Department of the Interior
U.S. Geological Survey

LANDSAT 5 (L5)
THEMATIC MAPPER (TM)
CALIBRATION PARAMETER FILE (CPF) DEFINITION

Version 4.0

January 2008

USGS
science for a changing world

LANDSAT 5 (L5) THEMATIC MAPPER (TM) CALIBRATION PARAMETER FILE (CPF) DEFINITION

January 2008

Prepared By:

____

E. Micijevic   Date
Calibration Analyst
SAIC

Reviewed By:

____

D. Strande   Date
IAS Systems Engineer
SAIC

Reviewed By:

____

C. Engebretson   Date
IAS Software Project Lead
SAIC

Approved By:

____

R. Hayes   Date
CalVal Task Lead
SAIC

Approved By:

____

J. Lacasse   Date
Mission Management Officer
U.S. Geological Survey

EROS
Sioux Falls, South Dakota

Executive Summary

This document describes the contents of the Calibration Parameter File (CPF) generated by the Thematic Mapper (TM) functionality of the Image Assessment System (IAS). The IAS routinely performs radiometric and geometric calibration and updates the CPF. This file is stamped with an applicability date range and is sent to the Landsat Archive Manager (LAM) for storage and eventual bundling with outbound Level 0 Reformatted Products (L0Rp). The CPF supplies the radiometric and geometric correction parameters required during Level 1 (L1) processing to create superior products of uniform consistency across the Landsat system.

Document History

Document Number	Document Version	Publication Date	Change Number
IAS-226	Version 1	August 9, 2004	NA
IAS-226	Version 2.0	October 28, 2005	CCR 4094
IAS-226	Version 3.0	September 2007	CCR 4254
IAS-226	Version 4.0	January 2008	NA

Contents

List of Tables

Section 1 Introduction

1.1 Background

In May 2003, the Landsat Project at the U.S. Geological Survey (USGS) Center for Earth Resources Observation and Science (EROS) and the National Aeronautics and Space Administration (NASA) Goddard Space Flight Center (GSFC) developed a joint charter to assess Landsat 4 (L4) and Landsat 5 (L5) data in an effort to enhance the radiometric and geometric accuracy of the image data products.

L4 and L5 comprise the suite of satellites with an on-board Thematic Mapper (TM) sensor. L5 has the only TM sensor in use at the time of this document's creation. Before TM functionality was incorporated into the Image Assessment System (IAS), any calibration of the TM instrument was reactive in nature. For the most part, when a problem was identified in the Level 0 (L0) and Level 1 (L1) production systems, system developers responded by fixing the anomalies as they were found.

The TM calibration effort will take a proactive approach to monitoring the TM data as acquired, validating the integrity of the archived products, and helping to identify and troubleshoot anomalies. Data quality will be assured via radiometric, geometric, and Modulation Transfer Function (MTF) characterization and calibration. Each area will generate inputs to the CPF that will be made available for Level 0 and L1 production systems.

1.2 Purpose and Scope

This document describes the contents of the L5 Calibration Parameter File (CPF) generated by the IAS. The L5 functionality of the IAS is responsible for offline assessment of L5 image quality. In addition to its assessment functions, the IAS is responsible for the radiometric and geometric calibration of L5 TM data. The IAS periodically performs radiometric and geometric calibration and updates the CPF. This file is stamped with an applicability date and is sent to the Landsat Archive Manager (LAM) for storage and eventual bundling with outbound Level 0 Reformatted Products (L0Rp). The CPF supplies the radiometric and geometric correction parameters required during L1 processing to create superior products of uniform consistency across the Landsat system.

Section 2 File Structure

All parameters are stored as American Standard Code for Information Interchange (ASCII) text using the Object Description Language (ODL) syntax developed by the NASA Jet Propulsion Laboratory (JPL). ODL is a tagged keyword language developed to provide a human-readable data structure to encode data for simplified interchange. The ODL interpreter developed by JPL may provide, in certain cases, for the handling of lexical elements (e.g., building blocks) included in the Consultative Committee for Space Data Systems (CCSDS) specification of the Parameter Value Language (PVL), which is a superset of ODL. The IAS CPF is a pure ODL implementation without any PVL extensions.

The body of the file is composed of two statement types:

1. Attribute assignment statement: Used to assign values to parameters
2. Group statements: Used to aid in file organization and enhance parsing granularity of parameter sets

The Planetary Data System Standards Reference provides ODL details (see References).

