JPL D-94650

ECOsystem Spaceborne Thermal Radiometer Experiment on Space Station (ECOSTRESS) Mission

Level 0 Product Specification Document

April 24, 2018

Albert Niessner, Eugene Chu
ECOSTRESS Algorithm Development Team

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, California 91109-8099
California Institute of Technology

ECOSTRESS Level 0 Product Specification Document

Prepared by:

Albert Niessner
ECOSTRESS Algorithm Development Team

Date

Eugene Chu
ECOSTRESS SDS System Engineer

Date

Approved by:

Benjamin Bornstein
ECOSTRESS Mission System Manager

Date

Concurred by:

Simon Hook
ECOSTRESS Principal Investigator

Date

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, California 91109-8099
California Institute of Technology

*** Original signature page on file in project physical repository ***

Document Change Log

Revision	Date	Sections Changed	Author
Preliminary	04/24/2018	All	Eugene Chu

Document TBD, TBR, TBS

Section/Page	Description	Date

Contacts

Readers seeking additional information about this document may contact the following ECOSTRESS Algorithm Development team members:

Albert F. Niessner
MS 158-268
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena CA 91109
Email: Albert.F.Niessner@jpl.nasa.gov
Office: (818) 354-0859

Eugene Y. Chu
MS 168-414
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena CA 91109
Email: Eugene.Y.Chu@jpl.nasa.gov
Office: (818) 354-6496

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 Identification

This is the Product Specification Document (PSD) for Level 0 (L0) data products of the ECOsystem Spaceborne Thermal Radiometer Experiment on Space Station (ECOSTRESS) project. The ECOSTRESS L0 products contain raw data (FLEX) packets and housekeeping (HK) data collected by the ECOSTRESS instrument.

1.2 Purpose and Scope

This Product Specification Document (PSD) describes the contents of the Level 0 FLEX packets, and the housekeeping data required to process the instrument data at the ECOSTRESS SDS at JPL. These include the detailed descriptions of the format and contents of the product and ancillary files that will be delivered to the Land Process Distributed Active Archive Center (LP-DAAC).

1.3 Mission Overview

The ECOSTRESS instrument measures the temperature of plants and uses that information to better understand how much water plants need and how they respond to stress.

ECOSTRESS addresses three overarching science questions:

- How is the terrestrial biosphere responding to changes in water availability?
- How do changes in diurnal vegetation water stress impact the global carbon cycle?
- Can agricultural vulnerability be reduced through advanced monitoring of agricultural water consumptive use and improved drought estimation?

The ECOSTRESS mission answers these questions by accurately measuring the temperature of plants. Plants regulate their temperature by releasing water through tiny pores on their leaves called stomata. If they have sufficient water, they can maintain their temperature. However, if there is insufficient water, their temperatures rise. This temperature rise can be measured with a sensor in space. ECOSTRESS uses a multispectral thermal infrared (TIR) radiometer to measure the surface temperature, deployed on the International Space Station. The instrument will measure radiances at 5 spectral bands in the 8-12.5 μm range with approximately 38 meter by 57 meter of spatial resolution on the ground.

1.4 Applicable and Reference Documents

“Applicable” documents levy requirements on the areas addressed in this document. “Reference” documents are identified in the text of this document only to provide additional information to readers. Unless stated otherwise, the document revision level is Initial Release. Document dates are not listed, as they are redundant with the revision level.

1.4.1 Applicable Documents

1. ECOSTRESS Project Level 3 Science Data System Requirements (JPL D-94088).
2. ECOSTRESS Science Data Management Plan (JPL D-94607)
3. 423-ICD-005 ICD Between ECOSTRESS SDS and LPDAAC

4. ECOSTRESS Level 1 Geolocation Algorithm Theoretical Basis Document (JPL D-94641)
5. ECOSTRESS Level 1 Calibration Algorithm Theoretical Basis Document (JPL D-94642)
6. ECOSTRESS Level 1 Algorithm Specification Document
7. ECOSTRESS Command Dictionary (JPL D-94639)

1.4.2 Reference Documents

8. FLEX ECOSTRESS Science Data Packet Specifications, memo 20161214.

1.5 ECOSTRESS Data Products

The ECOSTRESS mission will generate 13 different distributable data products. The products represent four levels of data processing, with data granules defined as an image scene. Each image scene consists of 44 scans of the instrument mirror, each scan taking approximately 1.181 seconds, and each image scene taking approximately 52 seconds. Each image scene starts at the beginning of the first target area encountered during each orbit. Each orbit is defined as the equatorial crossing of an ascending International Space Stations (ISS) orbit.

ECOSTRESS Level 0 data include spacecraft packets that have been pre-processed by the Ground Data System (GDS). Level 1 products include spacecraft engineering data, the time-tagged raw sensor pixels appended with their radiometric calibration coefficients, the black body pixels used to generate the calibration coefficients, geolocated and radiometrically calibrated at-sensor radiances of each image pixel, the geolocation tags of each pixel, and the corrected spacecraft attitude data. Level 2 products include the land surface temperature and emissivities of each spectral band retrieved from the at-sensor radiance data, and a cloud mask. Level 2 products also appear in image scene granules. Level 3 products contain evapotranspiration data derived from Level 2 data. Level 4 products contain evaporative stress index and water use efficiency derived from Level 3 data.

The ECOSTRESS products are listed in Table 1-1. This document will discuss the Level 0 products.

Table 1-1: ECOSTRESS Distributable Standard Products

Product type	**Description**
L0A_FLEX	Level 0 “raw” spacecraft packets
L0A_HK	Level 0 housekeeping packets
L1A_ENG	Spacecraft and instrument engineering data, including blackbody gradient coefficients
L1A_BB	Instrument Black Body calibration pixels
L1A_PIX	Raw pixel data with appended calibration coefficients
L1B_GEO	Geolocation tags, sun angles, and look angles, and calibrated, resampled at-sensor radiances
L1B_RAD	Radiometrically corrected, band-aligned, squared at-sensor radiance pixels
L1B_ATT	Corrected spacecraft ephemeris and attitude data
L2_LSTE	Land Surface temperature and emissivity
L2_CLOUD	Cloud mask
L3_ET_PT-JPL	Evapotranspiration retrieved from L2_LSTE using the PT-JPL Algorithm
L3_ET_ALEXIA	Evapotranspiration generated at JPL with the ALEXI/DisALEXI Algorithm over agriculture regions (not delivered to DAAC)
L3_ET_ALEXI	Evapotranspiration generated at JPL with the ALEXI/DisALEXI Algorithm over NON-agriculture regions (not delivered to DAAC)
L3_ET_ALEXIU	Evapotranspiration generated at USDA with the ALEXI/DisALEXI Algorithm over specific calibration sites
L4_ESI_PT-JPL	Evaporative Stress Index generated with PT-JPL
L4_ESI_ALEXIA	Evaporative Stress Index generated at JPL with the ALEXI/DisALEXI over agriculture regions (not delivered to DAAC)
L4_ESI_ALEXI	Evaporative Stress Index generated at JPL with the ALEXI/DisALEXI over NON-agriculture regions (not delivered to DAAC)
L4_ESI_ALEXIU	Evaporative Stress Index generated at USDA with the ALEXI/DisALEXI over specific calibration sites
L4_WUE	Water Use efficiency
L3_L4_QA	Quality Assessment fields for all ancillary data used in L3 and L4 products

2.0 DATA PRODUCT ORGANIZATION

2.1 Product File Format

All ECOSTRESS standard products are in the Hierarchical Data Format version 5 (HDF5). HDF5 is a general purpose file format and programming library for storing scientific data. The National Center for Supercomputing Applications (NCSA) at the University of Illinois developed HDF to help scientists share data regardless of the source. The following sections provide some key elements of HDF5 that will be employed in ECOSTRESS data products. Complete documentation of the HDF5 structure and application software can be found at http://www.hdfgroup.org/HDF5

2.2 HDF5 Notation

The key concepts of the HDF5 Abstract Data Model are Files, Groups, Datasets, Datatypes, Attributes and Property Lists. The following sections provide a brief description of each of these key HDF5 concepts.

2.2.1 HDF5 File

A File is the abstract representation of a physical data file. Files are containers for HDF5 Objects. These Objects include Groups, Datasets, and Datatypes.

2.2.2 HDF5 Group

Groups are containers for other Objects, including Datasets, named Datatypes and other Groups. In that sense, groups are analogous to directories that are used to categorize and classify files in standard operating systems.

The notation for files is identical to the notation used for Unix directories. The root Group is "/". Like Unix directories, Objects appear in Groups through "links". Thus, the same Object can simultaneously be in multiple Groups.

2.2.3 HDF5 Dataset

The Dataset is the HDF5 component that stores user data. Each Dataset associates with a Dataspace that describes the data dimensions, as well as a Datatype that describes the basic unit of storage element. A Dataset can also have Attributes.

