry Processing application for information on the data arriving on TReK and possible missing packets.

Note: Since data is sent via UDP, it may never arrive on the network to actually be picked up and processed by TReK. For more information on the Telemetry Processing Statistics dialog, see the Telemetry Processing Users Guide (TREK-USER-003).

Status Character: ‘&’
Definition: Calibration information not available
Explanation: Calibration information could not be found to perform a requested task. You should define calibration information about the parameter. You can create a calibrator in the Calibrators dialog in the Telemetry Processing application and use the Measurements dialog to associate that calibrator with the parameter.

Status Character: ‘u’
Definition: Warning High and Delta Error Violation
Explanation: A combination of the ‘H’ and ‘b’ status characters. See those status characters for more information.

Status Character: ‘q’
Definition: Warning Low and Delta Error Violation
Explanation: A combination of the ‘L’ and ‘b’ status characters. See those status characters for more information.

Status Character: ‘s’
Definition: Caution High and Delta Error Violation
Explanation: A combination of the ‘>’ and ‘b’ status characters. See those status characters for more information.

Status Character: ‘r’
Definition: Caution Low and Delta Error Violation
Explanation: A combination of the ‘<’ and ‘b’ status characters. See those status characters for more information.

Status Character: ‘c’
Definition: Calibration Error
Explanation: Most likely caused when the calibration default set number could not be found. If this occurs, you are probably using a database that has not been validated. Validate the database as described in the Telemetry Database Users Guide (TREK-USER-005).

Status Character: ‘g’
Definition: Sensing Error
Explanation: Most likely caused when the sensing default set number could not be found. If this occurs, you are probably using a database that has not been validated. Validate the database as described in the Telemetry Database Users Guide (TREK-USER-005).