

CCSDS Transfer Frame Processing

This section describes the TPS CCSDS implementation only. An example on using TPS to read incoming transfer frames, strip out the virtual channel data, and simultaneously read the virtual channel packet data is described in the attached document CCSDS1.DOC.

CCSDS Dialog Box

In addition to processing packets, TPS can also process incoming CCSDS Transfer Frames.

To configure TPS to accept transfer frames, select the `Transfer, Pkt Format` menu item and then click the `CCSDS` button from the packet format dialog box. A CCSDS configuration dialog box appears which allows specification of the following properties:

- individual virtual channels for processing

- Frame Error Control Word state on/off

- Virtual channels to be recorded

The CCSDS dialog box shows an overall master switch 'On' which, when checked, switches TPS to reading Transfer Frames, i.e. it assumes the incoming data are transfer frames and translates the Attached Synchronisation marker and 6-byte header accordingly.

The Frame Error Control Word is assumed present in the Transfer Frame when this item is checked. Note, TPS does not check its value, it merely reduces the Frame Data Field by two so that the control word is not extracted as part of the frame data.

The Virtual Channels group allows selective On/Off processing of each of the 8 channels (0-7). Any combination of channels can be processed.

If all 8 channels are disabled, i.e. unchecked, TPS will still treat the incoming data as CCSDS Transfer Frames if the master 'On' item is checked. Conversely, if the individual virtual channels are selected, but the master 'On' item is cleared, TPS will not read transfer frames and not extract any virtual channel data for any channel.

Note, virtual channel data is unconditionally recorded to a separate file (VCDATAN, n=channel number 0-7) for each channel enabled in the dialog box - you do not need to select the File, Record menu item to start recording individual virtual channels. Of course, if you wish to additionally record the incoming transfer frames in their entirety, you do need to manually start a recording.

CCSDS Transfer Frame Handling

TPS processes Transfer Frames and splits out the virtual channel packet data into telemetry source packets which can then be processed by a second TPS. The Transfer Frame format is as specified in ESA's PSS-04-106 'Packet Telemetry Standard' specification document which is based upon the CCSDS 102.0-B-2 'Blue Book'.

TPS Performs the following CCSDS processing:

- Read transfer frames of a fixed at 1119 or 1279 byte length (1119 without Reed Solomon check symbols. 4 bytes sync, fixed 1115 byte Data field, optional 160 byte Reed Solomon check symbols).
- Selectively strip out and record one or more virtual channels
- Read multiple TM packet streams per virtual channel and filter a particular channel according to the TM packet Application ID (used as a synchronisation pattern).

TPS and TPSCCSDS nomenclature

Throughout this section two versions of TPS are referred to, namely, TPS (or TPSTMPKT) and TPSCCSDS. They are the same basic model and differ only in their online configuration. TPSCCSDS is dedicated to reading transfer frames and is configured by checking the 'CCSDS' checkbox in the Pkt Format dialog box which, itself, is obtained by selecting the standard TPS `Transfer, Pkt format` menu item. TPS (or TPSTMPKT) is the standard TM packet reading version and is not configured to read transfer frames but reads the virtual channel TM packet data extracted by TPSCCSDS.

Whilst TPS and TPSCCSDS are the same basic TPS model, each has to be run separately and concurrently to fully process, in real-time, transfer frames and their packet contents. TPSCCSDS processes the transfer frames and generates packet files which are then read by TPS. For non-real-time operation, each can be run standalone and independent of the other.

Overview

TPSCCSDS runs as a front-end to TPS, reading in transfer frames, splitting all 8 virtual channels, any one of which can then be read by running a standard TPS on the same or another PC. The standard TPS, i.e. not TPSCCSDS, will simply be referred to as TPS hereafter.

TPSCCSDS is identical to TPS in all respects EXCEPT it expects to read transfer frames with a 6-byte primary header as defined in PSS-04-106. Amongst other things, it uses this header to extract the virtual channel id. Thus, if you use TPSCCSDS to read any data that does NOT conform to PSS-04-106, TPSCCSDS will still assume that the first 6 bytes after the synchronisation pattern comprise the PSS-04-106 primary header and strip the virtual channel data accordingly.