2.1 Calibration Parameter File Updates

The IAS regularly releases and distributes CPFs at the beginning of each calendar quarter. In addition to a new CPF for the coming calendar quarter, delivery also includes new versions of all CPFs for times affected by the most recent calibration update. Only the most recent CPFs should be used in TM data processing.

Prior to switching to bumper operational mode, CPFs needed to be released on a regular quarterly basis primarily because of the Universal Time Code (UTC) corrected (UT1) time corrections and pole wander predictions included in the file. However, the CPFs could be updated at any given time, if needed, and released for times shorter than a calendar quarter. For example, the first CPF interval covers March 1984 only, due to the satellite’s launch on March 1, 1984.

Since the switch to bumper operational mode in March 2002, multiple version updates can be expected during any given quarter due to the unpredictive nature of the scanning mirror bumper parameters. The irregular (mid-quarter) updates do not affect the three-month CPF release schedule.

2.1.1 Effective Dates

Each CPF is time-stamped with an effective date range. The third and fourth parameters in the file—Effective_Date_Begin and Effective_Date_End—designate the range of valid acquisition dates and are in yyyy-mm-dd format. After the Effective_Date_End, the file is without applicable UT1 time predictions. EROS maintains a database of CPF names and their effective dates for associating product orders with the appropriate parameter files. The parameter file that accompanies an order has an effective date range that includes the acquisition date of the ordered image.

2.2 File-Naming Conventions

Throughout the mission, a serial collection of CPFs will be generated and sent to the LAM for distribution with L0Rp products. The probability exists that a CPF will be replaced due to improved calibration parameters for a given period or perhaps a file error. The need for unique file version numbers becomes necessary as file contents change. Table 2-1 shows the components comprising the naming convention that the IAS uses for CPF files.

CPF file name: L5CPF$y_1y_1y_1y_1m_1m_1d_1d_1$_$y_2y_2y_2y_2m_2m_2d_2d_2$.nn

where	L5	=	constant for Landsat 5
	CPF	=	three-letter CPF designator
	$y_1y_1y_1y_1$	=	four-digit effectivity starting year
	m_1m_1	=	two-digit effectivity starting month
	d_1d_1	=	two-digit effectivity starting day
	_	=	effectivity starting/ending date separator
	$y_2y_2y_2y_2$	=	four-digit effectivity ending year
	m_2m_2	=	two-digit effectivity ending month
	d_2d_2	=	two-digit effectivity ending day
	.	=	ending day/sequence number separator
	nn	=	sequence number for this file (starts with 01)

Table 2-1. File Naming Procedure for the CPF

For example, if the IAS created four CPFs at three-month intervals, and then updated the first file twice and the second and third files once, the assigned file names would be as follows:

File 1 L5CPF19840301_19840331.01
L5CPF19840301_19840331.02
L5CPF19840301_19840331.03
File 2 L5CPF19840401_19840630.01
L5CPF19840401_19840630.02

File 3	L5CPF19840701_19840930.01
	L5CPF19840701_19840930.02
File 4	L5CPF19841001_19841231.01

This example assumes that the effective date ranges did not change. The effective date range for a file can change, however, if a specific problem (e.g., detector outage) is discovered somewhere within the nominal effective range. Assuming this scenario, two CPFs with new names and effective date ranges are spawned for the period under consideration. The Effective_Date_End for a new pre-problem CPF would change to the day before the problem occurred, and the Effective_Date_Begin would remain unchanged. A post-problem CPF with a new file name would be created with an Effective_Date_Begin corresponding to the imaging date when the problem occurred, and the assigned Effective_Date_End would be the original Effective_Date_End for the period under consideration. Both new CPFs, although they appear for the first time for a given effective date, would have a version number one higher than the CPF for the quarter in which they originated. New versions of all other CPFs affected by the updated parameter would also be created.

2.3 File Content Description

Table 3-1 lists all CPF parameters. Within this table, each parameter entry is characterized by five attributes:

1. Parameter group: This attribute identifies a related set of parameters.
2. Parameter name: This attribute uniquely identifies and describes the content of each parameter.
3. Value type: This attribute describes the parameter as either static or dynamic. A static value generally remains unchanged over the life of the mission. A dynamic value changes, or has the potential to change, over the life of the mission. Significant changes to dynamic values trigger a CPF update.
4. Data type: This attribute is referred to using a Hierarchical Data Format (HDF) number type nomenclature, type#, where type is given by the descriptors ‘char’ (character), ‘int’ (integer), or ‘float’ (floating point), and # is a decimal count of the number of bits used to represent the data type. The type mnemonics int and char may be preceded by the letter u, indicating an unsigned value. For example, the data type uint32 refers to an unsigned 32-bit integer value. **Error! Reference source not found.** describes the data types relevant to the CPF.

