TREK

HOW TO BUILD A VISUAL BASIC SCRIPT

TUTORIAL

November 2012

TABLE OF CONTENTS

FIGURES

FIGURE PAGE

1 What you need to know before you read this document

This tutorial assumes the following:

- You are familiar with the material in the TReK Getting Started User Guide (TREK-USER-001) and the TReK Telemetry Tutorial (TREK-USER-002).

- You are familiar with the following material in the TReK Telemetry Application Programming Interface Reference Manual (TREK-USER-027):
  - Sections 1 – 8
  - GetOneNewestConvertedIntegerValue Function Description

- You have some experience with Microsoft Visual Basic 6.0.

- You know how to start the TReK Telemetry Processing application, add a packet to the packet list, and activate the packet. (See TReK Telemetry Processing User Guide TREK-USER-003.)

- You know how to start the TReK Training Simulator application, add a packet to the packet list, and send the packet. (See TReK Training Simulator User Guide TREK-USER-004.)

If you are uncomfortable with any of the items listed above, some of the terminology and concepts presented in this tutorial may be difficult to understand.

2 Technical Support

If you are having trouble installing the TReK software or using any of the TReK software applications, please try the following suggestions:

Read the appropriate material in the manual and/or on-line help.

Ensure that you are correctly following all instructions.

Checkout the TReK Web site at http://trek.msfc.nasa.gov/ for Frequently Asked Questions.

If you are still unable to resolve your difficulty, please contact us for technical assistance:

TReK Help Desk E-Mail, Phone & Fax:

E-Mail:	trek.help@nasa.gov
Telephone:	256-544-3521 (8:00 a.m. - 4:30 p.m. Central Time)
Fax:	256-544-9353

TReK Help Desk hours are 8:00 a.m. – 4:30 p.m. Central Time Monday through Friday. If you call the TReK Help Desk and you get a recording please leave a message and someone will return your call. E-mail is the preferred contact method for help. The e-mail message is automatically forwarded to the TReK developers and helps cut the response time.

3 Introduction

This tutorial will show you how to use the TReK Application Programming Interface (API) to get data into a Visual Basic application. Although this tutorial is called How To Build a Visual Basic Script, in essence all you are really doing is building a Visual Basic program. A Visual Basic Script is very similar to a Visual C++ Computation. The only difference is that the script is built using Visual Basic and the computation is built using Visual C++. So the main difference between a script and a computation lies in what the COTS product will allow you to do with the program. Since Visual Basic programs can be interpreted or compiled you can use these features to run your program in different configurations. For example, with an interpreted program you can step through your program line by line and control each step of execution.

The Visual Basic "script" or application that you will build is called VBScript. The VBScript program has the following properties:

- VBScript is comprised of a Form, a Module, and the trek_user_api.bas module file.

- The form will not be seen when the program is run. The form is used to hold a Timer Control. The Timer Control is used to activate an event procedure that calls the TReK Application Programming Interface once every second to retrieve telemetry data.

- Each telemetry value retrieved is written to a file. Only 5 values will be retrieved and written to the file.

- The module file contains one procedure called Main, which is the first event procedure called when the program initializes. The Main procedure is used to load the form that contains the Timer Control. It also opens the file for write access.

- The Event Procedure for the Timer Control does most of the work. This event procedure calls the TReK API to retrieve 5 telemetry values (one every second). After the 5 values have been retrieved, this event procedure disables the Timer Control, closes the file, plays a Beep to alert the user that all tasks have been completed, and exits the program.

The VBScript project files that match the finished version of this tutorial can be found in the TReK installation directory under \examples\VisualBasic\VBScript directory. These files can be a good resource if you want to copy and paste the source code instead of typing it in from scratch. These files also provide an easy way to verify that you have entered the correct information. For example if you run into an error, check the example files and compare them to your own.

Remember to perform incremental saves as you work through the tutorial. You never know when there's going to be a power outage. ☺

Well that's about it – Have Fun!

4 Step-By-Step

1. First you need to create a directory for your VBScript files. Go to the Windows NT Explorer and create a new directory called VBScript. (In future steps this tutorial assumes the following directory C:\VBScript. However, you can select any directory you'd like.)

