

TREK

HPEG CONSOLE

USER GUIDE



November 2024

Approved for Public Release; Distribution is Unlimited.

TABLE OF CONTENTS

<u>PARAGRAPH</u>	<u>PAGE</u>
1 Welcome.....	1
1.1 Getting Started	1
2 Technical Support.....	1
3 Introduction.....	1
4 Overview of the User Interface.....	3
5 Quick Start Guides	4
5.1 How to Start the Application with Messaging to the Console Only	4
5.2 How to Start the Application for Troubleshooting Problems	4
6 Details.....	5
7 FAQ and Troubleshooting	5
7.1 Can I run the HPEG console as a Linux service?.....	5

TABLES

<u>TABLES</u>	<u>PAGE</u>
Table 1 HPEG Configuration Parameters.....	3
Table 2 HPEG Command Line Options	4

1 Welcome

The Telescience Resource Kit (TReK) is a suite of software applications and libraries that can be used to monitor and control assets in space or on the ground.

The TReK HPEG console application provides the capability to access the Huntsville Operations Support Center (HOSC) Payload Ethernet Gateway (HPEG) service.

1.1 Getting Started

Start with the Introduction which provides an application overview. Next, try the Quick Start Guides for “How Tos” for common functions. For help with details, reference the Details section. See the FAQ and Troubleshooting section for helpful hints and solutions to the common “gotchas”.

2 Technical Support

If you are having trouble installing the TReK software or using any of the TReK software, please contact us for technical assistance:

E-Mail: trek.help@nasa.gov

Messages sent to this address are automatically forwarded to the TReK team.

The HOSC Help Desk (256-544-5066) can provide assistance as needed and is available 24x7.

3 Introduction

The TReK HPEG console application provides the capability to automate access the Huntsville Operations Support Center (HOSC) Payload Ethernet Gateway (HPEG) service. The HPEG service provides access to payloads onboard the International Space Station (ISS) using standard network protocols and services. Services supported include:

- Ping over ICMP
- Hypertext Transfer Protocol Secure (https) over TCP on port 443
- Remote Desktop (rdp) over TCP on port 3389
- Secure Shell (ssh) over TCP on port 22
- CCSDS File Delivery Protocol (cfdp) over UDP on port 4560
- Delay Tolerant Network (dtn) over TCP on a configured port

Note: In the event this list does not match the Payload to Generic User Interface Definition Document (PGUIDD) SSP 50305, the information in the PGUIDD takes precedent.

Access to these services is based on HOSC account privileges.

The HPEG console application can be started at any time and will poll attempting to start the HPEG services. The following must occur before the HPEG console application can complete the HPEG connect:

- Establish a VPN session with HOSC. This can be either a gateway-to-gateway VPN or a VPN client.
- Establish a Login Session with the HOSC using the TReK HOSC Login application.

Figure 1 shows the standard IP communication that is possible between ground applications and flight applications once HPEG services have been started.

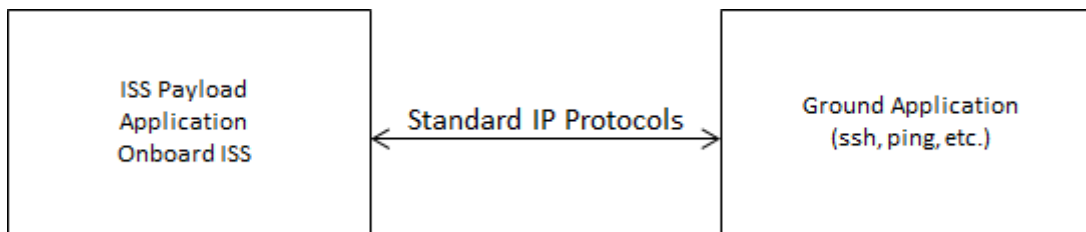


Figure 1 Standard IP Communication

HPEG Console Configuration

The HPEG Console application requires a configuration file which specifies how the HPEG application should behave. Some of the options in the file are optional and have default values are shown in Table 1.

Config Item	Required	Description
LocalIpAddr	Yes	The IP address to use for communication with the POIC systems.
LocalPort	No	The port to use for incoming HPEG connections. Default: 8919.
AutoActHpeg	No	If true, then the HPEG console application will automatically attempt to connect to the HPEG service. Default: true.
AutoStartAllDest	No	If true, then the HPEG console application will automatically attempt to start a session to all destinations when the HPEG service is activated. Default: true.
DisableIdleCheck	No	If true, the idle check performed by the POIC is disabled. This will prevent a session to a destination from being dropped by the POIC due to no socket activity. Default: true.

CommTimeout	No	Determines the amount of time in seconds to wait for communication from the POIC when activating HPEG. The minimum allowed value is 1. Default: 10.
MaxReconnectAttemptsHpeg	No	TBD
MaxReconnectAttemptsDest	No	TBD
HoscLoginName	Yes	The name used in the HOSC Login application.
LogInterfaceMessages	No	If true, protocol communication messages are logged which can provide helpful information for troubleshooting. Default: false.
InterfaceLogDir	Sometimes	Required if LogInterfaceMessages is true. The directory for the interface message log.
InterfaceLogFile	Sometimes	Required if LogInterfaceMessages is true. The filename for the interface message log. Messages are appended to a single file.
OnboardProxyIpAddress	Sometimes	Required if more than one is available through HPEG. If only one is available, then it is automatically selected.
LoginPollTimeout	No	Determines the amount of time in seconds to wait between attempts to use the HOSC login to start HPEG. Default: 10.

Table 1 HPEG Configuration Parameters

4 Overview of the User Interface

The HPEG console is intended to run as a background application but can also provide messages to the console window for debugging purposes. The configuration file described above controls the behavior of the application. There are also command line options that control what messages are printed and where they are printed. The HPEG console command line has the following format:

```
trek_hpeg_console [-h] [-l log_file] [-c] [-o {adeipw}] -f config_file
```

Each option is described in Table 2.

Option	Description
-h	Prints the options for the HPEG console application.
-l log_file	Specifies the log file to be used for messages generated by the application. If the log file is not specified messages generated by the application are lost unless also redirected to the console output. If the log file does not exist or cannot be created messages are lost.

-c	Sends messages to the console output. Can be used with or without logging (-l).
-o {adeipw}	Specifies the type of messages to print to the log. One or more of the following are allowed: <ul style="list-style-type: none"> • a – All messages (shortcut for deipw) • d – Debug messages. These messages are typically intended for troubleshooting issues and are useful for the TReK developers in determining what may cause the issue. • e – Error messages. Message that indicates failure. An error message may indicate that the application can no longer continue. In some situations, the application may be able to recover from an error. • i – Information messages. Messages that show progress/success that are important for a user to understand how the application is behaving. • p – Progress messages. Messages that indicate what is occurring in the application. Usually used as part of troubleshooting along with the debug messages. • w – Warning messages. Indicates that something didn't work as expected, but that the application is able to continue to function properly.
-f config_file	Specifies the configuration file to use for the HPEG console. This is a required command line argument.

Table 2 HPEG Command Line Options

5 Quick Start Guides

This section provides “How Tos” for common functions.

5.1 How to Start the Application with Messaging to the Console Only

To start the application with only messaging to the console, use the -c option. In the example below only information, warning, and error messages are written:

```
trek_hpeg_console -c -o eiw -f /path/to/config_file
```

5.2 How to Start the Application for Troubleshooting Problems

If you are experiencing problems with the HPEG console application, you can generate a message file and send it to trek.help@nasa.gov for assistance. In this case you should print all messages to the log file.

```
trek_hpeg_console -l /path/to/log_file -o a -f /path/to/config_file
```

6 Details

How to use HPEG API to retrieve info/get callbacks for automation.

7 FAQ and Troubleshooting

This section addresses Frequently Asked Questions and provides tips for troubleshooting common gotchas.

7.1 Can I run the HPEG console as a Linux service?

Yes...provide example files. Maybe? Eris api uses shared memory.