Transfer Frame Structure

The assumed structure of the transfer frame is detailed following, split into separate sections for each relevant component of the transfer frame.

Reed Solomon encoded Transfer frames can be read by TPS but TPS does not use the check symbols further and simply ignores them.

Attached Synchronisation Marker (ASM)

TPSCCSDS treats the Attached Synchronisation Marker the same as the standard TPS Synchronisation Pattern and uses it to locate the start of the transfer frame.

Whilst PSS-04-106 defines a fixed synchronisation pattern 0x1ACFFC1D (the PSS-04-106 'Attached Synchronisation Marker'), TPSCCSDS does not save this hard-coded, instead, it is online configured by the user in the normal method for defining the synchronisation bit pattern (`Transfer, Pkt Format` menu item) and is saved as a system setting. It can be changed if so desired. The pattern length can also be changed. For example, if it is changed to only 2 bytes instead of 4, TPS will assume the Primary Header always follows the sync pattern regardless of the sync length.

Transfer Frame Length

This length is between 1119 bytes or 1279 bytes (usually one or the other and normally the latter, i.e. 1279 bytes, when the frames has Reed Solomon check symbols appended with an interleave depth of 5). TPS always assumes the transfer frame starts with a 4 byte Attached synchronisation pattern followed by a fixed 1115 bytes header and data field. If the frame also has Reed Solomon check symbols, there is usually an extra 160 bytes appended for an interleave depth of 5. This is a standard configuration and, without Reed Solomon check symbols, you would use 1119 as the fixed transfer frame length. With Reed Solomon check symbols you would use 1279 bytes or between 1119 and 1279 if the interleave depth is less than 5.

Note, TPSCCSDS treats the ASM as part of the Transfer Frame whereas PS-04-106 describes the ASM as a separate prefixed 4 byte entity. This discrepancy is merely because TPS treats the ASM and Transfer Frame as a single 'telemetry packet'.

Because this length is configured online it can, of course, be set to any other value between 1119 and 1279 BUT you must always have a minimum of 1119 bytes as TPS assumes that the frame data field is always fixed at 1115 bytes. A future version of TPS may make this length optional but is fixed in TPS version 5.0 at 1115 bytes. Changing the total length between 1119 and 1279 may be necessary depending upon the use of Reed Solomon encoding and the interleave depth.

Second Synchronisation Marker

TPSCCSDS does not currently look for the Second Synchronisation Marker. Because it is inserted into the Data Field it will, if present, be stripped as part of the Data Field and included in the Virtual Channel data files produced by TPSCCSDS. See further below.

Transfer Frame Primary Header Format

Frame Identification (bytes 0,1 after the ASM)

Virtual Channel ID

3 bits extracted from the Frame Identification and used by TPSCCSDS to put the stripped Data Field into the correct TPSCCSDS virtual channel data file 'VCDATAN.DAT', see further below.
Displayed on TPSCCSDS Samples Disk Page 33.

Operation Control Field Flag

1 bit flag read by TPSCCSDS to determine the size of the Data Field. If the bit is set, the Data Field size is reduced by 4 bytes to allow for the 32-bit Command Link Control Word (CLCW) appearing at the end of the data field. The CLCW is not further processed by TPS and does not appear in the Virtual Channel Data file.

Displayed on TPSCCSDS Samples Disk Page 33.

Master Channel Frame Count

Extracted and used to determine missing Transfer Frames. This check is not currently implemented but may appear in a future version of TPSCCSDS.

Displayed on TPSCCSDS Samples Disk Page 33.

Virtual Channel Frame Count

Extracted and used to determine missing Transfer Frames belonging to a particular virtual channel. This check is not currently implemented but will appear in a future version of TPSCCSDS .

Displayed on TPSCCSDS Samples Disk Page 33.