2. Start the Visual Basic Application.

3. In the New Project dialog as shown in Figure 1 select Standard EXE and push **Open**. You may not see this dialog if you previously selected not to see it in the future.

Figure 1 New Project Dialog Box

4. Add a Timer control to the form as shown in Figure 2. In the Properties Window set the Interval property to 1000.

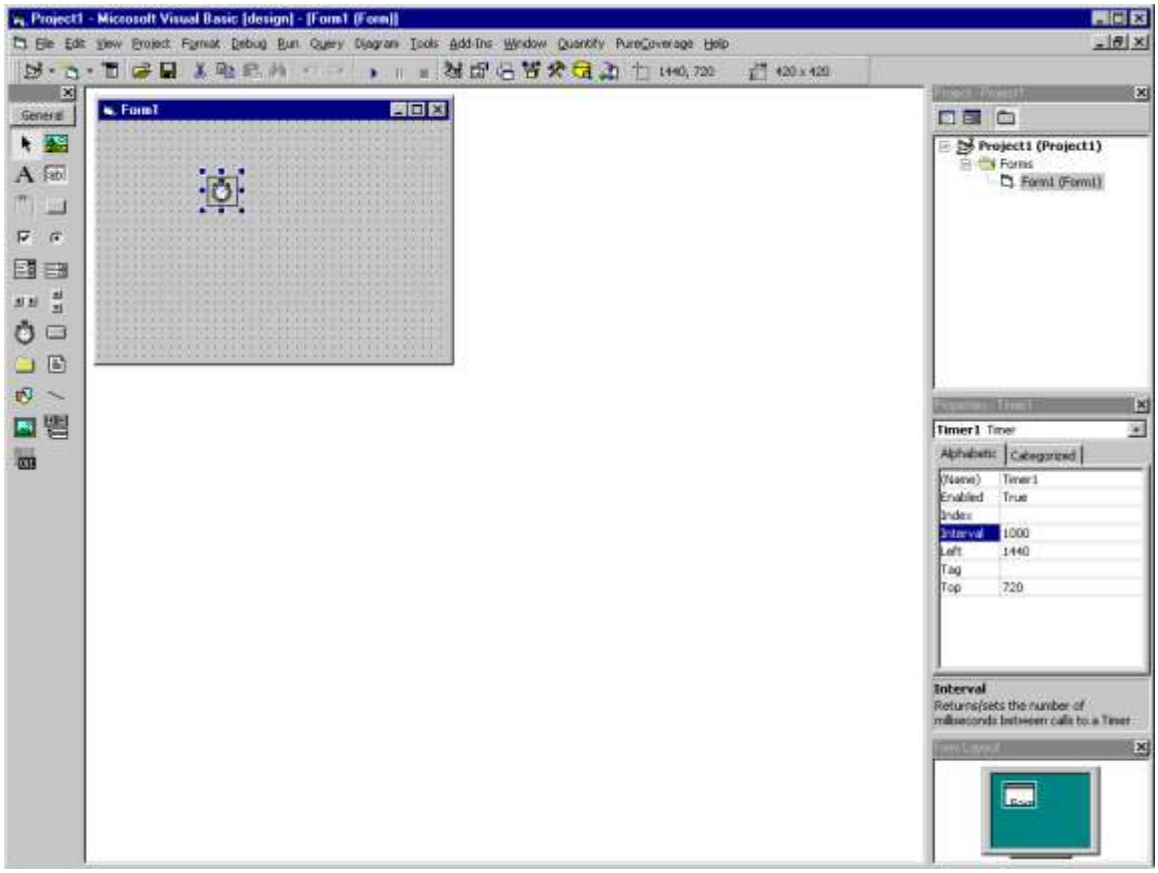

Figure 2 Timer Control

5. Select the Form and change the Form's **(Name)** property to **TimerForm**.

6. Go to the **File** menu and select **Save Project**. Visual Basic will prompt you to save the form. Save the form in the directory you created in step 1 and name the form VBScript.frm. Visual Basic will then prompt you to save the project file. Save the project file in the directory you created in step 1 and name the project file VBScript.vbp.