2.2.4 HDF5 Datatype

A Datatype describes a unit of data storage for Datasets and Attributes. Datatypes are subdivided into Atomic and Composite Types.

Atomic Datatypes are analogous to simple basic types in most programming languages. HDF5 Atomic Datatypes include Time, Bitfield, String, Reference, Opaque, Integer, and Float. Each atomic type has a specific set of properties. Examples of the properties associated with Atomic Datatypes are:

- Integers are assigned size, precision, offset, pad byte order, and are designated as signed or unsigned.
- Strings can be fixed or variable length, and may or may not be null-terminated.
- References are constructs within HDF5 Files that point to other HDF5 Objects in the same file.

HDF5 provides a large set of predefined Atomic Datatypes. Table 2-1 lists the Atomic Datatypes that are used in ECOSTRESS data products.

Table 2-1: HDF5 Atomic Datatypes

HDF5 Atomic Datatypes	Description
H5T_STD_U8LE	unsigned, 8-bit, little-endian integer
H5T_STD_U16LE	unsigned, 16-bit, little-endian integer
H5T_STD_U32LE	unsigned, 32-bit, little-endian integer
H5T_STD_U64LE	unsigned, 64-bit, little-endian integer
H5T_STD_I8LE	signed, 8-bit, little-endian integer
H5T_STD_I16LE	signed, 16-bit, little-endian integer
H5T_STD_I32LE	signed, 32-bit, little-endian integer
H5T_STD_I64LE	Signed, 64-bit, little-endian integer
H5T_IEEE_F32LE	32-bit, little-endian, IEEE floating point
H5T_IEEE_F64LE	64-bit, little-endian, IEEE floating point
H5T_STRING	character string made up of one or more bytes

Composite Datatypes incorporate sets of Atomic datatypes. Composite Datatypes include Array, Enumeration, Variable Length and Compound.

- The Array Datatype defines a multi-dimensional array that can be accessed atomically.
- Variable Length presents a 1-D array element of variable length. Variable Length Datatypes are useful as building blocks of ragged arrays.

Named Datatypes are explicitly stored as Objects within an HDF5 File. Named Datatypes provide a means to share Datatypes among Objects. Datatypes that are not explicitly stored as Named Datatypes are stored implicitly. They are stored separately for each Dataset or Attribute they describe.

None of the ECOSTRESS data products employ Enumeration or Compound data types.

2.2.5 HDF5 Dataspace

A Dataspace describes the rank and dimension of a Dataset or Attribute. For example, a "Scalar" Dataspace has a rank of 1 and a dimension of 1. Thus, all subsequent references to "Scalar" Dataspace in this document imply a single dimensional array with a single element.

Dataspaces provide considerable flexibility to HDF5 products. They incorporate the means to subset associated Datasets along any or all of their dimensions. When associated with specific properties, Dataspaces also provide the means for Datasets to expand as the application requires.

2.2.6 HDF5 Attribute

An Attribute is a small aggregate of data that describes Groups or Datasets. Like Datasets, Attributes are also associated with a particular Dataspace and Datatype. Attributes cannot be subsetted or extended. Attributes themselves cannot have Attributes.

2.3 ECOSTRESS File Organization

2.3.1 Structure

ECOSTRESS data products follow a common convention for all HDF5 Files. Use of this convention provides uniformity of data access and interpretation.

The ECOSTRESS Project uses HDF5 Groups to provide an additional level of data organization. All metadata that pertain to the complete data granule are members of the "/Metadata" Group. All other data are organized within Groups that are designed specifically to handle the structure and content of each particular data product.

2.3.2 Data

All data in HDF5 files are stored in individual Datasets. All related Datasets in an ECOSTRESS product are assigned to an HDF5 Group. A standard field name is associated with each Dataset. The field name is a unique string identifier. The field name corresponds to the name of the data element the Dataset stores. This document lists these names with the description of each data element that they identify.

Each Dataset is associated with an HDF5 Dataspace and an HDF5 Datatype. They provide a minimally sufficient set of parameters for reading the data using standard HDF5 tools.

2.3.3 Element Types

ECOSTRESS HDF5 employs the Data Attribute "Type" to classify every data field as a specific data type. The "Type" is an embellishment upon the standard HDF5 Datatypes that is designed specifically to configure ECOSTRESS data products.

Table 2-2 lists all of the "Type" strings that appear in the ECOSTRESS data products. The table maps each ECOSTRESS "Type" to a specific HDF5 Datatype in both the HDF5 file and in the data buffer. The table also specifies the common conceptual data type that corresponds to the "Type" in ECOSTRESS executable code.

Table 2-2: Element Type Definitions

Type	HDF5 Datatype (File)	HDF5 Datatype (Buffer)	Conceptual Type
Unsigned8	H5T_STD_U8LE	H5T_NATIVE_UCHAR	unsigned integer
Unsigned16	H5T_STD_U16LE	H5T_NATIVE_USHORT	unsigned integer
Unsigned32	H5T_STD_U32LE	H5T_NATIVE_UINT	unsigned integer
Unsigned64	H5T_STD_U64LE	H5T_NATIVE_ULLONG	unsigned integer
Signed8	H5T_STD_I8LE	H5T_NATIVE_SCHAR	signed integer
Signed16	H5T_STD_I16LE	H5T_NATIVE_SHORT	signed integer
Signed32	H5T_STD_I32LE	H5T_NATIVE_INT	signed integer
Signed64	H5T_STD_I64LE	H5T_NATIVE_LLONG	signed integer
Float32	H5T_IEEE_F32LE	H5T_NATIVE_FLOAT	floating point
Float64	H5T_IEEE_F64LE	H5T_NATIVE_DOUBLE	floating point
VarLenStr	H5T_STRING	H5T_NATIVE_CHAR	character string

2.3.4 File Level Metadata

All metadata that describe the full content of each granule of the ECOSTRESS data product are stored within the explicitly named "/Metadata" Group. Metadata are handled using exactly the same procedures as those that are used to handle data. The contents of each Attribute that stores metadata conform to one of the ECOSTRESS Types. Most metadata elements are stored as scalars. A few metadata elements are stored as arrays. The metadata appear in a set of HDF5 Groups under the "/Metadata" Group. These HDF5 Groups contain a set of HDF5 Attributes.

2.3.5 Local Metadata

ECOSTRESS standards incorporate additional metadata that describe each HDF5 Dataset within the HDF5 file. Each of these metadata elements appear in an HDF5 Attribute that is directly associated with the HDF5 Dataset. Wherever possible, these HDF5 Attributes employ names that conform to the Climate and Forecast (CF) conventions. Table 2-3 lists the CF names for the HDF5 Attributes that ECOSTRESS products typically employ.

Table 2-3: ECOSTRESS Specific Local Attributes

CF Compliant Attribute Name	Description	Required?
units	Units of measure. Appendix A lists applicable units for various data elements in this product.	Yes
valid_max	The largest valid value for any element in the Dataset. The data type in valid_max matches the type of the associated Dataset. Thus, if the associated Dataset stores float32 values, the corresponding valid_max will also be float32.	No
valid_min	The smallest valid value for any element in the Dataset. The data type in valid_min matches the type of the associated Dataset. Thus, if the associated Dataset stores float32 values, the corresponding valid_min will also be float32.	No
_FillValue	Specification of the value that will appear in the Dataset when an element is missing or undefined. The data type of _FillValue matches the type of the associated Dataset. Thus, if the associated Dataset stores float32 values, the corresponding _FillValue will also be float32.	Yes for all numeric data types
long_name	A descriptive name that clearly describes the content of the associated Dataset.	Yes

2.4 Data Definition Standards

The following sections of this document specify the characteristics and definitions of every data element stored in the ECOSTRESS data products. Table 2-4 defines each of the specific characteristics that are listed in those sections. Some of these characteristics correspond with the ECOSTRESS HDF5 Attributes that are associated with each Dataset. Data element characteristics that correspond to ECOSTRESS HDF5 Attributes bear the same name. The remaining characteristics are descriptive data that help users better understand the data product content.

In some situations, a standard characteristic may not apply to a data element. In those cases, the field contains the character string ’n/a’. Hexadecimal representation sometimes indicates data content more clearly. Numbers represented in hexadecimal begin with the character string ’0x’.

Table 2-4: Data Element Characteristic Definitions

Characteristic	Definition
Type	The data representation of the element within the storage medium. The storage class specification must conform to a valid ECOSTRESS type.
Units	Units of measure. Typical values include “deg”, “degC”, “Kelvin”, “meters/second”, “meters”, “m**2”, “seconds” and “counts”. Appendix A includes references to important data measurement unit symbols.

2.4.1 Double Precision Time Variables

ECOSTRESS double precision time variables contain measurements relative to the J2000 epoch. Thus, these variables represent a real number of Standard International (SI) compatible seconds since 11:58:55.816 on January 1, 2000 UTC.