Frame Data Field Status

Secondary Header Flag

1 bit flag read by TPSCCSDS to determine the size of the Data Field. If the bit is set, the Data Field size is reduced by 4 bytes to allow for the 32-bit Secondary Header appearing immediately after the 6-byte Primary Header. The Secondary Header is not further processed by TPS and does not appear in the Virtual Channel Data file.
Displayed on TPSCCSDS Samples Disk Page 33.

Sync. Flag

1 bit flag read by TPSCCSDS but ignored - ALL transfer frames are considered to contain synchronously inserted packets, i.e. this bit (bit 1) is always considered as zero.

Displayed on TPSCCSDS Samples Disk Page 33.

Packet Order Flag

1 bit flag read by TPSCCSDS but ignored - the packet sequence-count/segmentation order is always considered 'Forward', i.e. this bit (bit 2) is always considered as zero. This is in accordance with the setting of the sync flag which is also always considered as zero. (see above).

Displayed on TPSCCSDS Samples Disk Page 33.

Segment Length Identifier

Since TPSCCSDS cannot currently process segmented Source Packets, this 2-bit field is ignored. The Source Packets are considered fixed length, this length being configured in TPSTMPKT via the `Transfer, Pkt format` menu item under Packet Length.

Displayed on TPSCCSDS Samples Disk Page 33.

First Header Pointer

An 11 bit number which is only used by TPSCCSDS to remove IDLE data, i.e. the Frame Data Field contents are not inserted into the virtual channel data file when the First Header Pointer has a value 0x7FE, the data is considered IDLE.

An option may be added to a future iteration of TPSCCSDS to further process idle data.

Displayed on TPSCCSDS Samples Disk Page 33.

Secondary Header

TPS does not currently process the secondary header when enabled (see the Secondary Header Flag, above). If the Secondary Header is present, TPSCCSDS will reduce the data field size by 4 bytes and skip the first 4 bytes immediately following the Primary header. Thus, any secondary header is not inserted by TPSCCSDS into the virtual channel data file.

Transfer Frame Data Field

TPSCCSDS extracts the entire Frame Data Field and inserts it into a 'virtual channel data file', a separate file for each virtual channel. Details are described further below in the section entitled 'Stripping Virtual Channel Data'.

The Transfer Frame Data Field Length (TFDF_Length) is computed as follows (assumes a default interleave depth of 5 for this example - modify the numbers for any other interleave depth).

TFDF_Length =
1279 bytes default TPSCCSDS 'Packet Length'
minus 4 Byte Attached Synchronisation Marker
minus 6 byte Transfer Frame Primary Header
minus 160 byte Reed solomon Check Symbols (not used)

TFDF_Length = 1109

If (Secondary Header) TFDF_Length = TFDF_Length - 4 bytes

TFDF_Length = 1105

If (Operation Control Field) TFDF_Length = TFDF_Length - 4 bytes

TFDF_Length = 1101

If (Frame Error Control Word) TFDF_Length = TFDF_Length - 2 bytes

TFDF_Length = 1099

The Frame Error Control Word can be switched on/off from within the CCSDS dialog box.

Thus, when using the usual TPSCCSDS frame length of 1279 bytes, the actual Transfer Frame Data Field length ranges from 1099 bytes to 1109 bytes dependant upon the presence of the Secondary Header, Operational Control Field and Frame Error Control Word.

Stripping Virtual Channel Data

TPSCCSDS extracts the Frame Data Field from one or more of the 8 possible virtual channel streams transmitted down a single physical channel and puts the data in 'Virtual Channel Data Files' resident in the TPSCCSDS home directory, i.e. that to which it was installed. Each channel's data is stored in a separate disk file named VCDATAN.DAT where the last character n in the root name is 0,1,2...7 representing the Virtual channel ID as given in the Primary Header.

You switch on the virtual channels required via the CCSDS dialog box.

The data files are in TPS 'Raw Binary' format and can then be read by TPS as for a standard file replay.

To clarify: the virtual channel files contain ONLY the contents of the Frame Data field. They do NOT contain the ASM, the Transfer Frame Header (Primary or Optional Secondary header), the Transfer Frame trailer (CLCW and Frame Error Control Word).