7. In order to use the TReK API you need to tell Visual Basic how to find the API. To do this you need to include the trek_user_api.bas file in your VBScript project. This file contains multiple Declare statements that tell Visual Basic how to find the TReK API Dynamic Linked Library. This file also contains the function prototypes in the form of Declare statements for each of the TReK API functions that are available for

use with Visual Basic. First you need to copy the trek_user_api.bas file into your directory. Go to the TReK installation directory. You will find the trek_user_api.bas file in the lib directory. Copy the trek_user_api.bas file into your VBScript directory.

8. Now that you have the trek_user_api.bas file in your VBScript directory you need to include it in your VBScript project. Go to the Visual Basic **Project** menu and select **Add File…**. In the Add File… dialog box, select the trek_user_api.bas file as shown in Figure 3 and push Open.

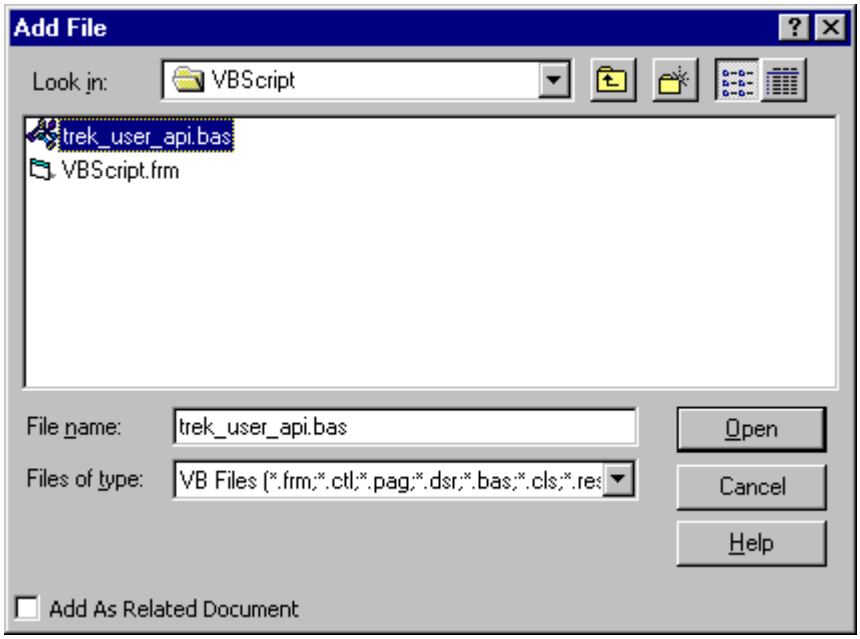

Figure 3 Add File Dialog Box

9. There are three more files that are needed for the script to run properly. They are MFC71.dll, msvcp71.dll, and msvcr71.dll. You need to copy these files into the folder with the script executable or into the Windows SYSTEM32 folder. The easiest is probably into the SYSTEM32 folder, because you will not have to copy the files for every project that you create. You can find these files in the TReK installation directory in the Examples\Visual Basic\Executables folder.

10. Now is a good time to perform another save. Go to the **File** menu and **select Save Project**.

11. During the next few steps you will add and complete the Module file. Go to the Visual Basic **Project** menu and select **Add Module**. In the Add Module dialog box select the **New** Tab and then select **Module** as shown in Figure 4. Push the **Open** button to add the new module to your VBScript project.