2.4.2 Array Representation

This document employs array notation to demonstrate and clarify the correspondence among data elements in different product data elements. The array notation adopted in this document is similar to the standards of the Fortran programming language. Indices are one based. Thus, the first index in each dimension is one. This convention is unlike C or C++, where the initial index in each dimension is zero. In multidimensional arrays, the leftmost subscript index changes most rapidly. Thus, in this document, array elements ARRAY(15,1,5) and ARRAY(16,1,5) are stored contiguously.

HDF5 is designed to read data seamlessly regardless of the computer language used to write an application. Thus, elements that are contiguous using the dimension notation in this document will appear in contiguous locations in arrays for reading applications in any language with an HDF5 interface.

This document differentiates among array indices based on relative contiguity of storage of elements referenced with consecutive numbers in that index position. A faster or fastest moving index implies that the elements with consecutive numbers in that index position are stored in relative proximity in memory. A slower or slowest moving index implies that the elements referenced with consecutive indices are stored more remotely in memory. For instance, given array element ARRAY(15,1,5) in Fortran, the first index is the fastest moving index and the third index is the slowest moving index. On the other hand, given array element array[4][0][14] in C, the first index is the slowest moving index and the third index is the fastest moving index.

3.0 ECOSTRESS PRODUCT FILES

Each ECOSTRESS product file will contain at least 3 groups of data: A standard metadata group that specifies the same type of contents for all products, a product specific metadata group that specifies those metadata elements that are useful for defining attributes of the product data, and the group(s) containing the product data. (Note: A product metadata is not to be confused with a HDF5 object metadata.)

L0A product file names will have the form:

```
ECOSTRESS_<PROD_TYPE>_<OOOOO>_<YYYYMMDD>T<hhmmssmmm>_<YYYYMMDD>T<hhmmssmmm
>_<BBbb>_<VV>.<TYPE>
```

Higher level product file names will have the form:

```
L0B_<OOOOO>_<SSS>_<YYYYMMDD>T<hhmmss>_<BBbb>_<VV>.<TYPE>
```

Where:

PROD_TYPE: Product type =

L0A_FLEX, Raw instrument data packets (non-distributed)
L0A_HK, Raw instrument engineering and housekeeping packets (non-distributed)
L1A_PIX, Time-tagged, image frames formed from L0A_FLEX packets
L1A_BB, Calibration black body pixels recorded from instrument with each image frame
L1A_ENG, Orbital engineering data
L1B_RAD, Calibrated at-sensor radiance image frames
L1B_GEO, Geolocation parameters of image frames
L1B_ATT, Refined spacecrafts orbital attitude and ephemeris parameters
L2_LSTE, Land surface Temperature and Emissivity data
L2_CLOUD, Level 2 Cloud mask data
L3_ET_PT-JPL, Evapotranspiration generated by JPL with PT-JPL
L3_ET_ALEXI, Evapotranspiration generated by JPL with ALEXI/DisALEXI
L3_ET_ALEXI-USDA, Evapotranspiration generated by USDA with ALEXI/DisALEXI
L4_ESI_PT-JPL, Evaporative Stress Index generated by JPL with PT-JPL
L4_ESI_ALEXI, Evaporative Stress Index generated by JPL with ALEXI/DisALEXI
L4_ESI_ALEXI-A, JPL ALEXI/DisALEXI ESI product over Agriculture Regions
L4_ESI_ALEXI-USDA, Evaporative Stress Index generated by USDA with ALEXI/DisALEXI
L4_WUE, Water Use Efficiency generated by JPL
L3_L4_QA, Quality Assessment fields for all ancillary data used in L3 and L4 products generated by JPL

L0B, All L0A FLEX and HK packets of a single orbit packed into a single file

OOOOO: Orbit number; starting at start of mission, ascending equatorial crossing
<YYYYMMDD>T<hhmmssmmm>: Starting and ending times of FLEX packets in Year, Month, day, hours, minutes, seconds, milliseconds (L0A)
YYYYMMDD: Year, month, day of scene start time (L0B)
hhmmss: Hour, minute, second of scene start time (L0B)
SSS: Scene ID; starting at first scene of each orbit (L0B)
BBbb: Build ID of software that generated product, Major+Minor (2+2 digits)
VV: Product version number (2 digits)
TYPE: File type extension=
h5 for the data file
h5.xml for the metadata file.

3.1 Standard Metadata

This is the minimal set of metadata that must be included with each product file. The standard metadata consists of the following:

Table 3-1: Standard Product Metadata

Name	**Type**	**Size**	**Example**
Group	**StandardMetadata**		
AncillaryInputPointer	String	variable	Group name of ancillary file list
AutomaticQualityFlag	String	variable	PASS/FAIL (of product data)
BuildID	String	variable	
CampaignShortName	String	variable	Primary
CollectionLabel	String	variable	
DataFormatType	String	variable	NCSAHDF5
DayNightFlag	String	variable	
EastBoundingCoordinate	LongFloat	8	
HDFVersionID	String	variable	1.8.16
ImageLines	Int32	4	11264 (5632, 5400)
ImageLineSpacing	Float32	4	68.754
ImagePixels	Int32	4	5400 (128 for BB)
ImagePixelSpacing	Float32	4	65.536
InputPointer	String	variable	
InstrumentShortName	String	variable	ECOSTRESS
LocalGranuleID	String	variable	
LongName	String	variable	ECOSTRESS
NorthBoundingCoordinate	LongFloat	8	
PGEName	String	variable	L0A, L0B
PGEVersion	String	variable	
PlatformLongName	String	variable	ISS
PlatformShortName	String	variable	ISS
PlatformType	String	variable	Spacecraft
ProcessingLevelID	String	variable	1
ProcessingLevelDescription	String	variable	Level 0
ProducerAgency	String	variable	JPL
ProducerInstitution	String	variable	Caltech
ProductionDateTime	String	variable	
ProductionLocation	String	variable	
RangeBeginningDate	String	variable	
RangeBeginningTime	String	variable	
RangeEndingDate	String	variable	
RangeEndingTime	String	variable	
RegionID	String	variable	
SceneID	String	variable	
ShortName	String	variable	L0A_FLEX, L0A_HK, L0B
SISName	String	variable	
SISVersion	String	variable	
SouthBoundingCoordinate	LongFloat	8	
StartOrbitNumber	String	variable	
StopOrbitNumber	String	variable	
WestBoundingCoordinate	LongFloat	8	

3.2 Product-Specific Metadata

Any additional metadata necessary for describing the product will be recorded in this group.

Table 3-2: Product Specific Metadata

Name	Type	Size	Example
Group	**L0A_FLEX Metadata (L0A_HK Metadata)**		

…

3.3 Product Data

The product data will be stored in the following groups.

The L0A products come from the GDS. They consist of variable numbers of files, each containing variable numbers of FPIE data packets.

3.3.1 L0A_FLEX – L0 instrument FLEX packets

Table 3-1: L0A FLEX Data Definitions

Field Name	Type	Units	Field Data
Group	**flex** (Size Unlimited x 197188 bytes per packet)		
FSW_TIME	Float64	Sec	Flight software time when packet was recorded
Packet	Uint8	bits	Raw FLEX packets recorded from instrument (197180 bytes of various units)

Raw FLEX packets contain 197180 bytes of data captured from the instrument. See Reference Document 8 in section 1.4.2 for contents details.

3.3.2 L0A_HK –L0 House Keeping packets

Table 3-2: L0A House Keeping Data Definitions

Field Name	Type	Units	Field Data
Group	**hk** (Size Unlimited x variable, up to 2.5k)		
FSW_TIME	Float64	Second	Flight software time when packet was recorded
LENGTH	UInt32		HK packets can vary in length but HDF arrays must be rectangular. This length is the number of valid bytes in each row of the Packet array.
Packet	Uint8	bits	HK data (up to 2 kbytes per packet) see Appendix A
Type	Uint8		Each HK packet is identified with a unique byte. This is the identification byte for each row in the Packet array

House keeping packets contain data recorded from the Broadcast Ancillary Data stream from the ISS. One packet is recorded every second. See Reference Document 9 in section 1.4.2 for contents details.

3.4 Product Metadata File

The product metadata for each product file will be generated by the PCS from the metadata contents of each product file. The metadata will be converted into extensible markup language (XML). These will be used by the DAAC for cataloging.