Thus, in principle, they contain a contiguous stream of multiplexed telemetry packets which can then be read by the TPSTMPKT.

Telemetry Packets and Source Packets

TPS currently expects Telemetry Packets and Source Packets to be identical, i.e. the current implementation does NOT support segmented packets.

Thus, Source packets must be less than or equal to LSEGMENT to avoid segmentation. This is the same concept as 'Idle Packets' whereby the Telemetry Packet is identical to the source packet.

Telemetry Packets and Source Packets are considered the same entity hereafter and will be referred to as Source Packets.

TPSCCSDS concerns itself only with Transfer Frames and, whilst it extracts the Frame Data Field and puts it into a virtual channel data file, it does not concern itself with the actual contents of the Data Field. On the other hand, the standard TPS, referred to as TPSTMPKT, or just 'TPSTMPKT' herein, reads the virtual channel data files and requires the Packet Header and Length information to be able to extract telemetry packets from the data. The following sections describe the relevant information.

'TPSTMPKT' and 'TPS' are assumed synonymous for the rest of this section. They process TM source packets and NOT transfer frames.

Packet Header

When processing TM packets, TPS does not assume any packets it receives are necessarily CCSDS conformant. TPS has no hard-coded knowledge of the PSS-04-106 Packet Header format and neither does it need the information. To read TM source packets, you configure the TPS synchronisation pattern to be that of the first two bytes (the 'Packet Identification') of the 6-byte 'Packet Header'. The remaining 4 bytes in the Packet Header are effectively ignored by TPS and assumed part of the Packet Data Field.

Packet Identification

As mentioned above, the leading 2-byte Packet Identification forms the 2-byte TPS synchronisation byte pattern and so TPS assumes that this is always a fixed pattern of bits. Because the Version Number, Type and Application Id fields in the 2-byte Packet Identification are fixed, this use of the Packet Identification as a synchronisation pattern is considered a safe assumption - EXCEPT for the Data Field Header Flag: this must also be fixed for all packets with the same Version Number, Type and Application ID.

Packet Length/Packet Data field

The total Source Packet Length, comprising both the Packet Header and Packet Data Field, can be fixed or variable length as of version 5.01.

Other Packet Fields

All fields in the Source Packets, EXCEPT the leading 2-byte Identification, are not used by TPS. However, they are displayed on the special Samples Disk Page 34 which has been specifically configured to show the Packet Header Data fields. Note, Samples Disk Page 33 shows the transfer frame header information.

How does TPS handle multiple packet formats (different Application Ids) ?

Because the virtual channel data files usually contain many different telemetry source packets with different Packet Identification information (specifically the 11-bit Application IDs), a single instance of TPS will only read those packets where the 2 byte Packet Identification matches the TPS configured 2-byte synchronisation pattern.

Normally, when reading packet data, TPS generates a 'Lost-Sync' error when an unrecognised Packet Identifier was encountered. To counter this, TPS has an option to inhibit the reporting of such lost-sync errors. The option is a checkbox 'Sync loss reporting' found in the TPS packet format dialog box as displayed when selecting the Transfer, Pkt format' menu item. Normally, this is set to the on state, i.e TPS will report all unrecognised synchronisation patterns. However, if you switch 'Sync loss reporting' off, TPS will just ignore the data without generating an error and will only extract those packets where the Packet Identification matches the TPS configured synchronisation pattern.

TPSCCSDS and TPS concurrent execution

Whilst running TPSCCSDS under NT, you can concurrently run TPS (installed to a different directory) on the same or another PC to read the virtual channel data files being generated, in real-time, by TPSCCSDS. Virtual Channel data files contain a contiguous stream of Telemetry Packets which, in the current implementation of TPSCCSDS, are equivalent to source packets.

On Windows NT you can, in principle, run up to 8 standard TPSs to read each of the eight virtual channels. The only real limitations are screen size and overall machine performance. To run multiple TPS's on Windows NT, you simply start separate copies.

Please see the V5.0 Supplement on the Documentation disk for full details on installing a CCSDS TPS system.