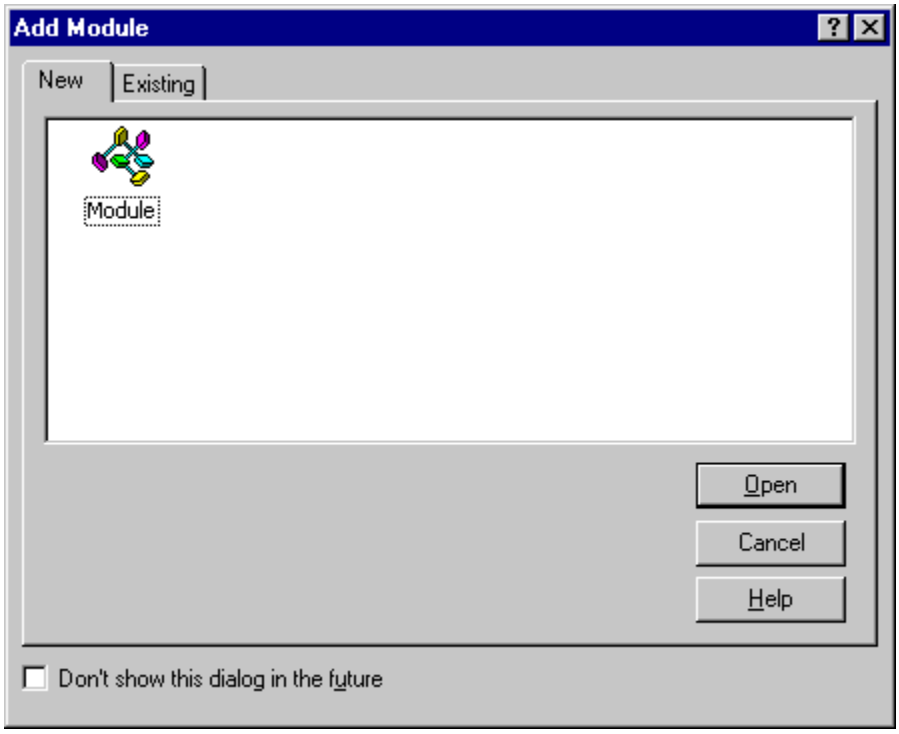

Figure 4 Add Module Dialog Box

12. Place the following event procedure in the module file. This will complete the Module file.

```
Public Sub main()
    Open "C:\\ScriptOutput.txt" For Output As 1
    Load TimerForm
End Sub
```

Note: If you do not have a C drive you need to modify the Open call and replace C: with a valid directory path.

13. The code in the Main event procedure in the module file needs to be executed first. Therefore, you need to tell Visual Basic to call Main first. To do this go to the Visual Basic **Project** menu and select **Project1 Properties**. (If you didn't change your project name, then it should still be called Project1.) In the Properties Window select the **General** tab and then set the **Startup Object** menu so **Sub Main** is selected as shown in Figure 5. Push the **OK** button.

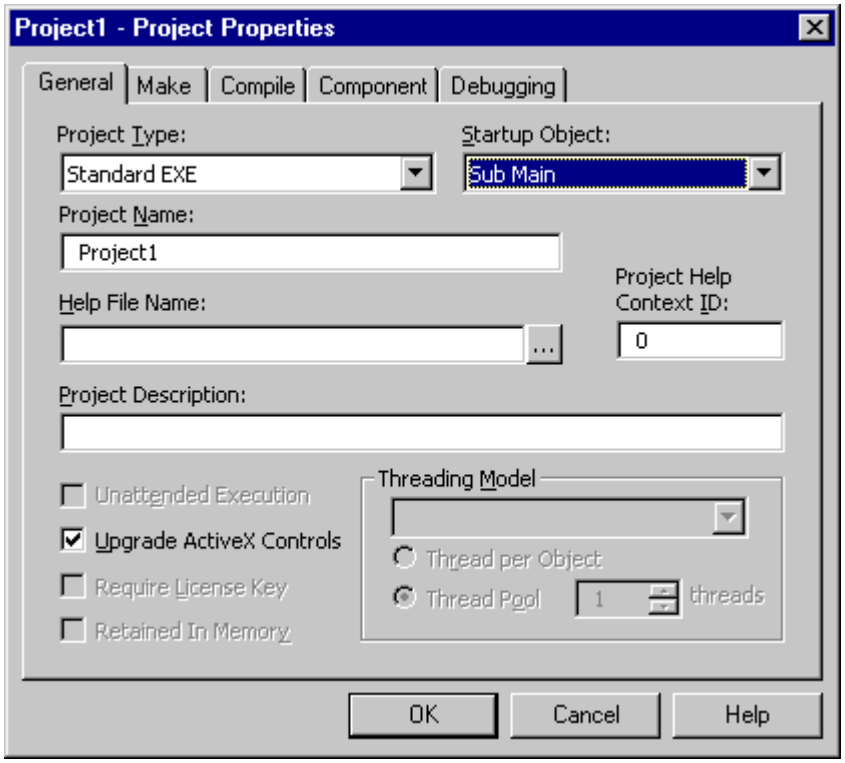

Figure 5 Properties Dialog Box

14. Go to the Visual Basic **File** menu and select **Save Project**. Visual Basic will prompt you to save the module file. Save the module file in the directory you created in step 1 and name the module file **VBScript.bas**.