Data Type	HDF Nomenclature
8-bit character	char8
8-bit unsigned integer	uint8
16-bit signed integer	int16
32-bit signed integer	int32
32-bit floating point number	float32
64-bit floating point number	float64

Table 2-2. Data Types Relevant to the CPF

5. Description: This attribute briefly describes the parameter, its format, and its nominal or expected value(s). The valid parameter format for numeric data is described using letters S, N, and E. S stands for the sign and can assume values + or –; if no sign is specified, the + sign is assumed. N stands for any digit between 0 and 9. The letter E is used in scientific (exponential) notation to represent the “multiplication by 10 raised to the power” specified by the value following the letter E. For example, the valid format SNNN.NNNNESNN can assume any positive or negative value with a significant ranging from 0.0000 to 999.9999 multiplied by 10 raised to the power of any whole number between –99 and 99.

Section 3 CPF Definition

Table 3-1 lists the Landsat 5 TM CPF parameters.

Parameter Groups	Parameter Name	Value Type	Data Type	Description
FILE_ATTRIBUTES	Spacecraft_Name	Static	char8	Descriptor used to identify the spacecraft for which the calibration parameters are applicable Valid format: Landsat_5
FILE_ATTRIBUTES	Sensor_Name	Static	char8	Descriptor used to identify the sensor for which the calibration parameters are applicable Valid format: Thematic_Mapper
FILE_ATTRIBUTES	Effective_Date_Begin	Dynamic	char8	Effective start date for this file Valid format: yyyy-mm-dd, where yyyy = 1984-2050, mm = 01-12, and dd = 01-31
FILE_ATTRIBUTES	Effective_Date_End	Dynamic	char8	Effective end date for this file Valid format: yyyy-mm-dd, where yyyy = 1984-2050, mm = 01-12, and dd = 01-31
FILE_ATTRIBUTES	CPF_File_Name	Dynamic	char8	Original file name assigned by IAS Valid format: L5CPFyyyymmdd_yyyymmdd.nn, where yyyymmdd = effective start date and effective end date, respectively, and nn = incrementing version for within a quarter (01-99)
EARTH_CONSTANTS	Ellipsoid_Name	Static	char8	Name of the ellipsoid used to represent the semi-major and semi-minor axes of the Earth Valid format: TTTTT, where TTTTT = WGS84
EARTH_CONSTANTS	Semi_Major_Axis	Static	float64	Earth semi-major axis; distance in meters from the center of the Earth to the equator Valid format: NNNNNNN.NNNN, where NNNNNNN.NNNN = 6378137.0000
EARTH_CONSTANTS	Semi_Minor_Axis	Static	float64	Earth semi-minor axis; distance in meters from the center of the Earth to the poles Valid format: NNNNNNN.NNNN, where NNNNNNN.NNNN = 6356752.3142
EARTH_CONSTANTS	Ellipticity	Static	float64	Ratio describing polar flattening or the Earth's deviation from an exact sphere (WGS84 standard) Valid format: N.NNNNNNNNNNNNNN, where N.NNNNNNNNNNNNNN = 0.00335281066474 = 1/298.257223563
EARTH_CONSTANTS	Eccentricity	Static	float64	Number describing the Earth ellipsoid eccentricity squared (WGS84 standard) Valid format: N.NNNNNNNNNNNNNN, where N.NNNNNNNNNNNNNN = 0.00669437999013
EARTH_CONSTANTS	Earth_Spin_Rate	Static	float64	Earth's diurnal spin rate in radians per second Valid format: NN.NNNNNNNNNESNN, where NN.NNNNNNNNNESNN = 72.921158553E-06
EARTH_CONSTANTS	Gravity_Constant	Static	float64	Universal gravitational constant x mass of Earth This parameter is given in units of meters cubed per second squared (m^3/s^2) Valid format: N.NNNNNNENN, where N.NNNNNNENN = 3.986005E14
EARTH_CONSTANTS	J2_Earth_Model_ Term	Static	float64	Term that describes Earth's spherical harmonic Valid format: NNNN.NNESNN, where NNNN.NNESNN = 1082.63E-06
ORBIT_PARAMETERS	WRS_Cycle_Days	Static	uint8	Time period, in days, required for the satellite to view the Earth once Valid format: NN, where NN = 16