APPENDIX A: HK PACKET CONTENTS

```
PACKET hAndS Product (fixed-size, 2096 bytes):
    0: uint32 PacketLength
    4: uint32 ProductSync
    8: time4s4ns HAndSProductTime
    16: uint8 ProductType
    17: uint8 ProductLength[0], array has 3 elements
    20: uint32 ProductDetails[0], array has 3 elements
    32: time4s4ns AnalogsSampleTime
    40: uint16 AnalogsVoltage_PCE_V_GB_V15
    42: uint16 AnalogsVoltage_PCE_V_GB
    44: uint16 AnalogsVoltage_PCE_V_GD
    46: uint16 AnalogsVoltage_PCE_V_GA
    48: uint16 AnalogsVoltage_PCE_V_GC
    50: uint16 AnalogsVoltage_PCE_VCC_5V
    52: uint16 AnalogsVoltage_MC_TEMP5
    54: uint16 AnalogsVoltage_MC_MTR_CUR_1
    56: uint16 AnalogsVoltage_MC_PHASE_VOLT_W_1
    58: uint16 AnalogsVoltage_MC_PHASE_VOLT_V_1
    60: uint16 AnalogsVoltage_MC_PHASE_VOLT_U_1
    62: uint16 AnalogsVoltage_MC_TEMP3
    64: uint16 AnalogsVoltage_MC_TEMP2
    66: uint16 AnalogsVoltage_MC_TEMP1
    68: uint16 AnalogsVoltage_MC_TEMP4
    70: uint16 AnalogsVoltage_MC_V5_TEMP
    72: uint16 AnalogsVoltage_MC_TEMP6
    74: uint16 AnalogsVoltage_MC_TEMP7
    76: uint16 AnalogsVoltage_WEBA_MOXA1_PWR_ON
    78: uint16 AnalogsVoltage_WEBA_MOXA2_PWR_ON
    80: uint16 AnalogsVoltage_WEBA_MOXA1_FAULT_RED
    82: uint16 AnalogsVoltage_WEBA_MOXA2_FAULT_RED
    84: uint16 AnalogsVoltage_WEBA_MOXA1_STATE_RED
    86: uint16 AnalogsVoltage_WEBA_MOXA2_STATE_RED
    88: uint16 AnalogsVoltage_WEBA_MOXA1_STATE_GREEN
    90: uint16 AnalogsVoltage_WEBA_MOXA2_STATE_GREEN
    92: uint16 AnalogsVoltage_WEBA_MOXA1_WLAN_GREEN
    94: uint16 AnalogsVoltage_WEBA_MOXA2_WLAN_GREEN
    96: uint16 AnalogsVoltage_WEBA_MOXA1_LAN_YELLOW
    98: uint16 AnalogsVoltage_WEBA_MOXA2_LAN_YELLOW
    100: uint16 AnalogsCurr_MSU_1
    102: uint16 AnalogsCurr_CC2
    104: uint16 AnalogsCurr_HTR
    106: uint16 AnalogsCurr_CCM_CTL
    108: uint16 AnalogsCurr_FPIE
    110: uint16 AnalogsCurr_MSU2
    112: uint16 AnalogsCurr_CC3
    114: uint16 AnalogsCurr_CEU
    116: uint16 AnalogsCurr_CC1
    118: uint16 AnalogsCurr_MC
    120: uint16 AnalogsCurr_REG
    122: uint16 AnalogsCurr_MOXA
    124: uint16 AnalogsTemp_CAL_RESISTOR_HIGH_1
    126: uint16 AnalogsTemp_FPA_MOLY_PLATE_1
    128: uint16 AnalogsTemp_FPA_MOLY_PLATE_2
    130: uint16 AnalogsTemp_FPA_STRAP_FPA_SIDE_1
    132: uint16 AnalogsTemp_FPA_STRAP_FPA_SIDE_2
    134: uint16 AnalogsTemp_FPA_CRYO_2_COLD_TIP
    136: uint16 AnalogsTemp_FPA_STRAP_COOLER_SIDE_2
    138: uint16 AnalogsTemp_FPA_CS_STRAP_SHIELD_SIDE
    140: uint16 AnalogsTemp_FPA_CRYO_3_COLD_TIP
    142: uint16 AnalogsTemp_FPA_MOORE_MOUNT
    144: uint16 AnalogsTemp_FPA_COLD_BAFFLE_STOP
    146: uint16 AnalogsTemp_FPA_COLD_SHIELD_1
    148: uint16 AnalogsTemp_FPA_COLD_SHIELD_2
    150: uint16 AnalogsTemp_FPA_COLD_SHIELD_3
    152: uint16 AnalogsTemp_FPA_CRYO_1_COLD_TIP
    154: uint16 AnalogsTemp_CAL_RESISTOR_LOW_1
    156: uint16 AnalogsTemp_CAL_RESISTOR_HIGH_2
    158: uint16 AnalogsTemp_BB_HOT_5
    160: uint16 AnalogsTemp_BB_HOT_4
    162: uint16 AnalogsTemp_BB_HOT_3
    164: uint16 AnalogsTemp_BB_HOT_2
    166: uint16 AnalogsTemp_BB_HOT_1
    168: uint16 AnalogsTemp_BB_COLD_5
    170: uint16 AnalogsTemp_BB_COLD_4
    172: uint16 AnalogsTemp_BB_COLD_3
    174: uint16 AnalogsTemp_BB_COLD_2
    176: uint16 AnalogsTemp_BB_COLD_1
    178: uint16 AnalogsTemp_BULKHEAD_M1
    180: uint16 AnalogsTemp_CONTAMINATION_ENCL
    182: uint16 AnalogsTemp_NADIR_PANEL
    184: uint16 AnalogsTemp_ZENITH_PANEL
    186: uint16 AnalogsTemp_CAL_RESISTOR_LOW_2
    188: uint16 AnalogsTemp_CC2_HX_COMP
    190: uint16 AnalogsTemp_CC2_HX_EXP
    192: uint16 AnalogsTemp_CC1_HX_COMP
    194: uint16 AnalogsTemp_CC1_HX_EXP
    196: uint16 AnalogsTemp_CC3_HX_COMP
    198: uint16 AnalogsTemp_CC3_HX_EXP
    200: uint16 AnalogsTemp_PCE_UMS
    202: uint16 AnalogsTemp_MSU_1_UMS
    204: uint16 AnalogsTemp_MSU_2_UMS
    206: uint16 AnalogsTemp_FPIE_UMS
    208: uint16 AnalogsTemp_CCE1_UMS_SN146
    210: uint16 AnalogsTemp_CCE2_UMS_SN147
    212: uint16 AnalogsTemp_CCE3_UMS_SN148
    214: uint16 AnalogsTemp_CCE4_UMS_SN143
    216: uint16 AnalogsTemp_CCE5_UMS_SN144
    218: uint16 AnalogsTemp_CCE6_UMS_SN145
    220: uint16 AnalogsTemp_PCE_TS2
    222: uint16 AnalogsTemp_PCE_TS1
    224: uint16 AnalogsTemp_PCE_TS4
    226: uint16 AnalogsTemp_PCE_TS3
    228: uint16 AnalogsTemp_WEBA_PRT1
    230: uint16 AnalogsTemp_WEBA_PRT2
    232: uint16 AnalogsTemp_WEBA_UMS
    234: uint16 AnalogsTemp_CCM_UMS
    236: uint16 AnalogsTemp_YOKE_PY
    238: uint16 AnalogsTemp_YOKE_NY
    240: uint16 AnalogsTemp_SCAN_MOTOR_1
    242: uint16 AnalogsTemp_CP_ISS_TSTAT
    244: uint16 AnalogsTemp_HRS_CP_INLET
    246: uint16 AnalogsTemp_HRS_OUTLET
    248: uint16 AnalogsTemp_CEU_CHASSIS
    250: uint16 AnalogsTemp_CEU_UMS
    252: uint16 CmdCmdsRcvd
    254: uint16 CmdCmdsFailed
    256: uint16 CmdCmdsExec
    258: time4s4ns CmdHist0Time
    266: uint16 CmdHist0Opcode
    268: uint8 CmdHist0Args[0], array has 8 elements
    276: time4s4ns CmdHist1Time
    284: uint16 CmdHist1Opcode
    286: uint8 CmdHist1Args[0], array has 8 elements
    294: time4s4ns CmdHist2Time
    302: uint16 CmdHist2Opcode
    304: uint8 CmdHist2Args[0], array has 8 elements
    312: time4s4ns CmdHist3Time
    320: uint16 CmdHist3Opcode
    322: uint8 CmdHist3Args[0], array has 8 elements
    330: time4s4ns CmdHist4Time
    338: uint16 CmdHist4Opcode
    340: uint8 CmdHist4Args[0], array has 8 elements
    348: time4s4ns CmdHist5Time
    356: uint16 CmdHist5Opcode
    358: uint8 CmdHist5Args[0], array has 8 elements
    366: time4s4ns CmdHist6Time
    374: uint16 CmdHist6Opcode
    376: uint8 CmdHist6Args[0], array has 8 elements
    384: time4s4ns CmdHist7Time
    392: uint16 CmdHist7Opcode
    394: uint8 CmdHist7Args[0], array has 8 elements
    402: time4s4ns CmdHist8Time
    410: uint16 CmdHist8Opcode
    412: uint8 CmdHist8Args[0], array has 8 elements
    420: time4s4ns CmdHist9Time
    428: uint16 CmdHist9Opcode
    430: uint8 CmdHist9Args[0], array has 8 elements
    438: time4s4ns CmdHist10Time
    446: uint16 CmdHist10Opcode
    448: uint8 CmdHist10Args[0], array has 8 elements
    456: time4s4ns CmdHist11Time
    464: uint16 CmdHist11Opcode
    466: uint8 CmdHist11Args[0], array has 8 elements
    474: time4s4ns CmdHist12Time
    482: uint16 CmdHist12Opcode
    484: uint8 CmdHist12Args[0], array has 8 elements
    492: time4s4ns CmdHist13Time
    500: uint16 CmdHist13Opcode
    502: uint8 CmdHist13Args[0], array has 8 elements
    510: time4s4ns CmdHist14Time
    518: uint16 CmdHist14Opcode
    520: uint8 CmdHist14Args[0], array has 8 elements
    528: time4s4ns CmdNextSchedSeqTime
    536: uint16 CmdNextSchedSeqID
    538: uint32 CmdNumDefinedSeqs