15. Move to the Declarations section of the Code Window and add the following code:

```
Dim msid049_token(3) As Long
Dim number_written As Integer
```

Note: The msid049_token variable is used in the call to the TReK API to retrieve the value of MSID049. The number_written variable is used to keep up with how many telemetry values have been retrieved. Remember the VBScript program retrieves 5 values of MSID049 and writes each one to the file called ScriptOutput.txt.

16. Using the Visual Basic Window menu switch back to the TimerForm window. Add the following code inside the `Private Sub Timer1_Timer()` event procedure:

```
Private Sub Timer1_Timer()

    Dim return_value As Long
    Dim msid049_value As Long
    Dim msid049_status As String

    ' If token_flag is false, which it will be the first
    ' time through the loop, then initialize the tokens.
    ' This should only be done once.  Set the token_flag
    ' to true at the end of the loop.
    If (token_flag = False) Then
        msid049_token(0) = 0
        msid049_token(1) = 0
        msid049_token(2) = 0
        token_flag = True
    End If

    ' Allocate memory for the status string buffer.
    msid049_status = String$(8, 0)

    return_value = GetOneNewestConvertedIntegerValue(PDSS_PAYLOAD,
                        "MSID049", "", REAL_TIME, NO_SENSE,
                        msid049_token(0), msid049_value,
                        msid049_status)

    ' Write the value of msid_049 to ScriptOutput.txt
    Write #1, msid049_value
    number_written = number_written + 1


    If (number_written > 4) Then
        Timer1.Enabled = False
        Close #1
        Beep
        End
    End If

End Sub
```

This code initializes the msid049_token to prepare for the first call to the TReK API. (Note: The token_flag variable was defined for you in the trek_user_api.bas file.) After the first call, the msid049_token will retain the value from the previous API call (which is how the token should be used). Memory is then allocated for the MSID049 status string. This string really isn't used in this program but it must be used in the call to the API. The GetOneNewestConvertedIntegerValue API call is used to retrieve the value for MSID049. This value is then written to the ScriptOutput file. Once five values are written, the Timer Control is disabled, the ScriptOutput file is closed, and the program plays a Beep to alert you that it has completed all tasks. The program then exits.

17. Go to the Visual Basic **File** menu and select **Save Project**.

18. If you are in the TimerForm Code Window, use the Visual Basic Window menu to switch back to the TimerForm Form Window. Double click on the Form to open up the Code Window. This will place `a Private Sub Form_Load()` event procedure in the Code Window. Add the following code to the `Private Sub Form_Load()` event procedure:

```
Private Sub Form_Load()
    token_flag = False
    number_written = 0
End Sub
```

Note: This code will initialize the token_flag to False and the number_written variable to zero. The token_flag is used to determine whether this is the first call to the API (which indicates that the token should be initialized).

19. Go to the Visual Basic **File** menu and select **Save Project**.

20. You have now completed the VBScript program. In order to run the program and see some results you need to configure your TReK system so you will have some data flow. To do this perform the following steps:

    1. Start the Telemetry Processing application.
    2. Add Packet ID 7 to the packet list and activate it (Packet Type = PDSS Payload, Data Mode = Real Time).
    3. Start the Training Simulator application.
    4. Add Packet ID 7 to the list and set the Run Time to 180 seconds.
    5. Once the Telemetry Processing application finishes activating Packet ID 7, use the Training Simulator application to Send Packet ID 7.
    6. Start your VBScript program.

Note: Remember that you won't really "see" anything happen since the program just writes data to a file. Once you hear the program beep to indicate it has completed all tasks, you can open up the ScriptOutput.txt file to see the data that has been written to the file.

21. If you aren't interested in running your program in interpreted mode, you can create a compiled version of your VBScript program. To do this go to the **File** menu and select **Make VBScript.exe**. This will provide a way to run the VBScript program without running Visual Basic.

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