Informal Compilation of Standard Formatted Data Unit (SFDU) Stuctures

Prepared by Betsy Wilson

This document contains pieces of three other documents. I am putting it together in order to give Galileo personnel the information they need without giving more than they need. The three documents this is taken from are:

- SFOC-5-SYS-*DU-NJPL
- SFOC-5-TIS-*DU-SFDU
- SFOC-5-TIS-*DU-GLLSFDU

Because this is an informal compilation, some of the figure/table numbers may not make sense; ignore this, and read for content only.

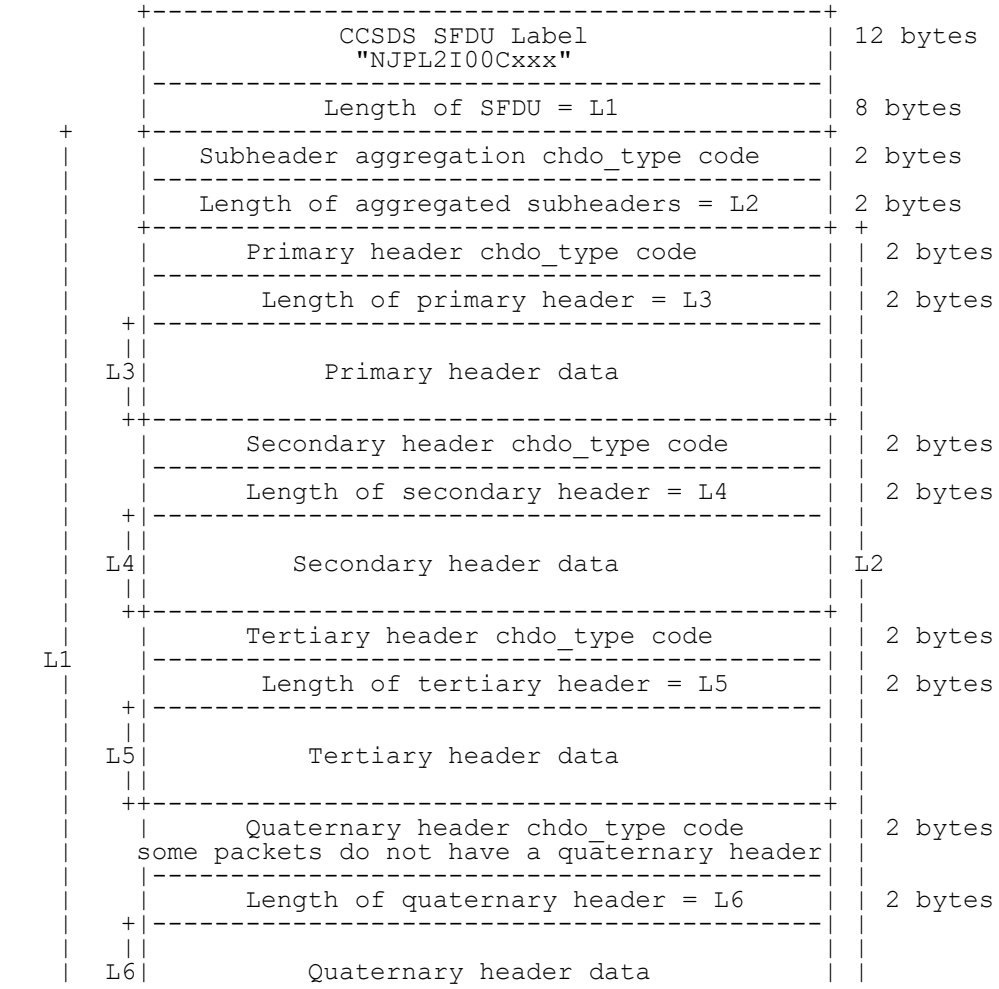
I am available for questions during work hours at:

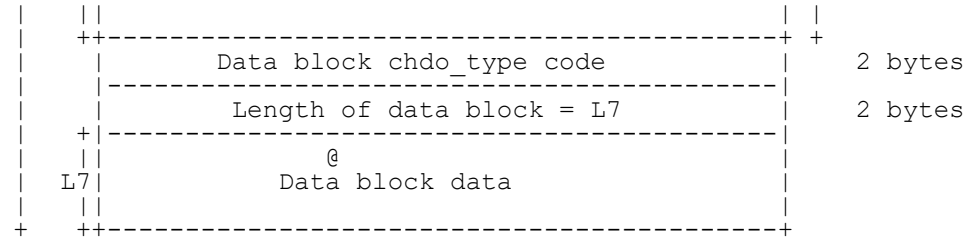
Elizabeth (Betsy) Wilson
phone: (818)354-8577
e-mail: betsy@devvax.jpl.nasa.gov
CC:Mail: Elizabeth.Wilson
JPL snail-mail: MS 179-206

CONTENTS

Composition of an AMMOS Standard CHDO-structured SFDU

This is a picture of an entire SFDU.

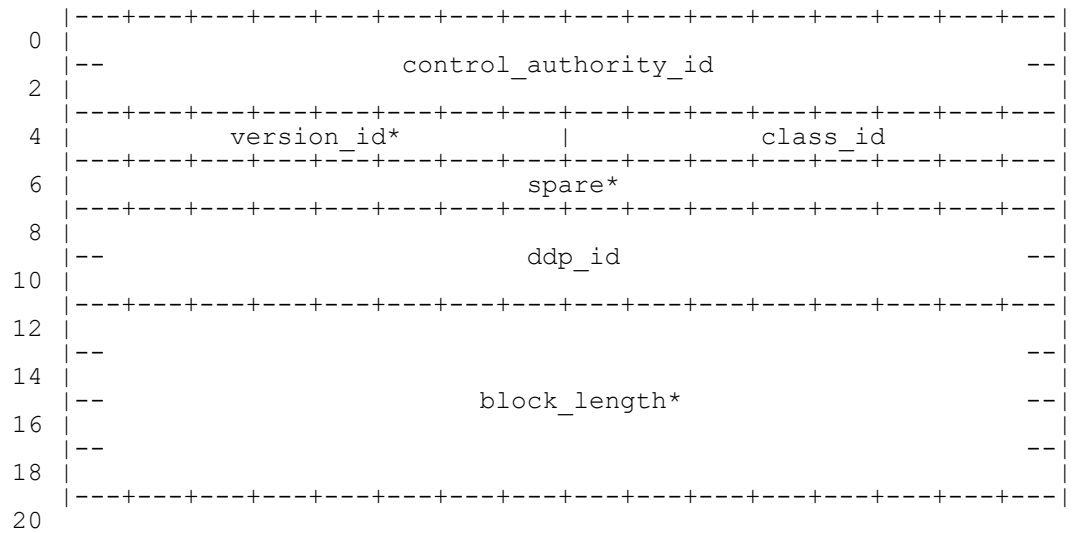




Return to Contents Links

Primary label (NJPL_____)

This label accompanies all SFDU records generated by AMMOS.



NOTE: * Beginning with the Mars Observer project, bytes 4,6, and 12-19 can have other interpretations in addition to those described in this section. In particular, byte 4 can be set to "3", byte 6 can contain delimiting types, and bytes 12-19 can have counts and markers.

Byte Offset	Field ID	Description
0-3	control_authority_id	This will always have a Value = NJPL, Restricted ASCII (RA), for AMMOS-internal SFDUs. It may also have a value = CCSD for certain other SFDU structures Note: Restricted ASCII consists of uppercase characters A-Z and digits 0-9.
4	version_id	Version ID for length field. Value = 2 (RA) indicates that the length field will be a 64-bit unsigned integer field, while value = 1 indicates an ASCII representation of length. For most AMMOS-internal SFDUs, the value = 2 option will be employed.
5	class_id	Identifies the label class. Value is one restricted ASCII character (capital letter or digit). See the "NJPL SIS."
6-7	Spares, each set to ASCII "zero".	

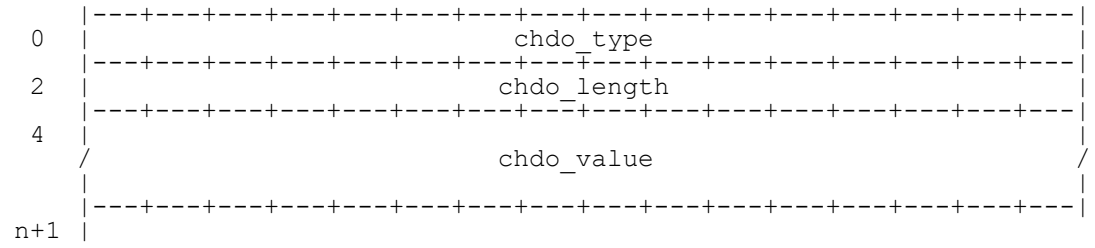
8-11	ddp_id	Data Description Package Identifier. Value is four restricted ASCII characters. Registered with the JPL Control Authority; identifies the type of data following this label (identifies the document describing data structure and content). The values that the ddp_id can assume (for AMMOS-generated records and AMMOS records received from DSN) are listed in Sections 4.2.2.3.2, 4.2.2.3.3, and 4.2.2.3.4. For AMMOS-generated standard CHDO-structured SFDUs the letter "C" is exclusively reserved as the first character of the ddp_id. Processors that receive AMMOS-produced standard CHDO-structured records as input will use the primary sub-header for record identification. Processors that receive CHDO-structured as well as non-CHDO-structured SFDUs should check record validity as follows:
		
		`if first character of ddp_id is "C"`
		`  check record id in primary sub-header`
		`  for validity`
		`else`
		`  check ddp_id for validity`
		`endif`

12-19	block_length	The length of the remainder (starting with byte 20) of this SFDU in bytes. The value must be even. 'block_length' may be either ASCII or binary, depending on the Version ID.

Return to Contents Links

Compressed Header Data Object (CHDO) Structure

A Compressed-Header Data Object (CHDO) is a Type-Length-Value Object (TLVO) that has been specially formatted to reduce the overhead introduced by the SFDU standard for structure within a single "logical" data object. The CHDO form of TLVO described in this document can only be used within AMMOS SFDU records and within SIS modules referenced here. The format of a CHDO is:



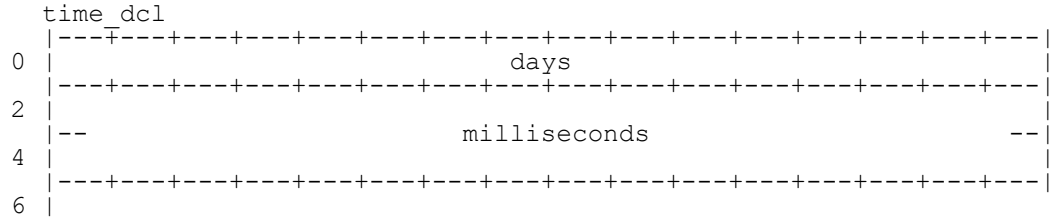
Byte Offset	Field ID	Description
0-1	chdo_type	Unsigned 16-bit integer identifying the CHDO structure. CHDO types are registered with the JPL Control Authority.
2-3	chdo_length	Unsigned 16-bit integer indicating the length of the chdo_value field in bytes (must be an even number). (value = n-3).
4-n	chdo_value	Contains any subheader or data

NOTE: The chdo_value field must be an even number of bytes in length, making all CHDOs an even byte-length, and thus keeping all SFDUs an even byte-length. Data which is naturally packaged in records of odd byte-length must be padded with an extra byte to make the length of the chdo_value field even. Any additional headers necessary to retain knowledge of the unpadded length is part of the data definition, not another CHDO header.

Return to Contents Links

Time structure

This is the format of the ERT, SCET, and RCT time fields.

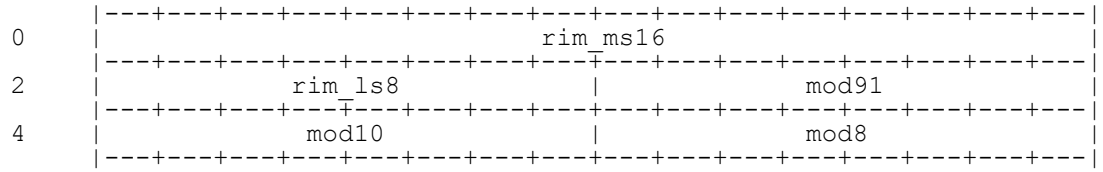


Byte Offset	Field ID	Description
0-1	days	Days since January 1, 1958, starting with 0.
2-5	milliseconds	Milliseconds of current day.

Return to Contents Links

SCLK field

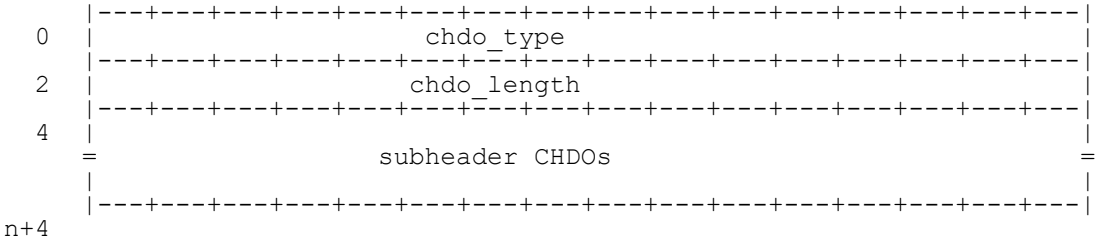
This is the structure of the SCLK field:



Byte Offset	Field ID	Description
0-1	rim_ms16	Most-significant 16 bits of the spacecraft clock RIM count.
2	rim_ls8	Least-significant 8 bits of the spacecraft clock RIM count.
3	mod91 count (range 0 - 90)	
4	mod10 count (range 0 - 9)	
5	mod8 count (range 0 - 7)	

Return to Contents Links

Subheader Aggregation CHDO

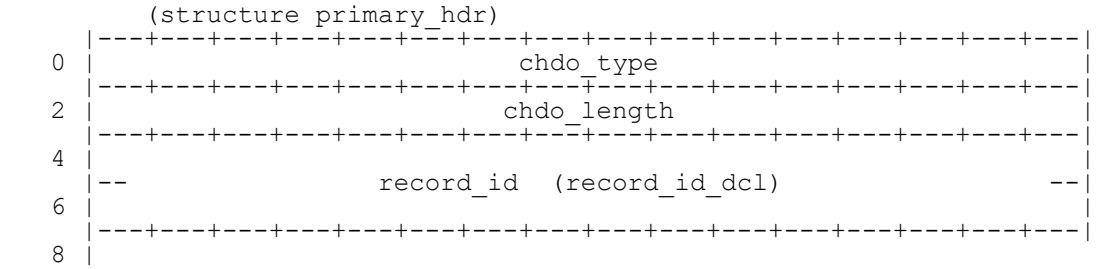


Byte Offset	Field ID	Description
0-1	chdo_type	Sub-header aggregation type code. (Value = 1).
2-3	chdo_length	Length of the combined size, n, of all subheaders that follow, in bytes, starting at the following byte. This provides an offset to the start of the data block (chdo_type).
4-n +3	subheader CHDOs	Combined subheader CHDOs occupying a total of n bytes.

Return to Contents Links

Primary Header (2)

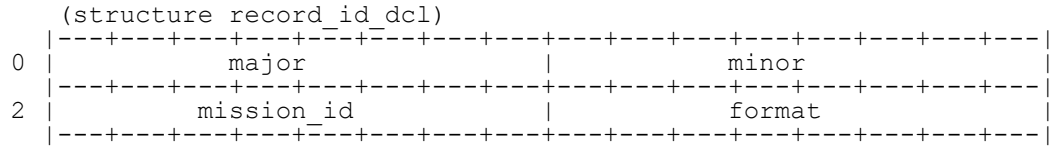
The primary header is used for all SFOC-generated CHDO-type SFDUs.



Byte Offset	Field ID	Description
0-1	chdo_type	Telemetry primary header CHDO type code. (Value = 2).
2-3	chdo_length	Length of CHDO value field (remainder of header). (Value = 4).

4-7	record_id	Record identifier (described below in structure record_id_dcl, section 4.2.1.4.1).

Record_ID



Byte Offset	Field ID	Description
0	major	SFDU major type. Types 0-127 are reserved for common types. Specifies the major data categorization. For the complete list of assigned values refer to the "record IDs at the end of this document.
1	minor	SFDU minor type. Types 0-127 are reserved for types that are defined and apply to more than one mission. Types 128-255 are used for mission-unique data. See the record ID section at the end of this document for a list.
2	mission_id	Mission identifier code. Always "1" for GLL
3	format	SFDU format type. This field is used in conjunction with major and minor to specifically define data types, which are mission-specific and defined in the mission-specific SFDU SIS module. Values are assigned as described for the minor type field above. See the record ID section at the end of this document.

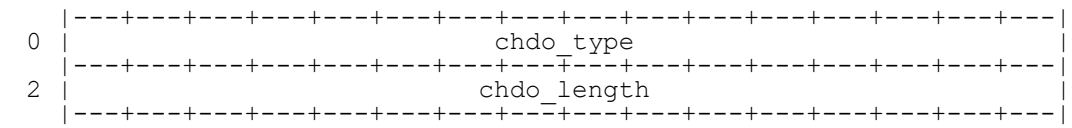
Return to Contents Links

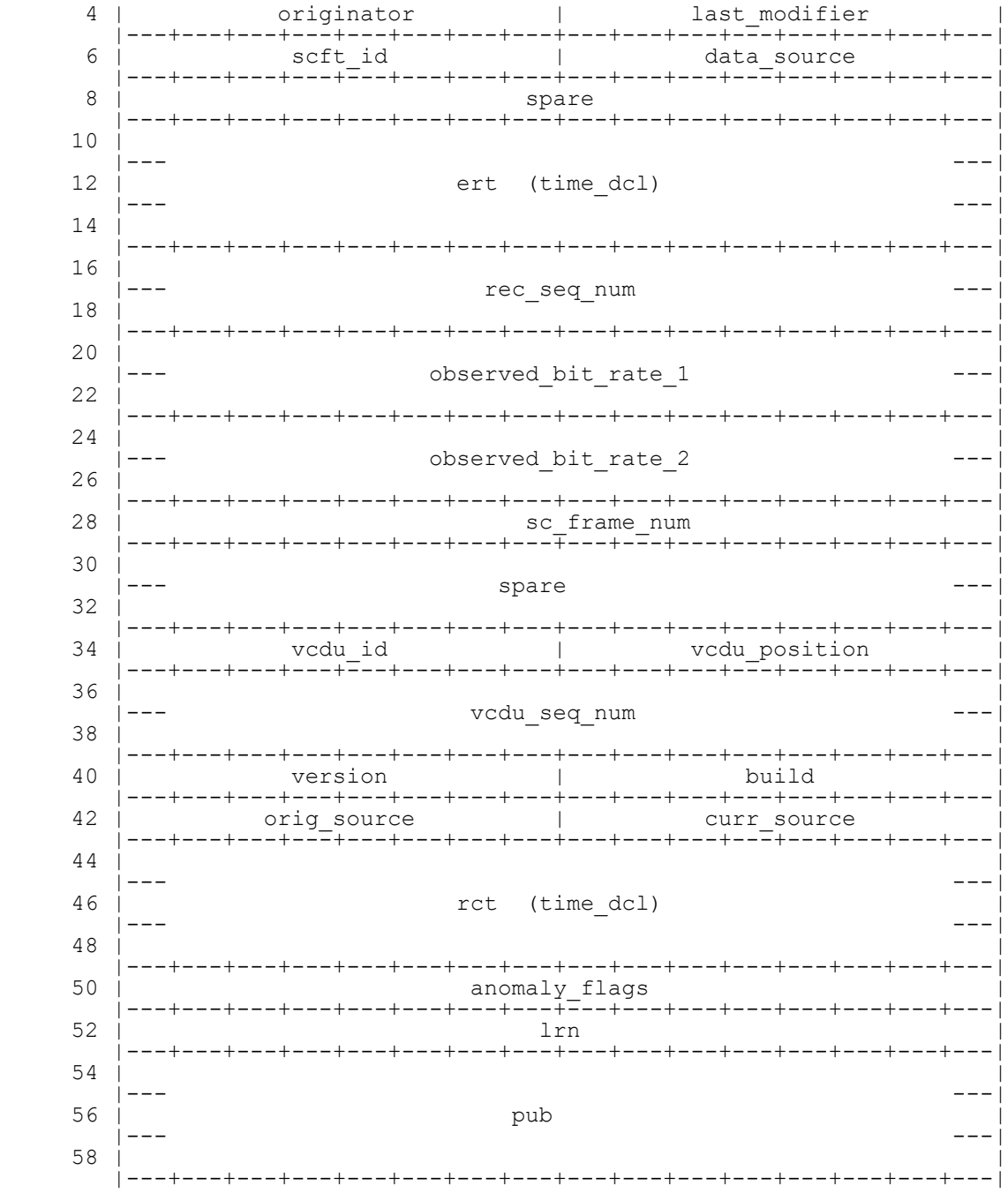
GLL Phase 2 Telemetry Frame Secondary Header (700)

NOTE: The Phase II telemetry frame header is in TDA document 820-13-TLM 3-12G and is not shown here. It is called "DGT Secondary Header" in that document.

Return to Contents Links

GLL VCDU Secondary Header (47)





NOTE: Many of these fields are taken directly from the DGT-generated secondary header, found in TLM 3-12G.

Byte Offset	Field Id	Description
0-1	chdo_type	VCDU secondary header CHDO type code. (Value = 47).
2-3	chdo_length	Length of CHDO value field (remainder of header). (Value = 50).
4	originator	Originator ID.
5	last_modifier	Last modifier ID. Set to the TIS identifier. Valid values are in SIS module SFOC-5-SYS-*DU-NJPL.

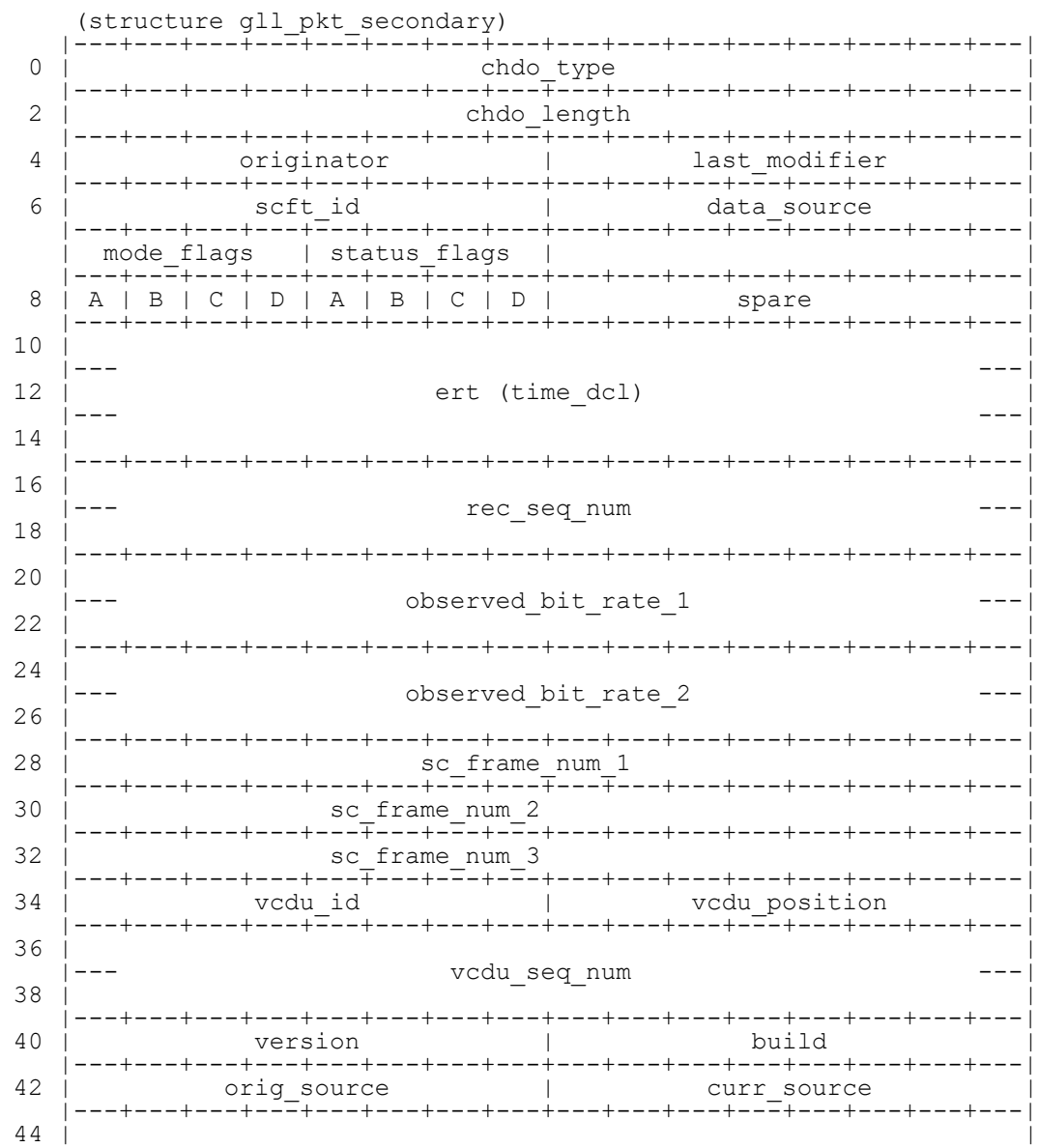
6	scft_id	Spacecraft identifier (ref. 820-13 OPS 6-21 for code value).
7	data_source	DSN station ID of source (ref. 820-13 OPS 6-3 for code value).
8-9	spare	
10-15	ert	Earth Received Time. TIS computes this ERT using the ERT of the transfer frame, and maximum bit rate to reflect the approximate receipt of the first bit of the Packet VCDU. The ERT calculation is always made at a bit rate of either 160 bps (maximum real-time rate) or 134Kbps (maximum test-bed rate). This is done because the rate may change mid-frame. Using the max rates means that while VCDUs and packets may have a time that is slightly sooner that it should be, no times overlap into the following packet or VCDU.
16-19	rec_seq_num	Record sequence number. This is the record sequence number in the DGT secondary header (same field name).
20-23	observed_bit_rate_1	First observed bit rate. IEEE floating point. See Applicable Document #1f.
24-27	observed_bit_rate_2	Second observed bit rate. IEEE floating point. See Applicable Document #1f.
28-29	sc_frame_num	Telemetry frame counter from FCD frame tertiary header field of same name
30-33	spare	(will be used for other 2 sc_frame numbers in packet header)
34	vcdu_id	VCDU channel identifier. Valid values are 0 thru 7.
35	vcdu_position	VCDU position. The position number of the VCDU within the transfer frame. Valid values are 1 thru 4.
36-39	vcdu_seq_num	VCDU sequence number. The sequence number increments within the channel and is unique throughout the Galileo Phase II. Only the least significant 20 bits of this number is used.
40	version	Software version number (0-255) of the TIS.
41	build	Software build number (0-255) of the TIS.
42	orig_source	Indicates the original input path of the data that caused the creation of this record. This field is set by TIS when data is actually being received from one of these interfaces. It is copied by TIS from the input record during replay from a spooler file or SFDU tape. Valid values are: 0 = Not applicable 1 = Router A 2 = Router B 3 = Wide band switch 4 = IDR tape 5 = DSN-GIF LAN I/F. 6 = CDA spooler file 7 = SFDU tape. 8 = DTS virtual circuit

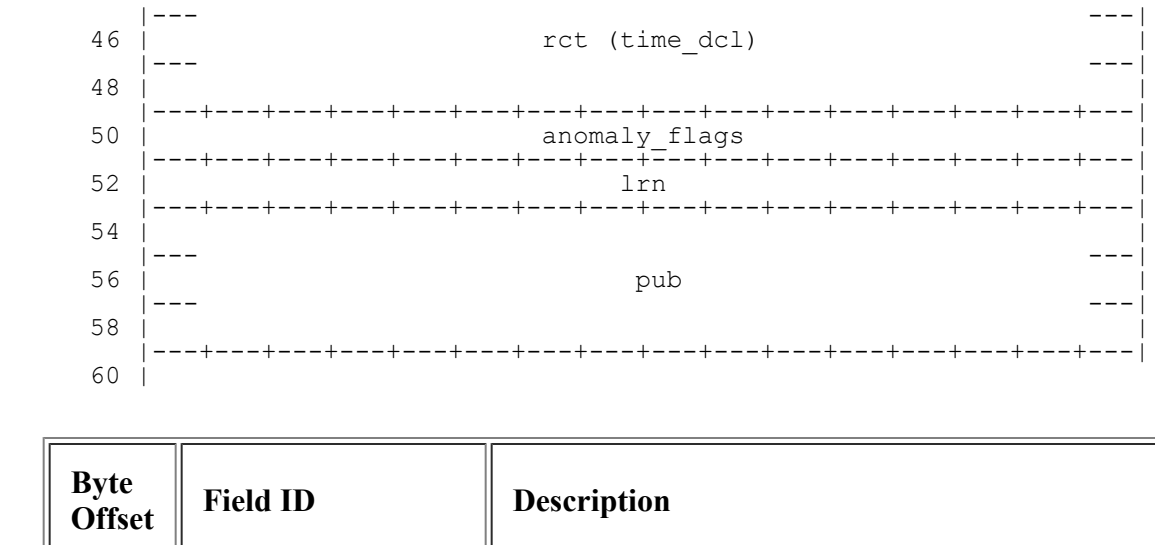
		9 = CDA bytestream file 10 = UNIX bytestream file 11 = SIM
43	curr_source	Indicates the current input path of the data that caused the creation of this record. This field is set by the TIS according to the current source of the input data. Valid values are: 0 = Not applicable 1 = Router A 2 = Router B 3 = Wide band switch 4 = IDR tape 5 = DSN-GIF LAN I/F 6 = CDA spooler file 7 = SFDU tape. 8 = DTS virtual circuit 9 = CDA bytestream file 10 = UNIX bytestream file 11 = SIM
NOTE: If original_source and curr_source differ, then curr_source must indicate either CDA spooler file or SFDU tape, DTS virtual circuit, CDA bytestream file, or UNIX bytestream file.		
44-49	rct	Record Creation Time. This field contains the system clock time, which is the SMC generated GMT time, at which the record was created by GIF or TIS. System clock time is maintained by SMC.
50-51	anomaly_flags	These flags are used to indicate an end in the sequence of normal sequential data of this record type. On normal good data, all flags are set to 0. An anomaly record is typically generated by copying the header of the last normal record and by setting the appropriate flags indicating the cause of the anomaly. Anomaly records contain a null data CHDO. When any of these flags are set, the data_val flag in field status_flags is also set. A spare. B upstream — Upstream anomaly - set only in conjunction with another anomaly flag. Indicates that the generating program received an anomaly record in its stream of input records and passed on the anomaly indication in its output stream. C other — Any reason not identifiable by other anomaly flags. D thru I spare. J off — Generating processor turned off. K timeout — Input timeout. L sequence — Break or regression in input record sequence. GIF tests field dsn_record_seq (block serial number) for input received from the DSN. Sequence checks on SFOC-generated SFDUs are always performed on field lrn. M overflow — Data was lost in real-time due to queue overflow.

		N interface An I/O error in an input interface caused loss of data. O-P spare.
52-53	lrn	Logical record number. Set by the creating process. Contains the sequence number of all SFDUs of the same data type (same major / minor / format type). This counter starts with 1 for the first record of a given type after subsystem start-up, increments by 1 for each SFDU of the same type, and wraps to 0 on overflow from 65,535. After subsystem start-up, this counter is never reset by causes other than overflow.
54-59	pub	An ASCII string of six project-unique bytes, entered into the TIS subsystem by user control directive. TIS places into these bytes whatever the user has entered.

Return to Contents Links

GLL Packet Secondary Header (48)





Byte Offset	Field ID	Description
0-1	chdo_type	Galileo Packet Secondary header CHDO type code. (Value = 48).
2-3	chdo_length	Length of CHDO value field (remainder of header). (Value = 56).
4	originator	Originator ID.
5	last_modifier	Last modifier ID. Set to the TIS identifier.
6	scft_id	Spacecraft identifier (ref. 820-13 OPS 6-8 for code value).
7	data_source	DSN station ID of source (ref. 820-13 OPS 6-8 for code value).
8	mode_flags	

	A pb_mode	Spacecraft realtime/playback flag. Set by the TIS based on VCDU type.
		Valid values are:
		`0 = Realtime or unknown`
		`1 = Direct Tape Recorder Playback.`

	B data_mode	Indicates whether data is real or simulated. Set to 1 by SIM.
		Valid value are:
		`0 = Real or unknown`
		`1 = Simulated.`

	C test_mode	Indicates whether data is test- or flight-generated.
		Valid values are:
		`0 = Test complex/GDS test generated`
		`1 = Flight (MOS) generated.`

	D replay_flag	SFOC realtime/replay flag. Set to 1 by TIS when data is replayed from spooler files, IDR tapes, or SFDU tapes; otherwise set to 0. Set to 1 by TIS when data is replayed from spooler files or SFDU tapes.

		Valid values are:
		`0 = SFOC realtime`
		`1 = SFOC replay.`

8	status_flags	
	A data_val	Data validity (0 = good data). Set to 1 by GIF and TIS for anomaly records. Anomaly records have at least one flag in field anomaly_flags set, and their normal data CHDO is replaced by a null CHDO.

	B scid_force	Spacecraft ID was forced by the operator.
		`0 = no.`
		`1 = yes.`

	C ert_val	Earth received time (0 = ert is valid, 1 = ert is known to be bad).

	D sclk_suspect	Set and used by the TDS to identify SCLKs which appear to be invalid. Only TDS should use this field.
		`0 = SCLK okay.`
		`1 = SCLK value is suspect.`

9	spare	
10-15	ert	Earth Received Time. TIS computes this ERT using the ERT of the transfer frame, and maximum bit rate to reflect the approximate receipt of the first bit of the Packet. The ERT calculation is always made at a bit rate of either 160 bps (maximum real-time rate) or 134Kbps (maximum test-bed rate). This is done because the rate may change mid-frame. Using the max rates means that while VCDUs and packets may have a time that is slightly sooner than it should be, no times overlap into the following packet or VCDU.
16-19	rec_seq_num	Record sequence number. This is the record sequence number in the DGT secondary header (same field name).
20-23	observed_bit_rate_1	First observed bit rate. IEEE floating point.
24-27	observed_bit_rate_2	Second observed bit rate. IEEE floating point.
28-29	sc_frame_num	Telemetry frame counter from FCD frame tertiary header field of same name
30-33	(unamed)	Each 2 bytes, respectively, the sc_frame_number of the second and third (if either) frame contributing bits to this packet. If this packet was made from one frame only, then these 4 bytes will contain zero.
34	vcdu_id	VCDU channel identifier. Valid values are 0 thru 7.
35	vcdu_position	VCDU position. The position number of the VCDU within the transfer frame. Valid values are 1 thru 4.

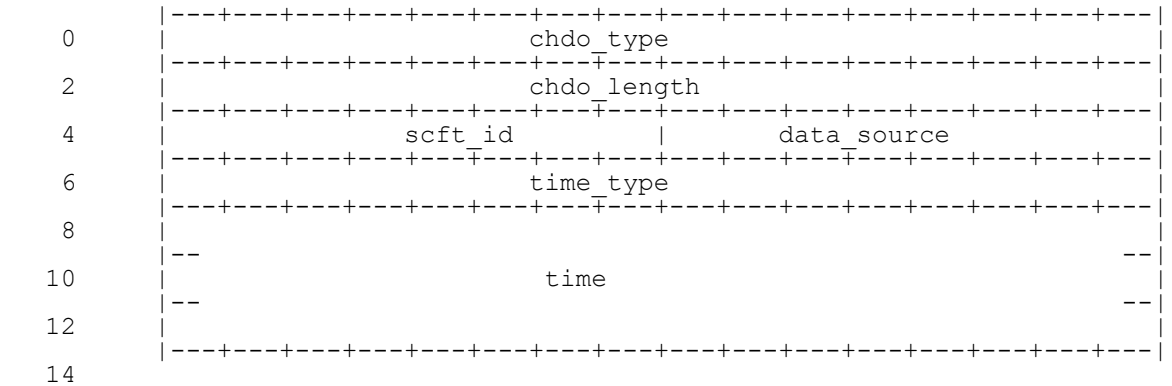
36-39	vcdu_seq_num	VCDU sequence number. The sequence number increments within the channel and is unique throughout the Galileo Phase II. Only the least significant 20 bits of this number is used.
40	version	Software version number (0-255) of the TIS.
41	build	Software build number (0-255) of the TIS.
42	orig_source	Indicates the original input path of the data that caused the creation of this record. This field is set by TIS when data is actually being received from one of these interfaces. It is copied by TIS from the input record during replay from a spooler file or SFDU tape. Valid values are: 0 = Not applicable 1 = Router A 2 = Router B 3 = Wide band switch 4 = IDR tape 5 = DSN-GIF LAN I/F 6 = CDA spooler file 7 = SFDU tape 8 = DTS virtual circuit 9 = CDA bytestream file 10 = UNIX bytestream file 11 = SIM
43	curr_source	Indicates the current input path of the data that caused the creation of this record. This field is set by the TIS according to the current source of the input data. Valid values are: 0 = Not applicable 1 = Router A 2 = Router B 3 = Wide band switch 4 = IDR tape 5 = DSN-GIF LAN I/F 6 = CDA spooler file 7 = SFDU tape 8 = DTS virtual circuit 9 = CDA bytestream file 10 = UNIX bytestream file 11 = SIM NOTE: If original_source and curr_source differ, then curr_source must indicate either CDA spooler file or SFDU tape, DTS virtual circuit, CDA bytestream file, or UNIX bytestream file.
44-49	rct	Record Creation Time. This field contains the system clock time, which is the SMC generated GMT time, at which the record was created by GIF or TIS. System clock time is maintained by SMC.
50-51	anomaly_flags	These flags are used to indicate an end in the sequence of normal sequential data of this record type. On normal good data, all flags are set to 0. An anomaly record is typically generated by copying the header of the last normal record and by setting the appropriate flags indicating the cause of the anomaly. Anomaly records contain a null data CHDO. When any of these flags are set, the data_val flag in field status_flags is also set. A spare B upstream Upstream anomaly - set only in

		conjunction with another anomaly flag. Indicates that the generating program received an anomaly record in its stream of input records and passed on the anomaly indication in its output stream. C other — Any reason not identifiable by other anomaly flags. D-I spare J off — Generating processor turned off. K timeout — Input timeout. L sequence — Break or regression in input record sequence. GIF tests field dsn_record_seq (block serial number) for input received from the DSN. Sequence checks on SFOC-generated SFDUs are always performed on field lrn. M overflow — Data was lost in real-time due to queue overflow. N interface — An I/O error in an input interface caused loss of data. O-P spare
52-53	lrn	Logical record number. Set by the creating process. Contains the sequence number of all SFDUs of the same data type (same major / minor / format type). This counter starts with 1 for the first record of a given type after subsystem start-up, increments by 1 for each SFDU of the same type, and wraps to 0 on overflow from 65,535. After subsystem start-up, this counter is never reset by causes other than overflow.
54-59	pub	An ASCII string of six project-unique bytes, entered into the TIS subsystem by user control directive. TIS places into these bytes whatever the user has entered.

Return to Contents Links

Secondary TDS Header - Channelized Data Record (16)

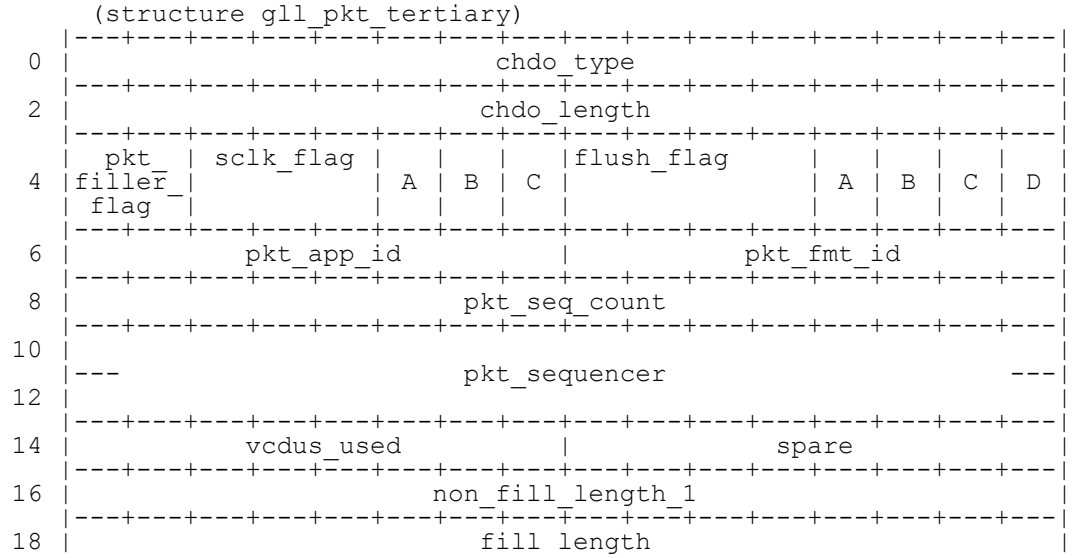
A TDS channelized data record is multi-mission and will always include the following secondary header:

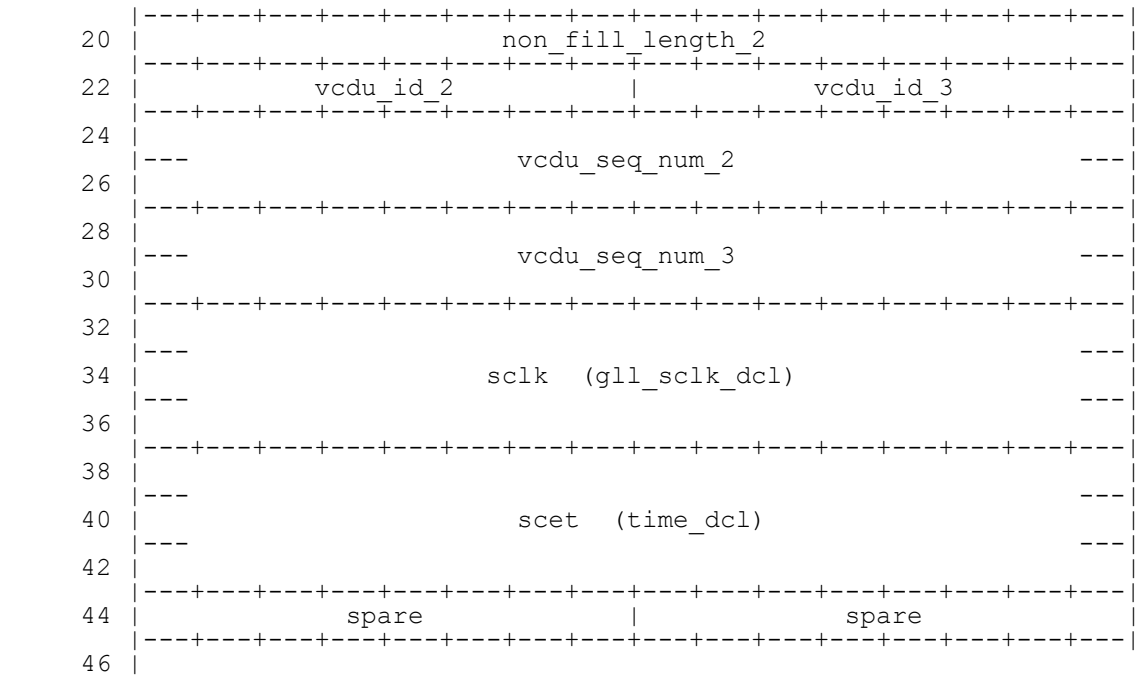


Byte Offset	Field ID	Description
0-1	chdo_type	TDS channel data secondary header CHDO type code. (Value = 16).
2-3	chdo_length	Length of CHDO value field (remainder of header) in bytes. (Value = 10).
4	scft_id	The spacecraft identifier as assigned by the DSN. Refer to 820-13 OPS 6-8 for the code value.
5	data_source	The DSN station ID of the source of the DSN monitor data. Refer to DSN 820-13 OPS 6-3 for the code value. This field will contain a value only when monitor channels have been queried.
6-7	time_type	This is a positive integer value which identifies the type of time, which is the next field. The type of time reported in this header always corresponds to the type of time used in the specification of the query. The following table identifies the possible values and their meaning, where format refers to the format designations used by SFOC-2-SYS-Any-TimeForms. time_type format Description 01 gll_sclk_dcl Galileo SCLK 101 time_dcl Time of Storage (TOS) 102 time_dcl Monitor Sample Time (MST) 102 time_dcl Radio Science Sample Time (RSST) 103 time_dcl Spacecraft Event Time (SCET) 104 time_dcl Earth Receive Time (ERT) 105 time_dcl Record Creation Time (RCT)
8-13	time	A 6-byte field containing the time value. The type of time is indicated by the field time_type.

Return to Contents Links

GLL Packet Telemetry Tertiary Header (49)





Byte Offset	Field ID	Description
0-1	chdo_type	Galileo Packet Telemetry Tertiary header CHDO type code. (Value = 49).
2-3	chdo_length	Length of CHDO value field (remainder of header). (Value = 42).
4	pkt_filler_flag	Indicates completeness of the packet. 0 = complete. 1 = partial packet (filler at end of packet). 2 = packet with a gap (filler in the middle) 3 = sub-packet with filler in the front
	sclk_flag	Indicates the derivation of the packet SCLK. 0 = packet contained an explicit SCLK. 1 = SCLK derived by forward extrapolation. 2 = SCLK derived by backward extrapolation. 3 = SCLK is zero. Could not derive. See flush_flag for reason.
	A	sclk_calc_suspect SCLK calculation suspect. Indicates whether missing neighbor packets makes the calculation of this packet's SCLK suspect. If it is suspect, then odd data results may not be what they seem, it could mean that the SCLK is actually incorrect. This reports only a "possibility" of the SCLK being incorrect, not an actuality. 0 = SCLK calculation is okay. 1 = SCLK calculation is suspect.
	B	sclk_unexpected The SCLK received on this packet was not what the ground system expected. This

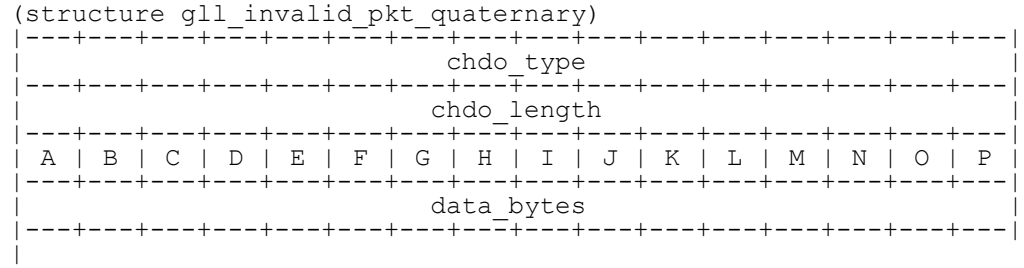
		could mean that the ground system calculations were off, the spacecraft de-selected and re-selected data - resulting in an unseen gap, or data was lost. It may mean that the SCLK of one or more immediately previous packets is incorrect. 0 = SCLK is okay 1 = SCLK was not expected value
	C	spare
5	flush_flag	Indicates reason the packet was flushed.
	bits 0 to 3	0 = Not flushed. 1 = Flushed by user request. 2 = Exceeded SCLK continuity number. 3 = Reached partial packet flush threshold. 4 = Overflow of packet hold. 5 = Job ended with packet in hold. 6 = Short packet received. 7 = FID/Image number changed. 8 = No SCLK available, no holding allowed 9 = Compressed data was not decompressed
	A scet_val	Indicates whether the SCET is valid. 0 = SCET invalid. 1 = SCET is valid.
	B scet_int	SCET Interpretation. Indicates whether the SCET time is the actual converted SCLK time or a predicted (interpolated from last record in SCET/SCLK conversion table) time. 0 = Actual 1 = Predicted
	C less_than_max	"Short packet". Packet size is less than the maximum number of bytes. Used only for packets where this information is needed, useful, or determinable. Note: This is not the same as a packet with filler. This flag is not related to, and not mutually exclusive with, the "partial packet" indicator. 0 = not less than max 1 = less than max
	D spare	
6	pkt_app_id	Packet application ID. This is the original numeric APID from the packet.
7	pkt_fmt_id	Packet format ID. This is the numeric format ID from the packet, if any.
8-9	pkt_seq_count	Packet sequence count. This is a wrapping sequence count with a maximum value of 127.
10-13	pkt_sequencer	Packet sequencer. This is a synthetic value produced for the purpose of maintaining the proper order among packets, particularly in those cases where a SCLK value is repeated for many packets. In fact, the pkt_sequencer will be unique and always increasing throughout the mission. The following indicates the components of pkt_sequencer.

		bits 0 to 3 All zeros. bits 4 to 23 The VCDU sequence number. bit 24 Indicates whether the packet sequence count has rolled over AND there was at least one preceding packet header of the same type in this VCDU. 0 = no rollover. 1 = rollover. bits 25 to 31 Packet sequence count. EXAMPLE: VCDU# PKT# (dec) pkt_sequencer (hex) 4 126 47E 5 127 57F 5 0 580 6 1 601 6 2 602
14	vcdus_used	Number of VCDUs used for this packet extraction (1-3).
15	spare	
16-17	non_fill_length_1	The length in bytes of the first contiguous set of valid packet bits extracted. For a complete packet, this is the same as the packet length. For an AACS sub-packet with fillerin the front, this value will be zero.
18-19	fill_length	The length in bytes of missing data for an incomplete packet. For an AACS sub-packet with filler in the front, this is the number of filler bytes in the front of the sub-packet.
20-21	non_fill_length_2	The length in bytes of the second contiguous set of packet bits extracted following a gap. This can only be non-zero in the case of a three-VCDU extraction with the middle VCDU missing or for an AACS sub-packet with filler in the front, this is the number of valid bytes in the back, after the filler.
22	vcdu_id_2	The VCDU channel ID of the second VCDU used in the packet extraction, if any. The first VCDU ID appears in the secondary header.
23	vcdu_id_3	The VCDU channel ID of the third VCDU used in the packet extraction, if any.
24-27	vcdu_seq_num_2	The VCDU sequence number of the second VCDU used in the packet extraction, if any. The first VCDU sequence number appears in the secondary header.
28-31	vcdu_seq_num_3	The VCDU sequence number of the third VCDU used in the packet extraction, if any.
32-37	sclk	Spacecraft clock. The SCLK is either extracted directly from the packet or derived from a neighboring packet. See the sclk_flag.
38-43	scet	Spacecraft event time. The TIS converts the SCLK time to Universal Time Constant (UTC) format using the SCLK/SCET correlation coefficients.

44-45	spare

Return to Contents Links

GLL Invalid Packet Quaternary Header (39)



Byte Offset	Field ID	Description
0-1	chdo_type	JPL telemetry quaternary type code (value = 39).
2-3	chdo_length	Length of CHDO value field (remainder of header) (value = 4).
4-5	pkt_error_flags -- Each invalid packet record will have one and only one of the bits below set. All other bits in this field will be set to zero. The set bit indicates the reason for the invalidity of the bytes in the data area of this record.	
	The next five flags (A-E) indicate that this record contains bytes at the beginning of a VCDU, prior to the byte pointed to by the First Header Pointer (FHP). These bytes cannot be assigned to a valid packet. They are put into a separate invalid record to distinguish them from bytes (if any) from a different VCDU.	
	A	missing_first_part - the VCDU with the sequence number before this one was missing.
	B	invalid_continuation - the last packet in the previous VCDU had an invalid APID, and it was made into an invalid packet.
	C	min_size_continuation - the last packet in the previous VCDU had a size smaller than the minimum size for that APID, and it was made into an invalid packet.
	D	max_size_continuation - the last packet in the previous VCDU has a size for that APID, and it was made into an invalid packet.
	E	bad_fhp - the FHP of the VCDU did not match the expected value. The previous packet (awaiting bytes from the VCDU with the incorrect FHP) was made into a partial packet, with filler at the end, but the bytes from this VCDU were not included.
	The next three flags (F-H) indicate a record containing bytes starting with the first byte of a packet, and ending with the last data byte of the same VCDU. If the VCDU with a sequence number one greater was received, the first few bytes will be made into another invalid packet, labelled with one of the five flags	

	above.	
	F	invalid_apid - this packet had an invalid APID.
	G	min_size - this packet size was less than the minimum size for this APID.
	H	max_size - this packet size was more than the maximum size for this APID.
	I	wrong_vcdu - this packet was in a VCDU whose ID is not legal for the packet type, although the APID is valid. The data area contains an entire packet, which may have spanned VCDUs.
	J	no_data_area - the VCDU containing the rest of this packet was missing, and this packet has either an incomplete header, or no data area. The data area of this record starts with the first byte of a packet and will never be longer than a packet header.
	K	no_sclk - this packet should have had an SCLK (according to the value of the packet sequence number, or the length of the previous packet (a short packet) and it did not.
	L	invalid_fid - this packet uses the FID, and the value was not a valid value.
	M	invalid_sclk - this packet had an SCLK with an invalid value. Either the M91 count was > 90, or the RTI field (PWH2/3 only) was > 9.
	N-P	spare
6-7	data_bytes	Number of bytes in the data area of this CHDO. This number is either equal to the number of bytes in the CHDO "length" field, or one less. If it is one less, then the extra byte is set to binary zero.

Return to Contents Links

GLL Packet Engineering Frame Quaternary Header (42)

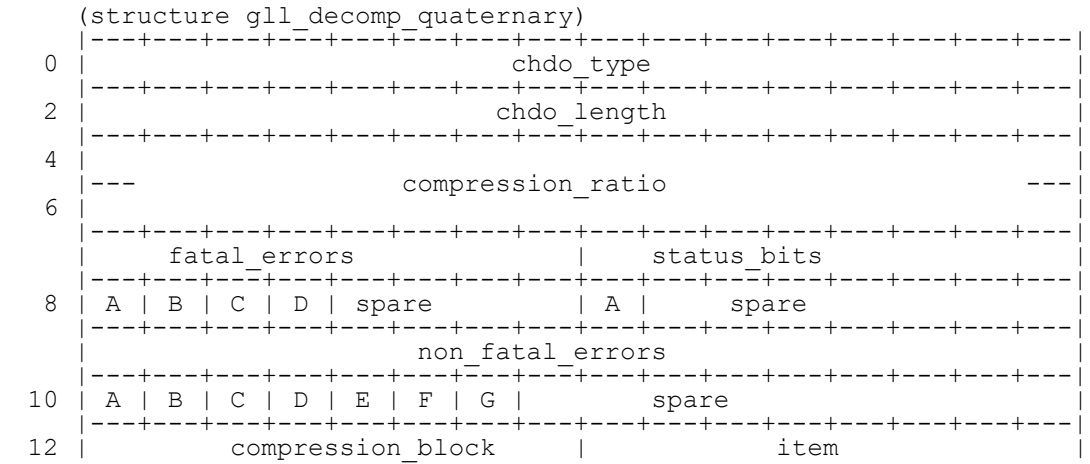
```
  (structure gll_eng_pfram_quaternary)
  |---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---|
0 |                          chdo_type                            |
  |---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---|
2 |                         chdo_length                           |
  |---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---|
4 |  rate |mro|  cmi  |    msn    | A | B | C |                   |
  |---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---|
6 |
```

Byte Offset	Field ID	Descripton
0-1	chdo_type	JPL telemetry quaternary type code (value = 42).
2-3	chdo_length	Length of CHDO value field (remainder of header) (value = 2).

4	rate_decom_flags	This is a copy of the ID field in the first byte of engineering data. If any of these fields have been forced (indicated by the force_flag) the forced values will appear here.
	rate	Data rate of engineering. 0 = 2 bps 1 = 10 bps 2 = 40 bps 3 = 1200 bps
	mro	Mro flag. 0 = no mro 1 = mro
	cmi	CMI, the commutation map index.
	msn	MSN, the map sequence number.
5	force_flags	
	A	MRO_forced 0 = mro not forced. 1 = mro forced.
	B	CMI_forced 0 = cmi not forced 1 = cmi forced
	C	MSN_forced 0 = msn not forced 1 = msn forced

Return to Contents Links

GLL Rice-Decompressed Packet Quaternary Header (38)



```
    |---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---|
  14
```

Byte Offset	Field ID	Description
0-1	chdo_type	JPL telemetry quaternary type code (value = 38).
2-3	chdo_length	Length of CHDO value field (remainder of header) (value = 10).
4-7	compression_ratio	Ratio of uncompressed data to compressed data, IEEE floating point number. On packets with filler, this represents the ratio as far as the good data is concerned, or is at least closely in the neighborhood.
8	fatal_errors	fatal errors from decompressing
	Fatal errors cause an SFDU with no data area. Non-fatal errors will have an SFDU data area, and users must understand risk in using data with any bit in the "decompression_status" words set (except the bit "short mfcount - an indicator of a short - not partial- packet, which is not an error).	
	Bit A -- bad_apid	Fatal decompression error, no data available.
	Bit B -- mfcount_toosmall	Given minor frame count is less than minimum size. Possible header error. Fatal error, no data available.
	Bit C -- mfcount_toobig	Given minor frame count exceeds nominal. Possible header error. Fatal error, no data available.
	Bit D -- internal_error	A fatal internal error has occured that is not considered to be data dependent. This is a s/w error, not data error, and can occur with any other non-fatal error.
	Bits E-H -- spares	
9	status_bits	These status bits are not errors, and may exist with any other values in either the fatal or non-fatal areas.
	A -- short_mfcount	Given minor frame count is less than nominal (maximum). Data OK. Status report only. Always seen on "short" packets (not partial), which is expected. Can occur with other error bits (although this is NOT an "error"): data_underrun, block_overrun, internal_error, block_overrun, recip_id_failure, filler_limit, ref_recovered. The data is still subject to other status conditions and may not be OK. Note that a short packet can also be a partial packet.
	B-H -- spares	
10	non_fatal_errors	Non-fatal decompression results. The status bits below are not mutually exclusive, IE more than one bit can be set. Combinations are stated with each bit or group of bits. A non-fatal error may or may not mean some bits are corrupted.