    542: time4s4ns CmdCurrSeqNextCmdTime
    550: uint16 CmdCurrSeqID
    552: uint16 CmdCurrSeqCmdOffset
    554: uint16 CmdCurrSeqLastDeferredCmd
    556: uint8 CmdOverrideEnabled
    557: uint16 CmdNumSequenceProducts
    559: uint16 CmdNumScheduleTableProducts
    561: uint32 CmdNumLowRateBadataProducts
    565: uint32 CmdNumHighRateBadataProducts
    569: uint32 CoreCurrentOpMode
    573: uint32 CoreMemBytesAlloc
    577: uint16 CoreMeanCPUPct
    579: uint16 CorePeakCPUPct
    581: uint8 CorePeakCPUSlot
    582: uint32 CoreX1553Bytes
    586: uint32 CoreEthernetBytes
    590: uint32 CoreStkUsage
    594: uint16 CoreNumDumpProducts
    596: uint16 CoreNumLogProducts
    598: uint32 CoreStagingStatus
    602: uint32 CoreStagingBytesUsed
    606: uint16 CoreDPUSBCV5Temp
    608: uint16 CoreFPIEV5Temp
    610: uint16 CoreDPUIOV5Temp
    612: uint8 CoreDPUIOMode
    613: uint32 CoreFSWVersion
    617: uint32 CoreFSWAddr
    621: uint32 CoreDPUSBCProASICVersion
    625: uint32 CoreDPUSBCV5Version
    629: uint32 CoreDPUSBCV5Addr
    633: uint8 CoreDPUSBCV5SBECount
    634: uint8 CoreDPUSBCV5DBEFlag
    635: uint32 CoreDPUIOProASICVersion
    639: uint32 CoreHKUProASICVersion
    643: uint32 Cryo1PowerUpCycles
    647: float64 Cryo1TempAmb
    655: float64 Cryo1VSensor
    663: float64 Cryo1VAC
    671: float64 Cryo1VDC
    679: float64 Cryo1CurrentDC
    687: float64 Cryo1SetPoint
    695: float64 Cryo1PGain
    703: float64 Cryo1IGain
    711: uint32 Cryo1ReadyWindow
    715: uint32 Cryo1RemoteReady
    719: float64 Cryo1OutputVNomTemp
    727: float64 Cryo1FreqOfACOutput
    735: float64 Cryo1GainLowAmb
    743: float64 Cryo1GainHighAmb
    751: float64 Cryo1TempNominal
    759: float64 Cryo1SlowStartFact
    767: float64 Cryo1SlowStartT1
    775: float64 Cryo1SlowStartT2
    783: float64 Cryo1SlowStartT3
    791: uint32 Cryo1VTempSensor1
    795: uint32 Cryo1VTempSensor2
    799: uint32 Cryo1Cmd1
    803: uint32 Cryo1Cmd2
    807: uint32 Cryo1Cmd3
    811: uint32 Cryo1Cmd4
    815: uint32 Cryo1CmdResponse1
    819: uint32 Cryo1CmdResponse2
    823: uint32 Cryo1CmdResponse3
    827: uint32 Cryo1CmdResponse4
    831: uint32 Cryo2PowerUpCycles
    835: float64 Cryo2TempAmb
    843: float64 Cryo2VSensor
    851: float64 Cryo2VAC
    859: float64 Cryo2VDC
    867: float64 Cryo2CurrentDC
    875: float64 Cryo2SetPoint
    883: float64 Cryo2PGain
    891: float64 Cryo2IGain
    899: uint32 Cryo2ReadyWindow
    903: uint32 Cryo2RemoteReady
    907: float64 Cryo2OutputVNomTemp
    915: float64 Cryo2FreqOfACOutput
    923: float64 Cryo2GainLowAmb
    931: float64 Cryo2GainHighAmb
    939: float64 Cryo2TempNominal
    947: float64 Cryo2SlowStartFact
    955: float64 Cryo2SlowStartT1
    963: float64 Cryo2SlowStartT2
    971: float64 Cryo2SlowStartT3
    979: uint32 Cryo2VTempSensor1
    983: uint32 Cryo2VTempSensor2
    987: uint32 Cryo2Cmd1
    991: uint32 Cryo2Cmd2
    995: uint32 Cryo2Cmd3
    999: uint32 Cryo2Cmd4
    1003: uint32 Cryo2CmdResponse1
    1007: uint32 Cryo2CmdResponse2
    1011: uint32 Cryo2CmdResponse3
    1015: uint32 Cryo2CmdResponse4
    1019: uint32 Cryo3PowerUpCycles
    1023: float64 Cryo3TempAmb
    1031: float64 Cryo3VSensor
    1039: float64 Cryo3VAC
    1047: float64 Cryo3VDC
    1055: float64 Cryo3CurrentDC
    1063: float64 Cryo3SetPoint
    1071: float64 Cryo3PGain
    1079: float64 Cryo3IGain
    1087: uint32 Cryo3ReadyWindow
    1091: uint32 Cryo3RemoteReady
    1095: float64 Cryo3OutputVNomTemp
    1103: float64 Cryo3FreqOfACOutput
    1111: float64 Cryo3GainLowAmb
    1119: float64 Cryo3GainHighAmb
    1127: float64 Cryo3TempNominal
    1135: float64 Cryo3SlowStartFact
    1143: float64 Cryo3SlowStartT1
    1151: float64 Cryo3SlowStartT2
    1159: float64 Cryo3SlowStartT3
    1167: uint32 Cryo3VTempSensor1
    1171: uint32 Cryo3VTempSensor2
    1175: uint32 Cryo3Cmd1
    1179: uint32 Cryo3Cmd2
    1183: uint32 Cryo3Cmd3
    1187: uint32 Cryo3Cmd4
    1191: uint32 Cryo3CmdResponse1
    1195: uint32 Cryo3CmdResponse2
    1199: uint32 Cryo3CmdResponse3
    1203: uint32 Cryo3CmdResponse4
    1207: uint32 CryoSideSelectState
    1211: uint32 NVRAMNumOps
    1215: uint32 NVRAMNextOpNum
    1219: uint8 NVRAMNextOp
    1220: time4s4ns EVRHist0Time
    1228: uint16 EVRHist0Code
    1230: uint8 EVRHist0DataLen
    1231: uint8 EVRHist0Details[0], array has 10 elements
    1241: time4s4ns EVRHist1Time
    1249: uint16 EVRHist1Code
    1251: uint8 EVRHist1DataLen
    1252: uint8 EVRHist1Details[0], array has 10 elements
    1262: time4s4ns EVRHist2Time
    1270: uint16 EVRHist2Code
    1272: uint8 EVRHist2DataLen
    1273: uint8 EVRHist2Details[0], array has 10 elements
    1283: time4s4ns EVRHist3Time
    1291: uint16 EVRHist3Code
    1293: uint8 EVRHist3DataLen
    1294: uint8 EVRHist3Details[0], array has 10 elements
    1304: time4s4ns EVRHist4Time
    1312: uint16 EVRHist4Code
    1314: uint8 EVRHist4DataLen
    1315: uint8 EVRHist4Details[0], array has 10 elements
    1325: time4s4ns EVRHist5Time
    1333: uint16 EVRHist5Code
    1335: uint8 EVRHist5DataLen
    1336: uint8 EVRHist5Details[0], array has 10 elements
    1346: time4s4ns EVRHist6Time
    1354: uint16 EVRHist6Code
    1356: uint8 EVRHist6DataLen
    1357: uint8 EVRHist6Details[0], array has 10 elements
    1367: time4s4ns EVRHist7Time
    1375: uint16 EVRHist7Code
    1377: uint8 EVRHist7DataLen
    1378: uint8 EVRHist7Details[0], array has 10 elements
    1388: time4s4ns EVRHist8Time
    1396: uint16 EVRHist8Code
    1398: uint8 EVRHist8DataLen
    1399: uint8 EVRHist8Details[0], array has 10 elements
    1409: time4s4ns EVRHist9Time
    1417: uint16 EVRHist9Code
    1419: uint8 EVRHist9DataLen
    1420: uint8 EVRHist9Details[0], array has 10 elements
    1430: uint32 EVRErrorsCount
    1434: uint32 EVRInfosCount
    1438: uint32 EVRTotalCount
    1442: uint16 EVRNumErrorDumpProducts
    1444: uint32 EthernetV5Version
    1448: uint8 EthernetV5SBECount
    1449: uint8 EthernetV5DBEFlag
    1450: uint32 EthernetAncBytesBuffed
    1454: uint32 EthernetAncBytesSent
    1458: uint32 EthernetFPIESDRAMRawPkts
    1462: uint32 EthernetFPIEUncompressedPkts
    1466: uint32 EthernetFPIECompressedPkts
    1470: uint32 EthernetFPIEBytesSent
    1474: uint32 EthernetDownlinkStart
    1478: uint32 EthernetDownlinkEnd
    1482: float32 EthernetDownlinkRate
    1486: uint32 EthernetNumDownlinksTableProducts
    1490: uint32 EthernetLastReqID
    1494: uint8 EthernetLastReadIndex
    1495: uint32 EthernetLastReadValue0
    1499: uint32 EthernetLastReadValue1
    1503: uint32 EthernetLastReadValue2
    1507: uint32 EthernetLastReadValue3
    1511: int32 EthernetLastMOXARSSI
    1515: uint8 EthernetLastReadReg
    1516: uint32 EthernetLastReadValue
    1520: uint16 X1553ErrorsFormatIn
    1522: uint16 X1553ErrorsLoopTest
    1524: uint16 X1553ErrorsDataRollover
    1526: uint16 X1553ErrorsIllegalCmd
    1528: uint16 X1553ErrorsWordCount
    1530: uint16 X1553ErrorsDataSync
    1532: uint16 X1553ErrorsInvalidWord
    1534: uint16 X1553ErrorsCmdWord
    1536: uint16 X1553ErrorsXmitTimeout
    1538: uint16 X1553ErrorsStkRollover
    1540: uint16 X1553ErrorsHandshake
    1542: uint16 X1553ErrorsAddrParity
    1544: uint16 X1553ErrorsFormatOut
    1546: uint32 FPCRC00
    1550: uint32 FPCRC01
    1554: uint32 FPCRC02
    1558: uint32 FPCRC03
    1562: uint32 FPCRC04
    1566: uint32 FPCRC05
    1570: uint32 FPCRC06
    1574: uint32 FPCRC07
    1578: uint32 FPCRC08
    1582: uint32 FPCRC09
    1586: uint32 FPCRC10
    1590: uint32 FPCRC11
    1594: uint32 FPCRC12
    1598: uint32 FPCRC13
    1602: uint32 FPCRC14
    1606: uint32 FPCRC15
    1610: uint32 FPCRC16
    1614: uint32 FPCRC17
    1618: uint32 FPCRC18
    1622: uint32 FPCRC19
    1626: uint32 FPCRC20
    1630: uint32 FPCRC21
    1634: uint32 FPCRC22
    1638: uint32 FPCRC23
    1642: uint32 FPCRC24
    1646: uint32 FPCRC25
    1650: uint32 FPCRC26
    1654: uint32 FPCRC27
    1658: uint32 FPCRC28
    1662: uint32 FPCRC29
    1666: uint32 FPCRC30
    1670: uint32 FPCRC31
    1674: uint32 FPCRC32
    1678: uint32 FPCRC33
    1682: uint32 FPCRC34
    1686: uint32 FPCRC35
    1690: uint32 FPCRC36
    1694: uint32 FPCRC37
    1698: uint32 FPCRC38
    1702: uint32 FPCRC39
    1706: uint32 FPCRC40
    1710: uint32 FPCRC41
    1714: uint32 FPCRC42
    1718: uint32 FPCRC43
    1722: uint32 FPCRC44
    1726: uint32 FPCRC45
    1730: uint32 FPCRC46
    1734: uint32 FPCRC47
    1738: uint32 FPCRC48
    1742: uint32 FPCRC49
    1746: uint16 FPNumCRCTableProducts
    1748: uint32 FPCurrentCRCRegion
    1752: uint32 FPEnabled000MSB
    1756: uint32 FPEnabled032MSB
    1760: uint32 FPEnabled064MSB
    1764: uint32 FPEnabled096MSB
    1768: uint32 FPEnabled128MSB
    1772: uint32 FPEnabled160MSB
    1776: uint32 FPEnabled192MSB
    1780: uint32 FPEnabled224MSB
    1784: uint32 FPTriggered000MSB
    1788: uint32 FPTriggered032MSB
    1792: uint32 FPTriggered064MSB
    1796: uint32 FPTriggered096MSB
    1800: uint32 FPTriggered128MSB
    1804: uint32 FPTriggered160MSB
    1808: uint32 FPTriggered192MSB
    1812: uint32 FPTriggered224MSB
    1816: time4s4ns FaultHist0Time
    1824: uint16 FaultHist0ID
    1826: uint8 FaultHist0Countdown
    1827: time4s4ns FaultHist1Time
    1835: uint16 FaultHist1ID
    1837: uint8 FaultHist1Countdown
    1838: time4s4ns FaultHist2Time
    1846: uint16 FaultHist2ID
    1848: uint8 FaultHist2Countdown
    1849: time4s4ns FaultHist3Time
    1857: uint16 FaultHist3ID
    1859: uint8 FaultHist3Countdown
    1860: time4s4ns FaultHist4Time
    1868: uint16 FaultHist4ID
    1870: uint8 FaultHist4Countdown
    1871: time4s4ns FaultHist5Time
    1879: uint16 FaultHist5ID
    1881: uint8 FaultHist5Countdown
    1882: time4s4ns FaultHist6Time
    1890: uint16 FaultHist6ID
    1892: uint8 FaultHist6Countdown
    1893: time4s4ns FaultHist7Time
    1901: uint16 FaultHist7ID
    1903: uint8 FaultHist7Countdown
    1904: time4s4ns FaultHist8Time
    1912: uint16 FaultHist8ID
    1914: uint8 FaultHist8Countdown
    1915: time4s4ns FaultHist9Time
    1923: uint16 FaultHist9ID
    1925: uint8 FaultHist9Countdown
    1926: uint16 FPNumFaultResponseTableProducts
    1928: uint8 FPIEV5ProgrammingState
    1929: uint32 FPIEV5Version
    1933: uint8 FPIEV5SBECount
    1934: uint8 FPIEV5DBEFlag
    1935: uint8 FPIELastReadReg
    1936: uint32 FPIELastReadValue
    1940: uint8 MotorV5ProgrammingState
    1941: uint32 MotorV5Version
    1945: uint8 MotorV5SBECount
    1946: uint8 MotorV5DBEFlag
    1947: uint8 MotorLastReadReg
    1948: uint32 MotorLastReadValue
    1952: int32 MotorPos1
    1956: time4s4ns MotorPos1Time
    1964: int32 MotorRate1
    1968: int32 MotorPos2
    1972: time4s4ns MotorPos2Time
    1980: int32 MotorRate2
    1984: int32 MotorPos3
    1988: time4s4ns MotorPos3Time
    1996: int32 MotorRate3
    2000: int32 MotorPos4
    2004: time4s4ns MotorPos4Time
    2012: int32 MotorRate4
    2016: int32 MotorPos5
    2020: time4s4ns MotorPos5Time
    2028: int32 MotorRate5
    2032: uint8 MotorWaitingForMove
    2033: uint32 MotorMode
    2037: uint32 MotorSunSafePosCounts
    2041: uint32 MotorBB1PosCounts
    2045: uint32 MotorBB2PosCounts
    2049: uint32 PCECircuitsOn
    2053: uint8 TempCtl0Mode
    2054: uint8 TempCtl0Sensor
    2055: uint8 TempCtl0Circuit
    2056: uint16 TempCtl0Target
    2058: uint16 TempCtl0Sample
    2060: uint8 TempCtl0HtrOn
    2061: uint8 TempCtl1Mode
    2062: uint8 TempCtl1Sensor
    2063: uint8 TempCtl1Circuit
    2064: uint16 TempCtl1Target
    2066: uint16 TempCtl1Sample
    2068: uint8 TempCtl1HtrOn
    2069: uint8 TempCtl2Mode
    2070: uint8 TempCtl2Sensor
    2071: uint8 TempCtl2Circuit
    2072: uint16 TempCtl2Target
    2074: uint16 TempCtl2Sample
    2076: uint8 TempCtl2HtrOn
    2077: uint8 TempCtl3Mode
    2078: uint8 TempCtl3Sensor
    2079: uint8 TempCtl3Circuit
    2080: uint16 TempCtl3Target
    2082: uint16 TempCtl3Sample
    2084: uint8 TempCtl3HtrOn
    2085: uint32 HAndSCRC
    2089: uint32 TrailerSync
    2093: uint8 Alignment[0], array has 3 elements
    Condition:
        ProductSync == 0x0c03ec05 && ProductType == HEALTH_AND_STATUS &&
        TrailerSync == 0xeeeeeeee
PACKET highRateBADProduct (fixed-size, 760 bytes):
    0: uint32 PacketLength
    4: uint32 ProductSync
    8: time4s4ns BADProductTime
    16: uint8 ProductType
    17: uint8 ProductLength[0], array has 3 elements
    20: uint32 ProductDetails[0], array has 3 elements
    32: uint16 HighRateBADData[0], array has 360 elements
    752: uint32 HighRateBADCRC
    756: uint32 TrailerSync
    Condition:
        ProductSync == 0x0c03ec05 && ProductType == HIGH_RATE_BAD &&
        TrailerSync == 0xeeeeeeee
PACKET lowRateBADProduct (fixed-size, 4040 bytes):
    0: uint32 PacketLength
    4: uint32 ProductSync
    8: time4s4ns BADProductTime
    16: uint8 ProductType
    17: uint8 ProductLength[0], array has 3 elements
    20: uint32 ProductDetails[0], array has 3 elements
    32: uint16 LowRateBADData[0], array has 2000 elements
    4032: uint32 LowRateBADCRC
    4036: uint32 TrailerSync
    Condition:
        ProductSync == 0x0c03ec05 && ProductType == LOW_RATE_BAD &&
        TrailerSync == 0xeeeeeeee
PACKET memoryDumpProduct (variable-size, 65576 bytes max):
    0: uint32 PacketLength
    4: uint32 ProductSync
    8: time4s4ns DumpProductTime
    16: uint8 ProductType
    17: uint8 ProductLength[0], array has 3 elements
    20: uint32 MemoryBank
    24: uint32 StartAddr
    28: uint32 NumBytes
    32: uint8 MemoryDump[0], array has NumBytes elements
    -1: uint32 MemoryDumpCRC
    -1: uint32 TrailerSync
    Condition:
        ProductSync == 0x0c03ec05 && ProductType == MEM_DUMP &&
        TrailerSync == 0xeeeeeeee
PACKET scheduleTableProduct (fixed-size, 6072 bytes):
    0: uint32 PacketLength
    4: uint32 ProductSync
    8: time4s4ns TableProductTime
    16: uint8 ProductType
    17: uint8 ProductLength[0], array has 3 elements
    20: uint32 ProductDetails[0], array has 3 elements
    32: uint16 TableMagic
    34: uint8 UploadType
    35: uint8 TableVersion
    36: uint16 NumEntries
    38: uint16 TableID
    40: uint8 SHA1[0], array has 20 elements
    60: uint32 Pad32
    64: scheduleTableEntry ScheduleTable[0], array has 500 elements
    6064: uint32 ScheduleTableCRC
    6068: uint32 TrailerSync
    Condition:
        ProductSync == 0x0c03ec05 && ProductType == SCHEDULE_TABLE &&
        TrailerSync == 0xeeeeeeee
PACKET logDumpProduct (fixed-size, 428 bytes):
    0: uint32 PacketLength
    4: uint32 ProductSync
    8: time4s4ns LogProductTime
    16: uint8 ProductType
    17: uint8 ProductLength[0], array has 3 elements
    20: uint32 ProductDetails[0], array has 3 elements
    32: mainLogEntry MainLogData[0], array has 16 elements
    160: isrLogEntry ISRLogData[0], array has 16 elements
    288: evrLogEntry EVRLogData[0], array has 10 elements
    408: uint8 AssertFile[0], array has 10 elements
    418: uint16 AssertLineNum
    420: uint32 LogDumpCRC
    424: uint32 TrailerSync
    Condition:
        ProductSync == 0x0c03ec05 && ProductType == LOG_DUMP &&
        TrailerSync == 0xeeeeeeee
PACKET crcTableProduct (fixed-size, 872 bytes):
    0: uint32 PacketLength
    4: uint32 ProductSync
    8: time4s4ns TableProductTime
    16: uint8 ProductType
    17: uint8 ProductLength[0], array has 3 elements
    20: uint32 ProductDetails[0], array has 3 elements
    32: uint16 TableMagic
    34: uint8 UploadType
    35: uint8 TableVersion
    36: uint16 NumEntries
    38: uint16 TableID
    40: uint8 SHA1[0], array has 20 elements
    60: uint32 Pad32
    64: crcTableEntry CRCTable[0], array has 50 elements
    864: uint32 CRCTableCRC
    868: uint32 TrailerSync
    Condition:
        ProductSync == 0x0c03ec05 && ProductType == CRC_TABLE &&
        TrailerSync == 0xeeeeeeee
PACKET faultResponseTableProduct (fixed-size, 8264 bytes):
    0: uint32 PacketLength
    4: uint32 ProductSync
    8: time4s4ns TableProductTime
    16: uint8 ProductType
    17: uint8 ProductLength[0], array has 3 elements
    20: uint32 ProductDetails[0], array has 3 elements
    32: uint16 TableMagic
    34: uint8 UploadType
    35: uint8 TableVersion
    36: uint16 NumEntries
    38: uint16 TableID
    40: uint8 SHA1[0], array has 20 elements
    60: uint32 Pad32
    64: faultResponseEntry FaultResponseTable[0], array has 256 elements
    8256: uint32 FaultResponseTableCRC
    8260: uint32 TrailerSync
    Condition:
        ProductSync == 0x0c03ec05 && ProductType == FAULT_RESPONSE_TABLE &&
        TrailerSync == 0xeeeeeeee
PACKET sequenceProduct (fixed-size, 72 bytes):
    0: uint32 PacketLength
    4: uint32 ProductSync
    8: time4s4ns SeqProductTime
    16: uint8 ProductType
    17: uint8 ProductLength[0], array has 3 elements
    20: uint32 SeqID
    24: uint32 StartingCmd
    28: uint32 NumCmds
    32: uint16 TableMagic
    34: uint8 UploadType
    35: uint8 TableVersion
    36: uint16 NumEntries
    38: uint16 TableID
    40: uint8 SHA1[0], array has 20 elements
    60: uint32 Pad32
    64: uint32 SequenceCRC
    68: uint32 TrailerSync
    Condition:
        ProductSync == 0x0c03ec05 && ProductType == SEQUENCE && NumCmds == 0
&&
        TrailerSync == 0xeeeeeeee
PACKET sequenceProduct (variable-size, 7209000 bytes max):
    0: uint32 PacketLength
    4: uint32 ProductSync
    8: time4s4ns SeqProductTime
    16: uint8 ProductType
    17: uint8 ProductLength[0], array has 3 elements
    20: uint32 SeqID
    24: uint32 StartingCmd
    28: uint32 NumCmds
    32: sequenceEntry Sequence[0], array has NumCmds elements
    -1: uint32 SequenceCRC
    -1: uint32 TrailerSync
    Condition:
        ProductSync == 0x0c03ec05 && ProductType == SEQUENCE &&
        (NumCmds % 2) == 0 && TrailerSync == 0xeeeeeeee
PACKET sequenceProduct (variable-size, 7209002 bytes max):
    0: uint32 PacketLength
    4: uint32 ProductSync
    8: time4s4ns SeqProductTime
    16: uint8 ProductType
    17: uint8 ProductLength[0], array has 3 elements
    20: uint32 SeqID
    24: uint32 StartingCmd
    28: uint32 NumCmds
    32: sequenceEntry Sequence[0], array has NumCmds elements
    -1: uint32 SequenceCRC
    -1: uint32 TrailerSync
    -1: uint16 Pad
    Condition:
        ProductSync == 0x0c03ec05 && ProductType == SEQUENCE &&
        (NumCmds % 2) == 1 && TrailerSync == 0xeeeeeeee
PACKET downlinksTableProduct (fixed-size, 12072 bytes):
    0: uint32 PacketLength
    4: uint32 ProductSync
    8: time4s4ns TableProductTime
    16: uint8 ProductType
    17: uint8 ProductLength[0], array has 3 elements
    20: uint32 ProductDetails[0], array has 3 elements
    32: uint16 TableMagic
    34: uint8 UploadType
    35: uint8 TableVersion
    36: uint16 NumEntries
    38: uint16 TableID
    40: uint8 SHA1[0], array has 20 elements
    60: uint32 Pad32
    64: downlinksEntry DownlinksTable[0], array has 1000 elements
    12064: uint32 DownlinksTableCRC
    12068: uint32 TrailerSync
    Condition:
        ProductSync == 0x0c03ec05 && ProductType == DOWNLINKS_TABLE &&
        TrailerSync == 0xeeeeeeee
PACKET errorDumpProduct (fixed-size, 2440 bytes):
    0: uint32 PacketLength
    4: uint32 ProductSync
    8: time4s4ns ErrorDumpProductTime
    16: uint8 ProductType
    17: uint8 ProductLength[0], array has 3 elements
    20: uint32 ProductDetails[0], array has 3 elements
    32: errorDumpEntry ErrorDump[0], array has 100 elements
    2432: uint32 ErrorDumpCRC
    2436: uint32 TrailerSync
    Condition:
        ProductSync == 0x0c03ec05 && ProductType == ERROR_DUMP &&
        TrailerSync == 0xeeeeeeee
PACKET scienceData (fixed-size, 200708 bytes):
    0: uint32 SciDataLength
    4: uint32 SyncWord
    8: uint32 PacketID
    12: uint32 LineIDs[0], array has 64 elements
    268: uint32 InstrState
    272: time4s4ns FSWTimestamp
    280: uint8 FPIETimestamp[0], array has 8 elements
    288: uint8 FPIEAtFSWCorr[0], array has 8 elements
    296: uint32 PktLen
    300: uint8 FPIEData[0], array has 200408 elements
    Condition:
        SciDataLength == 0x31000 && SyncWord == 0x00ff00ff &&
        PktLen == 0x3c020300
```