Parameter Groups	**Parameter Name**	**Value Type**	**Data Type**	**Description**
ORBIT_PARAMETERS	WRS_Cycle_Orbits	Static	uint8	Number of orbits or paths in a complete World Reference System (WRS) cycle Valid format: NNN, where NNN = 233
ORBIT_PARAMETERS	Scenes_Per_Orbit	Static	uint8	Number of scenes or row locations per orbit Valid format: NNN, where NNN = 248
ORBIT_PARAMETERS	Orbital_Period	Static	float64	Time required, in seconds, to complete one orbit Valid format: NNNN.NNNN, where NNNN.NNNN = 5933.0472
ORBIT_PARAMETERS	Angular_Momentum	Static	float64	Angular momentum in orbit, specified in meters squared per second (m^2/s) Valid format: NN.NNNNNNEN, where NN.NNNNNNEN = 53.136250E9
ORBIT_PARAMETERS	Orbit_Radius	Static	float64	Nominal distance in kilometers (km) from the Earth’s center to the spacecraft track Valid format: NNNN.NNNN, where NNNN.NNNN = 7083.4457
ORBIT_PARAMETERS	Orbit_Semimajor_Axis	Static	float64	Nominal semi-major axis in km of the satellite's orbit Valid format: NNNN.NNNN, where NNNN.NNNN = 7083.4457
ORBIT_PARAMETERS	Orbit_Semiminor_Axis	Static	float64	Nominal semi-minor axis in km of the satellite's orbit Valid format: NNNN.NNNN, where NNNN.NNNN = 7083.4408
ORBIT_PARAMETERS	Orbit_Eccentricity	Static	float64	Nominal eccentricity of the satellite's orbit Valid format: N.NNNNNNNN, where N.NNNNNNNN = 0.00117604
ORBIT_PARAMETERS	Inclination_Angle	Static	float64	Angle in degrees formed by the Earth’s equatorial and satellite plane Valid format: NN.NNNN, where NN.NNNN = 98.2096
ORBIT_PARAMETERS	Argument_Of_Perigee	Static	float32	Nominal angle in degrees of point nearest the Earth in orbit as measured from ascending node in the direction of satellite motion Valid format: NN.N, where NN.N = 90.0
ORBIT_PARAMETERS	Descending_Node_Row	Static	uint8	Row corresponding to the Earth's equator Valid format: NN, where NN = 60
ORBIT_PARAMETERS	Long_Path1_Row60	Static	float32	Longitude in degrees west of the point at which path 1 crossed the equator (row 60) Valid format: SNN.N, where SNN.N = –64.6
ORBIT_PARAMETERS	Descending_Node_Time_Min	Static	char8	Minimum local solar time of descending node in AM hours and minutes Valid format: HH:MM, where HH:MM = 09:10
ORBIT_PARAMETERS	Descending_Node_Time_Max	Static	char8	Maximum local solar time of descending node in AM hours and minutes Valid format: HH:MM, where HH:MM = 10:15
ORBIT_PARAMETERS	Nodal_Regression_Rate	Static	float64	Rate in degrees per day that the orbital plane rotates with respect to the Earth Valid format: N.NNNNNNNNN, where N.NNNNNNNNN = 0.985647366
SCANNER_PARAMETERS	Lines_Per_Scan_30	Static	uint8	Detectors per scan for Bands 1-5 and 7 Valid format: NN, where NN = 16
SCANNER_PARAMETERS	Lines_Per_Scan_120	Static	uint8	Detectors per scan for Band 6 Valid format: N, where N = 4
SCANNER_PARAMETERS	Scans_Per_Scene	Static	int16	Scans per nominal WRS scene Valid format: NNN, where NNN = 374
SCANNER_PARAMETERS	Swath_Angle	Dynamic	float32	Object space angle in radians of scan mirror travel during active scan time Valid format: N.NNNNN, where N.NNNNN = 0.26861

Parameter Groups	Parameter Name	Value Type	Data Type	Description
SCANNER_ PARAMETERS	Scan_Rate	Static	float32	Angular scan velocity in radians per second of the scan mirror Valid format: N.NNNNN, where N.NNNNN = 2.21095
SCANNER_ PARAMETERS	Dwell_Time_30	Static	float64	Detector sample time in microseconds for Bands 1-5 and 7 Valid format: N.NNNNNNN, where N.NNNNNNN = 9.6106302
SCANNER_ PARAMETERS	Dwell_Time_120	Static	float64	Detector sample time in microseconds for Band 6 Valid format: N.NNNNNNN, where N.NNNNNNN = 38.4425208
SCANNER_ PARAMETERS	IC_Line_Length_30	Static	int16	Nominal number of detector samples for the internal calibrator for Bands 1-5 and 7 Valid format: NNNN, where NNNN = 1100
SCANNER_ PARAMETERS	IC_Line_Length_120	Static	int16	Nominal number of detector samples for the internal calibrator for Band 6 Valid format: NNN, where NNN = 275
SCANNER_ PARAMETERS	Scan_Line_Length_30	Static	int16	Nominal number of detector samples during active scan time for Bands 1-5 and 7 Valid format: NNNN, where NNNN = 6320
SCANNER_ PARAMETERS	Scan_Line_Length_120	Static	int16	Nominal number of detector samples during active scan time for Band 6 Valid format: NNNN, where NNNN = 1580
SCANNER_ PARAMETERS	Filter_Frequency_30	Static	float32	Bandwidth in kilohertz (kHz) of detector presample filter (defined by 3-dB roll-off point) for Bands 1-5 and 7 Valid format: NN.NN, where NN.NN = 52.02
SCANNER_ PARAMETERS	Filter_Frequency_120	Static	float32	Bandwidth in kHz of detector presample filter (defined by 3-dB roll-off point) for Band 6 Valid format: NN.NNN, where NN.NNN = 13.005
SCANNER_ PARAMETERS	IFOV_B1234	Static	float32	Angle in µrad subtended by a detector in Bands 1, 2, 3, and 4 when the scanning motion is stopped Valid format: NN.N, where NN.N = 42.5
SCANNER_ PARAMETERS	IFOV_B57_along_ scan	Static	float32	Along-scan angle in µrad subtended by a detector in Bands 5 and 7 when the scanning motion is stopped Valid format: NN.N, where NN.N = 42.5
SCANNER_ PARAMETERS	IFOV_B57_across_ scan	Static	float32	Across-scan angle in µrad subtended by a detector in Bands 5 and 7 when the scanning motion is stopped Valid format: NN.N, where NN.N = 42.5
SCANNER_ PARAMETERS	IFOV_B6	Static	float32	Angle in µrad subtended by a Band 6 detector when the scanning motion is stopped Valid format: NN.N, where NNN.N = 170.0
SCANNER_ PARAMETERS	Scan_Period	Static	float64	Time in milliseconds of a complete scan cycle, including forward and reverse scans Valid format: NNN.NNNNNN, where NNN.NNNNNN = 142.922000
SCANNER_ PARAMETERS	Scan_Frequency	Static	float32	Number of scans in 1 second (hertz [Hz]) Valid format: N.NNNN, where N.NNNN = 6.9968
SCANNER_ PARAMETERS	Active_Scan_Time	Static	float32	Time in µs required for scan mirror to travel from its scan-line-start to end-of-line (EOL) Valid format: NNNNN.NNN, where NNNNN.NNN = 60743.013
SCANNER_ PARAMETERS	Turn_Around_Time	Static	float32	Time in milliseconds from EOL to the next scan-line-start, during which the scan mirror motion reverses direction Valid format: NN.NNN, where NN.NNN = 10.719

Parameter Groups	Parameter Name	Value Type	Data Type	Description
SPACECRAFT_PARAMETERS	ADS_Interval	Static	float32	Time in milliseconds between Angular Displacement Sensor (ADS) samples Valid format: N.N, where N.N = 2.0
SPACECRAFT_PARAMETERS	ADS_Roll_Offset	Static	float32	Amount of time in milliseconds from the start of a Payload Correction Data (PCD) cycle to the roll axis measurement Valid format: N.NNN, where N.NNN = 0.375
SPACECRAFT_PARAMETERS	ADS_Pitch_Offset	Static	float32	Amount of time in milliseconds from the start of a PCD cycle to the pitch axis measurement Valid format: N.NNN, where N.NNN = 0.875
SPACECRAFT_PARAMETERS	ADS_Yaw_Offset	Static	float32	Amount of time in milliseconds from the start of a PCD cycle to the yaw axis measurement Valid format: N.NNN, where N.NNN = 1.375
SPACECRAFT_PARAMETERS	Data_Rate	Static	float32	TM output bit rate in megabits per second (Mbps) