Telescience Resource Kit (TReK)

POIWG Splinter

July 22, 2015

NASA/EO50/Jeff Lippincott
jeff.lippincott@nasa.gov
256-544-4260

Agenda

- Background
- TReK Capabilities Overview
- EXPRESS

Background

- TReK is one of the Huntsville Operations Support Center (HOSC) remote operations solutions. It can be used to monitor and control International Space Station payloads from anywhere in the world. It is comprised of a suite of software applications and libraries that provide generic data system capabilities and access to HOSC services.

- The TReK Software has been operational since 2000. A new cross-platform version of TReK is under development. The new software is being released in phases during the 2014-2016 timeframe.
  - The TReK Release 3.x series of software is the original TReK software that has been operational since 2000. This software runs on Windows. It contains capabilities to support traditional telemetry and commanding using CCSDS packets.
  - The TReK Release 4.x series of software is the new cross platform software. It runs on Windows and Linux. The new TReK software will support communication using standard IP protocols and traditional telemetry and commanding.
  - All the software listed above is compatible and can be installed and run together on Windows.

- The new TReK software contains a suite of software that can be used by payload developers on the ground and onboard (TReK Toolkit).
  - TReK Toolkit is a suite of lightweight libraries and utility applications for use onboard and on the ground.
  - TReK Desktop is the full suite of TReK software – most useful on the ground.
  - When TReK Desktop is released, the TReK installation program will provide the option to choose just the TReK Toolkit portion of the software or the full TReK Desktop suite.

- The ISS program is providing the TReK Toolkit software as a generic flight software capability offered as a standard service to payloads.

- TReK Software Verification will be conducted during the October-December 2015 timeframe. Payload teams using the TReK software onboard can reference the TReK software verification.

Background

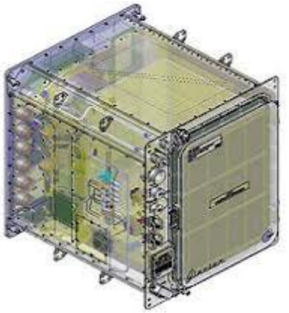
- TReK will be demonstrated on-orbit running on an ISS provided T61p laptop. Target Timeframe: Early 2016.

- The on-orbit demonstration will collect benchmark metrics, and will be used in the future to provide live demonstrations during ISS Payload Conferences.

- Benchmark metrics and demonstrations will address the protocols described in SSP 52050-0047 Ku Forward section 3.3.7.

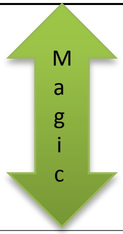
TReK Capabilities Overview

ISS



TReK Capabilities On-Board

- Send/Receive data using standard IP protocols (Unicast, Multicast, TCP Listener, TCP Server, TCP Client).
- Create, populate, build, and decompose custom packets. Includes support for pre-defined and custom headers.
- Record data.
- Transfer files (send and receive) using CFDP.
- Configure and Manage (start, stop, monitor) ION DTN node.
- Support for the following EXPRESS messages (via Ethernet):   Payload Health and Status, PEP Bundle Request, PEP Procedure Execution Request. Rack Time Request, Ancillary Data Config Control, Payload Telemetry Downlink Data.



Payload Ground Site

TReK Capabilities Ground

- Send/Receive data using standard IP protocols (Unicast, Multicast, TCP Listener, TCP Server, TCP Client).
- Create, populate, build, and decompose custom packets. Includes support for pre-defined and custom headers.
- Record and playback data.
- Transfer files (send and receive) using CFDP.
- Configure and Manage (start, stop, monitor) ION DTN node.
- Use HPEG application to log into POIC and start/stop HOSC Payload Ethernet Gateway (HPEG) session with payload. (HPEG session provides support for SSH, HTTPS, RDP, CFDP, etc.).
- Other capabilities provided via TReK Desktop (data display, data statistics, traditional telemetry & commanding (CCSDS), etc.)

Note: All capabilities are available on Windows and Linux.

TReK Operational Software Releases through CY2015

Capability	Release	Date	Description	Contents
Ku-Forward Capabilities	4.0.0	10-2-2014	This release contains capabilities to access International Space Station payloads using standard network protocols and services. It is suitable for use on the ground and onboard ISS.	• CFDP Application • CFDP Console Application • CFDP Library • Device Services Library • Data Library • HPEG Application • TReK Help Application (Integrated Help)
DTN Capabilities	4.1.0	6-5-2015	This release provides support for Delay Tolerant Networking. It is suitable for use on the ground and onboard ISS.	Adds DTN Capabilities. This release includes all the capabilities from the previous release (4.0.0) with additions and updates for DTN. • All Previous Release Content (4.0.0): • IONconfig Application • IONizer Application • IONizer Library
EXPRESS Capabilities	4.2.0	9-30-2015	This release provides support for the EXPRESS Payload to ISS C&DH System Ethernet interface.	Adds EXPRESS Capabilities. This release includes all the capabilities from the previous release (4.1.0). • All Previous Release Content (4.1.0): • Support for EXPRESS interface.

TReK 0.5.0, the first beta release with TReK EXPRESS Capabilities, will be available on August 6, 2015.

TReK Schedule information is posted on the TReK Web Site here: http://trek.msfc.nasa.gov/trek_schedule.htm).

EXPRESS APIs

- The EXPRESS APIs are grouped in the following categories:
  - *Configuration* – Functions that are typically called once to initialize API features.
  - *Sending Data* – Functions that send a message to the Rack Interface Controller (RIC) or Payload Ethernet Hub Gateway (PEHG).
  - *Data Retrieval* – Functions to get individual parameter values from ancillary data.
  - *Callbacks* – Functions that provide payload unique processing for data received from the RIC.
- All API functions are ANSI-C.
- Detailed help, including examples, will be provided for all API functions.
- At least one example EXPRESS payload application will be included.

Note: The functions listed on the following slides are subject to change.

EXPRESS APIs (Configuration)

- **InitExpressToolkit()** – Initializes the EXPRESS API including payload unique data (e.g., function code, IP address).
- **ConfigureExpressTelemetryPacket()** – Configure to send science data via RIC as an EXPRESS packet.
- **ConfigureBypassTelemetryPacket()** – Configure to bypass RIC to send science data as CCSDS packet via the PEHG.
- **GetExpressStatistics()** – Provides internal API counters for payload use (e.g., packets sent, commands received).
- **ResetExpressStatistics()** – Resets EXPRESS statistics counters to zero.

EXPRESS APIs (Sending Data)

- **SendPepBundleRequest()** – Install, halt, or remove PEP bundles.
- **SendPepProcedureExecutionRequest()** – Start, stop, or resume PEP procedure execution.
- **SendRackTimeRequest()** – Request time message.
- **SendUniqueAncillaryDataConfigurationControl()** – Start/stop unique ancillary data.
- **SendBroadcastAncillaryDataConfigurationControl()** – Start/stop broadcast ancillary data.
- **AddHeaderAndSendHealthAndStatus()** – Add the EXPRESS header to user provided data and send as health and status message to RIC.
- **AddHeaderAndSendExpressTelemetryPacket()** – Adds the EXPRESS header to user provided data and sends as science packet to RIC.
- **SendUserExpressMessage()** – Sends a user generated EXPRESS packet to the RIC. Allows user to create EXPRESS packets directly.
- **AddHeaderAndSendBypassTelemetryPacket()** – Adds CCSDS header and sends packet to designated PHEG.
- **SendBypassTelemetryPacket()** – Sends a user generated CCSDS packet to the designated PHEG.

EXPRESS APIs (Data Retrieval)

- **GetAncillaryIntegerValue()** – Retrieves an ancillary data value as an integer.
- **GetAncillaryUnsignedValue()** – Retrieves an ancillary data value as an unsigned integer.
- **GetAncillaryDoubleValue()** – Retrieves an ancillary data value as a double precision floating point number.

EXPRESS APIs (Callbacks)

- **RegisterExpressProcessCommandCallback()** – Registers a user defined callback for handling commands.
- **RegisterExpressRackTimeResponseCallback()** – Registers a user defined callback for handling the rack time response.
- **RegisterExpressPayloadResponseCallback()** – Registers a user defined callback for handling payload responses (may be unsolicited).
- **RegisterExpressProtocolErrorCallback()** – Registers a user defined callback for handling protocol errors that are identified.
- **RegisterUniqueAncillaryDataCallback()** – Registers a user defined callback for receiving unique ancillary data.*
- **RegisterBroadcastAncillaryDataCallback()** – Registers a user defined callback for receiving broadcast ancillary data.*

** - The EXPRESS API can process the ancillary data. User processing is not required.*