APPENDIX B: ABBREVIATIONS AND ACRONYMS

ALEXI	Atmospheric-Land Exchange Inversion
ARS	Agricultural Research Service
ASD	Algorithm Specifications Document
ATBD	Algorithm Theoretical Basis Document
CCB	Change Control Board
CDR	Critical Design Review
CF	Climate and Forecast (metadata convention)
CM	Configuration Management
CONUS	Continental United States
COTS	Commercial Off The Shelf
DAAC	Distributed Active Archive Center
dB	DeciBel
DCN	Document Change Notice
deg	Degrees
deg/sec	Degrees per Second
DEM	Digital Elevation Model
DisALEXI	ALEXI Disaggregation algorithm
DN	Data Number
EASE	Equal Area Scalable Earth
ECI	Earth Centered Inertial coordinate system
ECR	Earth Centered Rotating coordinate system
ECS	EOSDIS Core System
ECOSTRESS	ECOsystem Spaceborne Thermal Radiometer on Space Station
EOS	Earth Observing System
EOSDIS	EOS Data and Information System
ESDIS	Earth Science Data and Information System
ESDT	Earth Science Data Type
FOV	Field of View
FSW	Flight Software
GB	gigabytes, 10^9 bytes
GDS	Ground Data System
GHA	Greenwich Hour Angle
GHz	Gigahertz, 10^9 hertz
GMAO	Global Modeling and Assimilation Office
GMT	Greenwich Mean Time
GPP	Gross Primary Production
GSE	Ground Support Equipment
GSFC	Goddard Space Flight Center
HDF	Hierarchical Data Format
HK	Housekeeping (telemetry)
HRSL	Hydrology and Remote Sensing Laboratory
Hz	Hertz
HSD	Health and Status Data
I&T	Integration and Test
ICD	Interface Control Document
I/O	Input/Output

IOC	In-Orbit Checkout
IPA	Inter-Project Agreement
ITAR	International Traffic in Arms Regulation
JPL	Jet Propulsion Laboratory
K	Kelvin
KHz	Kilohertz
Km	kilometer, 1000 meters
L0 – L4	Level 0 through Level 4
LAN	Local Area Network
LEO	Low Earth Orbit
LOE	Level of Effort
LOM	Life of Mission
LP	Land Processes
LSTE	Land Surface Temperature and Emissivity
m	meter
MB	megabytes, 10^6 bytes
Mbps	Mega bits per second
MHz	Megahertz
MMR	Monthly Management Review
MOA	Memorandum of Agreement
MODIS	Moderate Resolution Imaging Spectroradiometer
MOS	Mission Operations System
m/s	meters per second
ms	milliseconds
MS	Mission System
NASA	National Aeronautics and Space Administration
NCEP	National Centers for Environmental Protection
NCSA	National Center for Supercomputing Applications
netCDF	Network Common Data Format
NISN	NASA Integrated Services Network
NOAA	National Oceanic and Atmospheric Administration
OA	Operations Agreement
ODL	Object Description Language
OODT	Object Oriented Data Technology
ORR	Operational Readiness Review
ORT	Operational Readiness Test
PDR	Preliminary Design Review
percent	%, per hundred
PR	Problem Report
PSD	Product Specifications Document
PT-JPL	Priestly-Taylor-JPL
QA	Quality Assurance
rad	radians
RDD	Release Description Document
RFA	Request For Action
S/C	Spacecraft
SCP	Secure Copy
SDP	Software Development Plan
SDS	Science Data System

sec, s	seconds
SITP	System Integration and Test Plan
SMP	Software Management Plan
SOM	Software Operators Manual
TAI	International Atomic Clock
T_b	Brightness Temperature
TBD	To Be Determined
TBS	To Be Specified
TOA	Time of Arrival
TPS	Third Party Software
USDA	United State Department of Agriculture
USGS	United States Geological Society
UTC	Coordinated Universal Time
V&V	Verification and Validation
XML	Extensible Markup Language

APPENDIX C: INTERMEDIATE AND NON-DELIVERED PRODUCTS

3.5 Intermediate, Temporary, and other non-delivered products

The following groups define formats of various intermediate and temporary products generated and used by L0 PGEs, but are not standard products that are delivered to the DAAC.

3.5.1 L0B – Sorted L0 packets grouped by orbits

Table 4-1: L0B Data Definitions

Field Name	Type	Units	Field Data
Group	**StandardMetadata** (as defined above)		
Group	**flex** (Size Unlimited x 196896 bytes per packet)		
bip	UInt16	DN	Band interleaved pixels (64 focal planes by 256 lines per FP by 6 bands, 196608 bytes per packet)
Id_line	Uint32		Focal plane encoder values (64 FP, 256 bytes per packet)
Id_packet	Uint32		Numeric ID of packet
state	Uint32		Instrument state (see telemetry dictionary for all possible states)
time_fsw	Float64	GPS Sec	Flight software time when packet was recorded, in GPS time unit
time_sync_fpie	Uint64	μsec	FPIE SYNC clock of first FP in packet
time_sync_fsw	Uint64	μsec	FPIE SYNC clock when FSW_TIME was recorded into packet
Group	**HK/bad/hr** (Size Unlimited x 56 bytes per packet)		
attitude	Float32	Quaternion	4 element quaternion defining spacecraft attitude (4 floating point values, 16 bytes per packet)
position	Float32	Meters	X,Y,Z position of spacecraft in meters (3 floating point values, 12 bytes per packet)
time_fsw	Float64	GPS sec	Flight software time when packet was recorded, in GPS time unit
time	Float64	GPS sec	Flight software time when sample was recorded, in GPS time unit
velocity	Float32	m/s	X,Y,Z velocity of spacecraft (3 floating point values, 12 bytes per packet)
Group	**HK/status** (Size Unlimited x 36 bytes per packet)		
temperature	Uint16	DN	D/A values of PRT sensors on black bodies (2 BB by 5 sensors each)
time	Float64	GPS sec	FSWT when PRT were recorded
time_fsw	Float64	GPS sec	BADProductTime
Group	**HK/status/mode** (Size Unlimited x 5 bytes per packet)		

dpuio	Uint8	None	CoreDPUIOMode: DPUIO_IDLE = 0x00,
			RECEIVE_FPIE_DATA = 0x01,
			STAGE_WITH_COMPRESS = 0x10,
			STAGE_WITHOUT_COMPRESS = 0x20,
			TRANSMIT = 0x80

op	Uint32		CoreCurrentOpMode: DECONTAMINATION = 0, SAFEHOLD, IDLE, SCANNING, SCIENCE
Group	**HK/status/motor** (Size Unlimited x 81 bytes per packet)		

bb1	Uint32		AnalogsTemp_BB_*_1
bb2	Uint32		AnalogsTemp_BB_*_2
mode	Uint32		MotorMode: STOPPED = 0, SPINNING = 1, BLACKBODY_1 = 2, BLACKBODY_2 = 3, SUNSAFE = 4, MANUAL = 5
position	Uint32		MotorPos: Last 5 position values in encoder counts, 5 sec int.
pstate	Uint32		MotorV5ProgrammingState: NOT_PROGRAMMED, READING_NOR_FLASH, STARTING_PROGRAMMING, WAITING_FOR_COMPLETION, DONE
rate	Uint32		MotorRate: Last 5 motor velocity values, EV/sec
sun_safe	Uint32		MotorSunSafePosCounts EV
time	Uint32		AnalogsSampleTime - Last 5 times at which analogs were sampled
wait	Uint8		MotorWaitingForMove - moving toward position
Group	**HK/status/motor/last** (Size Unlimited x 5 bytes per packet)		
register	Uint8		Last register read by command
value	Uint32		Last value read