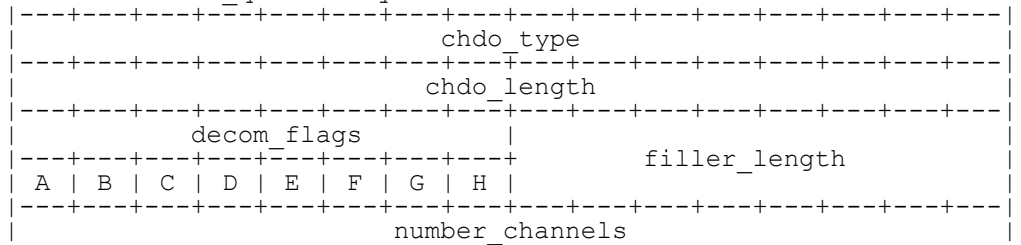
Following 2 fields are set only after all decoding is done, each bit (B and C) can be set together with other (non-fatal) errors: internal_error, short_mfcount, recip_id_failure	
Bit A -- data_underrun	Number of bits decoded were less than the given packet data area length, not fatal, but all decompressed data in packet are in doubt.
Bit B -- data_overrun	Number of bits decoded exceeds the given packet data area length size, not fatal, but all decompressed data in packet are in doubt.
Bit C -- block_overrun	Indicates that the number of bits decoded in the specified (see field "compression_block") block exceeded the maximum allowed for one compression block, given the number of values expected. Further decoding was aborted. This error can occur together with any of: internal_error, short_mfcount, recip_id_failure, filler_limit, ref_recovered. The only use for the "item" field in a record with this bit set is to indicate the last decoded item only for the situation when the "filler_limit" AND "zero_option" bits are set. "zero_option" is the only condition in which the data items are decoded in the actual order received thereby allowing a "last decoded item" to be specified.
Bit D -- recip_id_failure	Reciprocal ID computation did not match given option ID. Decompressed data failed integrity check after indicated "compression_block" and "item" numbers (see other fields in this same header, below). Data still provided. After decompression, the resulting decompressed data items are used to recalculate the coder option ID using the same algorithm as the compressor. The result is unique and must match the ID given in the data stream. If it does not, the data in the reported block and beyond must be considered corrupted, or an error occurred during decompression (detected or not). If other errors were detected that _may_ have corrupted the data, but this bit is not set, odds are better that the data was not corrupted. (This error can occur once per compression block, but only the first block is reported.) This error can occur with: data_underrun, data_overrun, internal_error, short_mfcount, filler_limit, ref_recovered.
NOTE - the three bits below are one group, used only when the packet contains some filler (a partial packet). "filler_limit" means there was some filler in the packet, "ref_recovered" means the decompressor got a reference value for the compression block in which the filler started. "zero_option" means that data samples up to and including the sample number in field "item" are good. "ref_recovered" and "zero_option" both must be set with "filler_limit". "zero_option" implies "ref_recovered" is also set, but "ref_recovered" does not imply that "zero_option" is set. (In other words, the only three possible mixtures are: (C), (C & D), (C & D & E)).	
Bit E -- filler_limit	Partial packet limit reached. Data was decompressed to the point of the given valid data limit. Filler was used for the balance of the packet. First "bad" block number provided in field "compression_block". Can occur with errors: internal_error, short_mfcount1, recip_id_failure, ref_recovered, zero_option and default_option.
Bit F -- ref_recovered	Used in conjunction with filler_limit. Indicates that the block in which the data interruption occured contained a recovered reference value. This is only meaningful when this is a partial packet (contained some filler). Can occur with

		errors" internal_error, short_mfcount1, recip_id_failure, ref_recovered, zero_option.
	Bit G -- zero_option	Used in conjunction with filler_limit. Indicates that the block in which the data interruption occured was coded with code option zero and data up to and including the item indicated in field "item" (below) was recovered. Can occur with errors: bad_apid, internal_error, short_mfcount, recip_id_failure, filler_limit, ref_recovered. (Cannot occur with default_option) ("item" is last good measurement, starting at measurement 1. All measurements from after "compression_block" are bad. "Compression_block" is the last good block. (This option is normally used for low-entropy data - not much variation.))
	Bit H -- default_option	Used in conjunction with filler_limit. Data up until field "compression_block" is good. All blocks after "compression_block" are bad. Within the given "compression_block", measurements are bad until (but not including) measurement whose number is in field "item". "Item" is the first good measurement, and measurements are good until the END of the block. If "ref_recovered" is set, measurement 1 (one) of the compression block is also good (it is the reference value). Can occur with errors: bad_apid, internal_error, short_mfcount, recip_id_failure, filler_limit, ref_recovered. (Cannot occur with zero_option). For example, if "item" = 17, and the total number of measurements in a block is 20, then measurements 17, 18, 19, and 20, are good, and measurements 1-16 are bad (unless ref_recovered is set, in which case measurement 1 is also good and 2-16 are bad.) (This option is normally used for high-entropy data - lots of variation.)
11	spare	
12	compression_block	Set to non-zero if any filler was in this packet, and it was decompressed anyway, or if decompression failed in the middle of the packet (see other status bits above). This number represents the compression block in which the filler first started, values of 1-N, where the packet normally contains "N" compression blocks . (I.e., this is the first "bad" block, not the last good block.) The first reported compression_block number will be 1.
13	item	Applicable only to "code option 0" (zero) items.

Return to Contents Links

Quaternary Header - Channelized Data (27)

This header accompanies all channelized data records.



```
  |---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---|
8 |                           map_id                              |
  |---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---|
10|
```

Byte Offset	Field ID	Description
0-1	chdo_type	Channelized data quaternary header type code. (Value = 27).
2-3	chdo_length	Length of CHDO value field (remainder of header). (Value = 6).
4	decom_flags	
	A map_valid	Indicates whether the decommutation map used was a valid one. Rules for selecting a valid map are mission-specific. 0 = Valid map used 1 = Invalid map used.
	B-H spare.	
5	filler_length	This is the number of filler bits at the end of this SFDU record added in order to pad the data out to an even word length. Filler bits will follow the actual data area of the SFDU. Value may be 0 - 15.
6-7	number_channels	Number of channels in this SFDU.
8-9	map_id	Contains the version_id field of the SFDU K header of the map used to perform decommutation for this SFDU. If this field does not exist, this is set to hex FFFF. If the version_id is X.Y, then the left byte contains X and the right byte contains Y. Values are 1 to 216-1.

Return to Contents Links

Quaternary Header - Expanded Channelized Data (ECDR) (32)

```
  +---+---+---+---+---+---+---+---|---+---+---+---+---+---+---+---+
0 |                           chdo_type                           |
  |---+---+---+---+---+---+---+---|---+---+---+---+---+---+---+---|
2 |                          chdo_length                          |
  |---+---+---+---+---+---+---+---|---+---+---+---+---+---+---+---|
4 |                           num_items                           |
  |---+---+---+---+---+---+---+---|---+---+---+---+---+---+---+---|
6 |                           spare16                             |
  +---+---+---+---+---+---+---+---|---+---+---+---+---+---+---+---+
```

Byte Offset	Field ID	Description
0-1	chdo_type:	value = 32

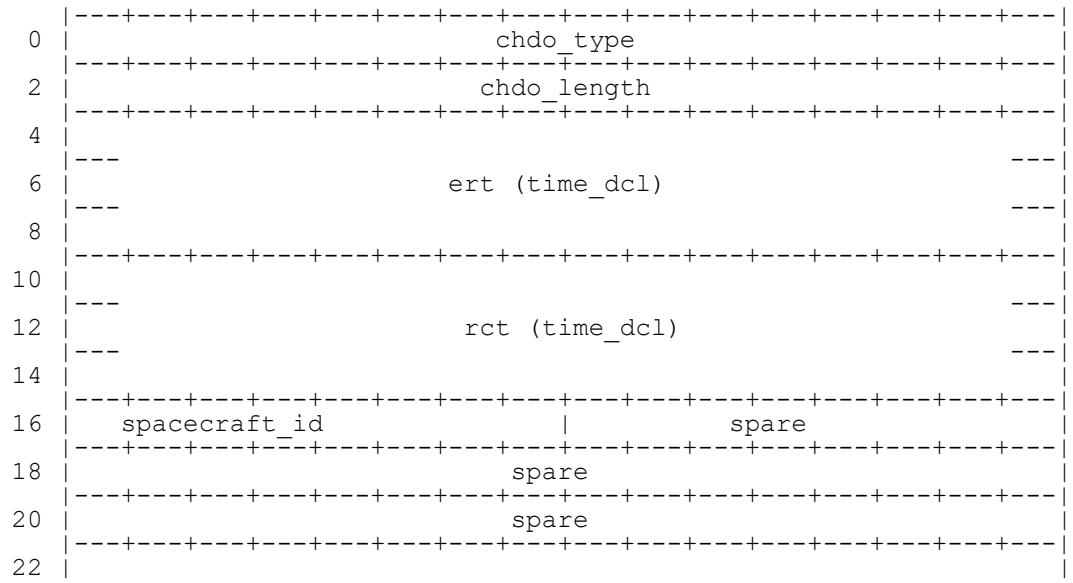
2-3	chdo_length:	Length in bytes of this structure (remainder of header) (value = 4).
4-5	num_items:	Number of channel structures (ECDR Data Blocks) in this ECDR SFDU.
6-7	spare16:	Spare 16 bits

Return to Contents Links

GLL DGT Performance Monitor Records Tertiary Header (703)

DGT performance monitor records are received and channelized by the TIS. Because the format of the time fields in the records is non-standard, TIS translates the time to a standard format, and places the times in this header. This header is not on the original received record, but is on all versions of the record downstream of the TIS. This header is not mentioned in TLM 3-12G (the DSN document), because DSN does not make it nor know about it.

The original header on all Performance Monitor records in a Secondary Header of type=255. This can be seen in the TLM 3-12G document.



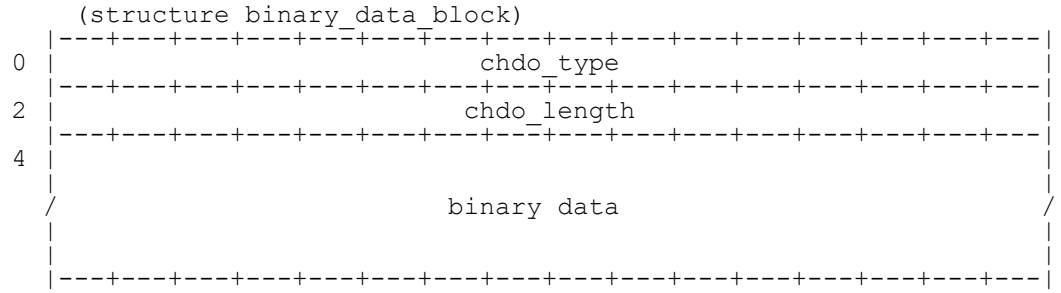
Byte Offset	Field ID	Description
0-1	chdo_type	JPL telemetry tertiary type code (value = 703).
2-3	chdo_length	Length of CHDO value field (remainder of header) (value = 18).
4-9	ert	Data (Earth Received) time of latest record written by the DGT sub-system creating this record, prior to the creation of this record. (either the BTD, SCD, or FCD)
10-15	rct	Wall-time of creation of this record by the DGT subsystem.

16	spacecraft_id	spacecraft ID
17-21	spare	saved for future use

Return to Contents Links

Data Block Header - Standard Non-structured Binary (10)

This data block type is used for all binary data that does not require any further breakdown into individual fields for processing purposes within SFOC. The data field may be of any length (within CHDO length constraints).

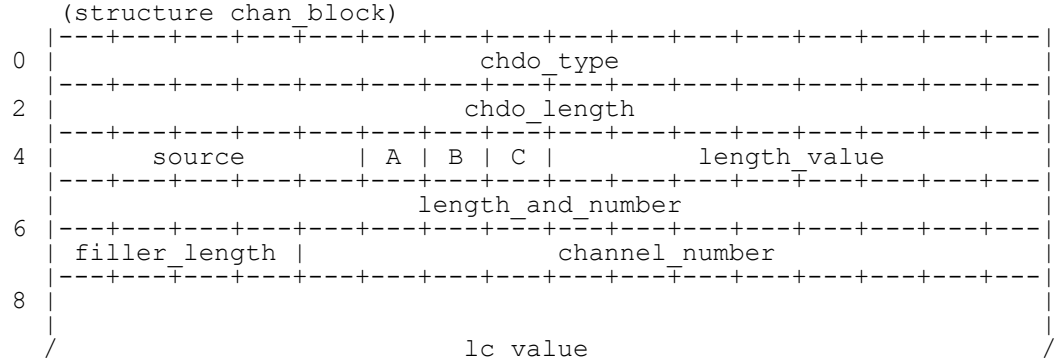


Byte Offset	Field ID	Description
0-1	chdo_type	Binary data block type code. (Value = 10).
2-3	chdo_length	Length of CHDO value field (data) in bytes, starting at the following byte. This value will always be an even number of bytes.
4-n	data	Data of variable length, depending on record_id (major, minor, mission ID, and format).

Return to Contents Links

Data block - Channelized Data (28)

This data block is used for all types of channelized records.



```
   |          (this field may not exist for a channel)             |
 n |                                                               |
   |---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---|
```

Bytes 4 - lc_value are repeated once for each channel.

Byte Offset	Field ID	Description
0-1	chdo_type	Channelized data block type code. (Value = 28).
2-3	chdo_length	Length of CHDO value field (data) in bytes, starting at the following byte. This value will always be an even number of bytes.
NOTE: The previous 4 bytes are at the start of the data area only; the next bytes are repeated for each channel. The total length of the next structure depends on the size of the channel.		
4	source	Specifies the alphabetic character used in the channel identifier. (Values = 1 - 23 (1=A, 2=B, etc.) (X, Y, and Z are not allowed)
	A lv_flag	This flag identifies whether bits 8 through 15 (length_value) contain the actual value of the channel (if it will fit in a single byte), or the length of the actual value in words. 0 = length_value contains a length 1 = length_value contains a value and field lc_value does not exist for this channel.
	B bad_data	Indicates whether valid data or filler was decommutated. 0 = Valid data was used in decommutation process 1 = Filler data was decommutated.
	C Spare	
5	length_value	This field contains either the actual value of the channel or the length of the actual value field (number of 16-bit words) stored in lc_value. Flag lv_flag determines which applies, e.g., a 3 byte value uses 2 16-bit words of space.
6-7	length_and_number	
	Bits 0-3 -- filler_length	The number of bits preceding the most significant bit of the actual channel value (in lc_value), since lc_value is right-justified. (Values are 0 - 15), e.g., a 14-bit channel would have a filler_length = 2.
	Bits 4-15 -- channel_number	The numerical portion of the channel identifier (Values are 0 - 4095).
8-n	lc_value	Long channel value. If the value for a given channel cannot fit into a single byte, its value will then be stored here. The value should be stored in one to two

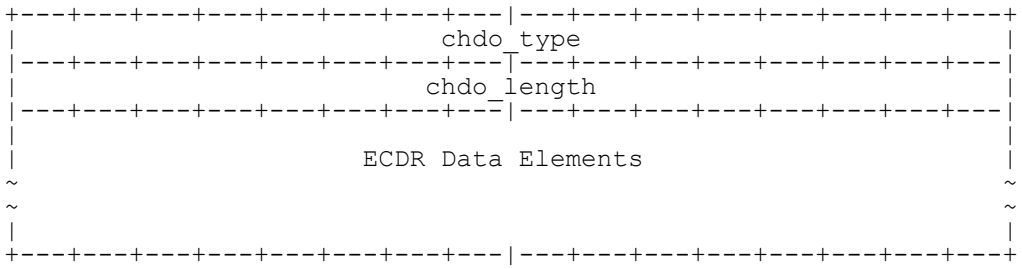
		hundred fifty-five 16-bit words. If this field contains a channel value, flag lv_flag must be 0 and length_value must contain the number of 16-bit words used to store the channel value. If flag lv_flag = 1, then this field does not exist for this channel.

Note: Bytes 0-3 are record oriented (once per record). Bytes 4-9 are channel oriented; i.e., entries are on a per channel basis. There can be "n" channels per record, where "n" is determined by the internal data structure and the decommutation information provided for that record.

Return to Contents Links

Data Block Header - Expanded Channelized Data (ECDR) (29)

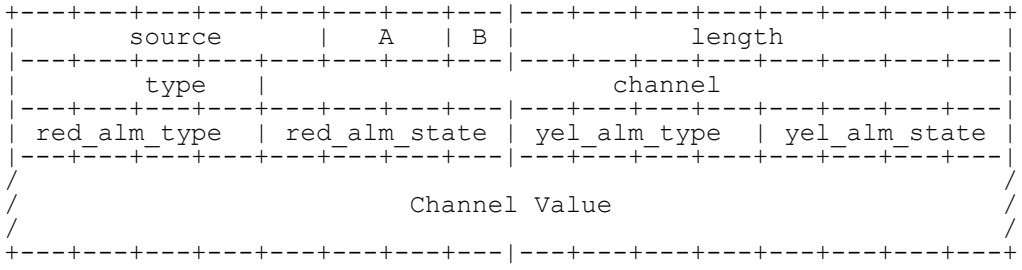
The ECDR Data CHDO consists of a type field, a length field, and a value field consisting of a variable number of ECDR Data Elements.



Byte Offset	Field ID	Description
0-1	chdo_type:	value = 29
2-3	chdo_length:	Length in bytes of this structure (remainder of header).
4-end	ECDR Elements:	A variable number of ECDR Data Elements (subject to SFDU length restrictions).

ECDR Data Element

For each channel that occurs in, or is derived from, a given input CDR SFDU, there will be an ECDR Data Element placed in the data portion of the output ECDR sfdu. The ECDR Data Element structure is:



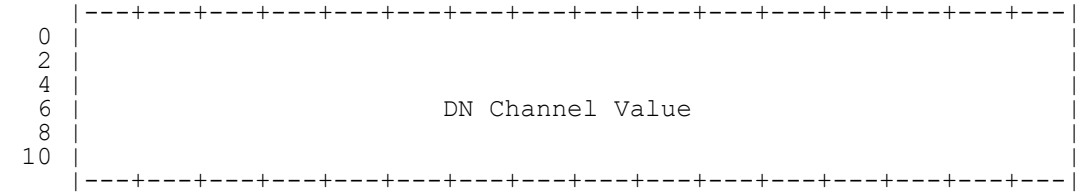
Note: yel is the abbreviation for yellow.

Byte Offset	Field ID	Description
0		BITS 0-4 source - represents the single alphabetic character that is used in a channel name, i.e., the "E" in the channel identifier E- 1024. Source is actually represented here by a number where 1 stands for A, 2 stands for B, etc. X, Y, and Z are not valid for use in a channel name. Valid source values are in the integer range 1- 31, inclusive. Values 24-31 are used to represent slash channels defined in SFOC-1-SYS-Any-ChannelID (Applicable Document #4b). A Bits 5 and 6 are spare. B eu_present (Bit 7). If the value is 0, the val field contains only the Data Number (DN) value of the channel, i.e., the value of the channel in the decommutated data. If the value of this field is 1, then the Channel Value field contains both a DN value and a channel value that has been converted to Engineering Units (EU).
1	length	number bytes in current ECDR Data Element following this field.
2	type	Bits 0-3. Type indicates the channel type defined in the Channel Parameter Table (CPT) for any given channel. The valid values are the integers 1 through 6: 1 for integer, 2 for unsigned, 3 for digital, 4 for status, 5 for float, and 6 for ASCII.
2-3	channel	Bits 4-15 of the 16 Bit Word. Channel number of channel, i.e., for a channel E-1024, the channel number is 1024. Valid channel numbers are integers in the range 0-4095.
4	red_alm_type	Bits 0-3. This field indicates how DMD checks for red alarms. The valid range is integers 0-5. The alarm types are: Type Value Definition NULL 0 No red alarm checking was done for this channel. MASK 1 Mask alarm types are discussed in User's Guide. (Digital & Status Channels only). LOW 1 The channel goes into low alarm if the channel value is lower than a specified value. (Note this is a duplicated value because the channel types that use the two values are

		different.) HIGH 2 If the channel value rises above a specified value, the channel goes into alarm. INCLUSIVE 3 A channel will go into alarm if the channel value is between a high and low limit. EXCLUSIVE 4 A channel will go into alarm if the channel value goes outside the range of previously set high and low limits. CHANGE 5 A channel will go into alarm if the channel value changes from the previous value. The channel type determines which alarm types are valid (see User's Guide for more discussion on alarm types). The legal alarms for each channel type are: **Channel Type** — **Alarm Types** Int, Unsigned Int, Float — OFF (NULL), LOW, HIGH, INCLUSIVE, EXCLUSIVE, CHANGE Digital, Status — OFF (NULL), MASK, CHANGE ASCII — OFF (NULL), CHANGE
4	red_alm_state:	Bits 4-7. An alarm state indicates the condition of the channel based on its current value, the alarm type, and alarm criteria (limits, hysteresis). Value Definition 0 Null. The channel is not in alarm. A channel of any alarm type may be in this state. 1 Mask or low. For DIGITAL or STATUS channels, the channel value met the mask alarm criteria. For INTEGER, UNSIGNED, or FLOAT channel types, the alarm indicates a value lower than the alarm limit. Channels with alarm types of mask, low, or exclusive, may be in this state. 2 High. Indicates a channel value greater than the high alarm limit. Channels of alarm types high or exclusive may be in this state. 3 Inclusive. Indicates a channel value between the alarm high and low limits. Only a channel of alarm type inclusive can be in this state. 4 Change. Indicates the channel value has changed with an alarm type of change. Only a channel of alarm type change can be in this state.
5	yellow_alm_type:	Bits 0-3. Yellow Alarm Types are defined exactly the same as the Red Alarm Types above.

5	yellow_alm_state:	Bits 4-7. Yellow Alarm States are defined exactly the same as Red Alarm States above.
6-end	Channel Value:	The Channel Value field contains either a Space Craft Data Number (DN) value or an Engineering Unit (EU), DN value pair. DN values are stored in a variable length field up to 12 bytes long. If the Channel Value field contains an EU and a DN value, the eu_present bit (C) in the ECDR Data Block will be set to 1 (see above). The format of the Channel Value field is defined below.

DN Channel Value Fields



Byte Offset	Field ID	Description
0-11	dn:	This is a variable length field that can contain either an array of up to 12 ASCII characters or a 32-bit signed integer or a 32-bit unsigned integer or a 64-bit IEEE long floating point number.

ASCII channels are stored as ascii characters, left justified and padded on the right with a null character if the length of the ASCII string is an odd number of bytes.

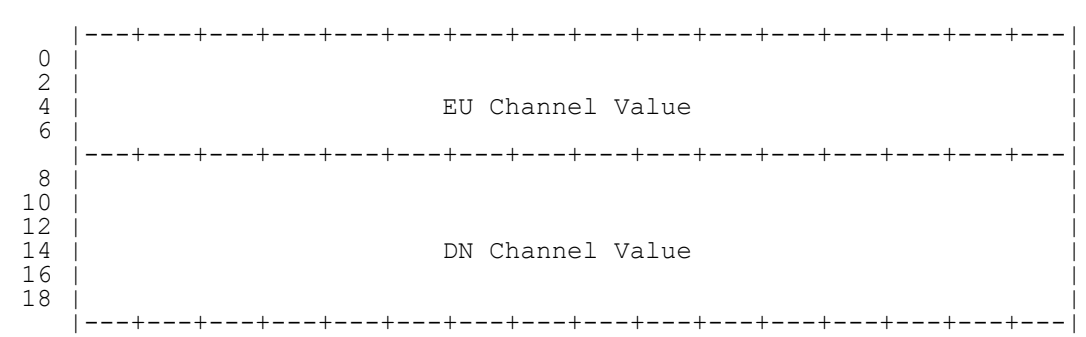
INTEGER channels are stored as 32-bit signed integers, right justified.

DIGITAL, UNSIGNED, and STATUS channels are stored as 32-bit unsigned integers.

FLOAT channels are stored as 64-bit IEEE long floating point numbers.

DN/EU Channel Value Fields

When an EU value is present, it will be placed in the Channel Value field before the DN value. EU Values are always expressed as 64 bit IEEE long floating point numbers. The DN Channel is defined as above.



Return to Contents Links

Galileo Record IDs

Here are the IDs of all the GLL Phase 2 specific records that the reader may find useful.

Note that the most useful engineering (channelized) record is DDP-ID = C657, the most useful (channelized) AACS record is DDP-ID = C658.

The types of CHDOs in each record's SFDU are listed (under "CHDO components"). Thus you can identify each CHDO of each record. Not all CHDOs are in this document, because I assume you will be looking only at packets, sub-packets, and invalid packets, not upstream structures. (But they are available on request.)

Composition of SFOC Standard CHDO-structured Logical SFDUs

DDP ID	Maj Typ	Min Typ	Fmt ID	Msn ID	CHDO Components: Sec	Ter	Qtr	Data	DESCRIPTION
Invalid Packet									
C680	8	128	0	1	48	0	39	10	Invalid Packet, GLL
Engineering Packets/Frames									
C654	2	135	1	1	48	49	---	10	ENG1 R/T Pkt, GLL
C654	2	135	2	1	48	49	---	10	ENG2 P/B Pkt, GLL
C655	2	136	1	1	48	49	---	10	AACS1 R/T UnComp Pkt, GLL
C655	2	136	2	1	48	49	---	10	AACS2 P/B UnComp Pkt, GLL
C681	2	139	1	1	48	49	---	10	AACS3 P/B Comprs Pkt, GLL
C655	2	136	3	1	48	49	---	10	AACS4 RRCC UnComp Pkt, GLL
C655	2	136	4	1	48	49	---	10	AACS3D P/B DeComp Pkt, GLL
C656	2	137	1	1	48	49	42	10	Pkt Eng Frame 2 bps, GLL
C656	2	137	2	1	48	49	42	10	Pkt Eng Frame 10 bps, GLL
C656	2	137	3	1	48	49	42	10	Pkt Eng Frame 40 bps, GLL
C656	2	137	4	1	48	49	42	10	Pkt Eng Frame 1200 bps, GLL
C653	2	138	1	1	48	49	---	10	Sub Pkt AACS R/T Uncomp, GLL
C653	2	138	2	1	48	49	---	10	Sub Pkt AACS P/B Uncomp, GLL
C653	2	138	3	1	48	49	---	10	Sub Pkt AACS RRCC Uncomp, GLL
C653	2	138	4	1	48	49	38	10	Sub Pkt AACS P/B Decomp, GLL
Channelized Engineering Packets/Frames									
C657	11	131	1	1	48	49	27	28	Ch Pkt Eng Frame 2 bps, GLL
C657	11	131	2	1	48	49	27	28	Ch Pkt Eng Frame 10 bps, GLL
C657	11	131	3	1	48	49	27	28	Ch Pkt Eng Frame 40 bps, GLL
C657	11	131	4	1	48	49	27	28	Ch Pkt Eng Frame 1200 bps, GLL
C658	11	132	1	1	48	49	27	28	Ch Sub Pkt AACS R/T Uncomp, GLL
C658	11	132	2	1	48	49	27	28	Ch Sub Pkt AACS P/B Uncomp, GLL
C658	11	132	3	1	48	49	27	28	Ch Sub Pkt AACS RRCC Uncomp, GLL
C658	11	132	4	1	48	49	27	28	Ch Sub Pkt AACS P/B Decomp, GLL
Science Packets									
C675	3	140	1	1	48	49	---	10	DDS1 R/T Pkt, GLL
C675	3	140	2	1	48	49	---	10	DDS2 P/B Pkt, GLL
C675	3	140	3	1	48	49	---	10	DDS3 RRCC Pkt, GLL
C661	3	141	1	1	48	49	---	10	EPD1 R/T Pkt, GLL
C661	3	141	2	1	48	49	---	10	EPD2 P/B Pkt, GLL
C661	3	141	3	1	48	49	---	10	EPD3 RRCC Pkt, GLL
C662	3	142	1	1	48	49	---	10	EUV1 R/T Pkt, GLL
C662	3	142	2	1	48	49	---	10	EUV2 P/B Pkt, GLL
C663	3	143	1	1	48	49	---	10	HIC1 R/T Pkt, GLL
C663	3	143	2	1	48	49	---	10	HIC2 P/B Pkt, GLL
C663	3	143	3	1	48	49	---	10	HIC3 RRCC Pkt, GLL
C664	3	144	1	1	48	49	---	10	MAG1 R/T UnComp Pkt, GLL
C664	3	144	2	1	48	49	---	10	MAG2 P/B UnComp Pkt, GLL
C683	3	156	4	1	48	49	---	10	MAG3 P/B Comprs Pkt, GLL
C664	3	144	3	1	48	49	---	10	MAG4 RRCC UnComp Pkt, GLL
C664	3	144	5	1	48	49	38	10	MAG3D P/B DeComp Pkt, GLL

C665	3	145	1	1	48	49	---	10	NIMS1 R/T UnComp Pkt, GLL
C665	3	145	2	1	48	49	---	10	NIMS2 P/B 11.52K UnComp Pkt, GLL
C665	3	145	3	1	48	49	---	10	NIMS3 P/B 6.168K UnComp Pkt, GLL
C665	3	145	4	1	48	49	---	10	NIMS4 P/B 2.592K UnComp Pkt, GLL
C684	3	157	5	1	48	49	---	10	NIMS5 P/B 11.52K Comprs Pkt, GLL
C684	3	157	6	1	48	49	---	10	NIMS6 P/B 6.168K Comprs Pkt, GLL
C684	3	157	7	1	48	49	---	10	NIMS7 P/B 2.592K Comprs Pkt, GLL
C666	3	146	4	1	48	49	---	10	OPN1 Limb R/T Pkt, GLL
C666	3	146	5	1	48	49	---	10	OPN2 Star R/T Pkt, GLL
C666	3	146	6	1	48	49	---	10	OPN3 Limb P/B Pkt, GLL
C666	3	146	7	1	48	49	---	10	OPN4 Star P/B Pkt, GLL
C667	3	147	1	1	48	49	---	10	PLS1 R/T UnComp Pkt, GLL
C667	3	147	2	1	48	49	---	10	PLS2 P/B UnComp Pkt, GLL
C685	3	158	4	1	48	49	---	10	PLS3 P/B Comprs Pkt, GLL
C667	3	147	3	1	48	49	---	10	PLS4 RRCC UnComp Pkt, GLL
C667	3	147	5	1	48	49	38	10	PLS3D P/B DeComp Pkt, GLL
C668	3	148	1	1	48	49	---	10	PPR1 P/B UnComp Pkt, GLL
C686	3	159	1	1	48	49	---	10	PPR2 P/B Comprs Pkt, GLL
C668	3	148	2	1	48	49	---	10	PPR3 Burst UnComp Pkt, GLL
C686	3	159	2	1	48	49	---	10	PPR4 Burst Comprs Pkt, GLL
C668	3	148	5	1	48	49	38	10	PPR2D P/B DeComp Pkt, GLL
C668	3	148	6	1	48	49	38	10	PPR4D Burst DeComp Pkt, GLL
C669	3	149	4	1	48	49	---	10	PWH1 Fill Pkt, GLL
C669	3	149	1	1	48	49	---	10	PWH2 P/B MPW Pkt, GLL
C669	3	149	2	1	48	49	---	10	PWH3 P/B MPP Pkt, GLL
C669	3	149	3	1	48	49	---	10	PWH4 P/B HPW Pkt, GLL
C669	3	149	6	1	48	49	---	10	PWH5 LPW Golay Pkt, GLL
C670	3	150	1	1	48	49	---	10	PWL1 R/T E-Pkt Comprs Pkt, GLL
C670	3	150	2	1	48	49	---	10	PWL2 R/T B-Pkt Comprs Pkt, GLL
C672	3	152	2	1	48	49	---	10	PWL3 P/B UnComp Pkt, GLL
C672	3	152	3	1	48	49	---	10	PWL4 RRCC UnComp Pkt, GLL
C687	3	160	4	1	48	49	---	10	SSI1 ICT Comprs Pkt, GLL
C687	3	160	5	1	48	49	---	10	SSI2 BARC Comprs Pkt, GLL
C673	3	153	6	1	48	49	---	10	SSI3 Hskp UnComp Pkt, GLL
C674	3	154	1	1	48	49	---	10	UVS1 R/T UnComp Pkt, GLL
C674	3	154	2	1	48	49	---	10	UVS2 P/B UnComp Pkt, GLL
C688	3	161	4	1	48	49	---	10	UVS3 P/B Comprs Pkt, GLL
C674	3	154	5	1	48	49	38	10	UVS3D P/B DeComp Pkt, GLL

Channelized Science Packets/Frames

C689	11	153	0	1	48	49	27	28	Ch Pkt SSI3 Hskp, GLL

QQC Records

C561	13	0	21	1	201	311	---	00	TIS QQC Heartbeat In-Sync, GLL
C562	13	0	22	1	202	312	---	00	TIS QQC Heartbeat Out-Sync,GLL
C563	13	0	23	1	203	313	---	00	TIS QQC Heartbeat No-Data, GLL
C564	13	0	24	1	204	314	---	00	TIS QQC TDM Fail Acq, GLL
C565	13	0	26	1	206	315	---	00	TIS QQC In-Sync, GLL
C566	13	0	27	1	207	316	---	00	TIS QQC Out-of-Sync,GLL
C418	13	0	28	1	208	---	---	00	TIS QQC No-Data, GLL
C419	13	0	29	1	209	---	---	00	TIS QQC SCLK Change, GLL
C568	13	0	30	1	210	318	---	00	TIS QQC FID Change, GLL
C569	13	0	31	1	211	319	---	00	TIS QQC Decommutation, GLL
C422	13	0	32	1	212	---	---	00	TIS QQC Extract Begin, GLL
C423	13	0	33	1	213	---	---	00	TIS QQC Extract End, GLL
C570	13	0	34	1	214	320	---	00	TIS QQC Summary, GLL
C571	13	0	35	1	215	321	---	00	TIS QQC Data Summary, GLL
C503	13	0	38	1	201	311	324	00	TIS QQC Ph II Heartbeat In-Sync, GLL
C504	13	0	39	1	206	315	325	00	TIS QQC Ph II In-Sync, GLL
C594	13	0	36	1	214	322	---	00	TIS QQC Phase II Summary, GLL
C597	13	0	37	1	215	323	---	00	TIS QQC Phase II Data Summary, GLL
C567	13	5	1	1	220	---	---	00	TIS QQC MIPS ICT Status SSI, GLL
C567	13	5	2	1	220	---	---	00	TIS QQC MIPS ICT Status PWS, GLL
C578	13	5	3	1	221	---	---	00	TIS QQC NIMS Rice Status, GLL

Return to Contents Links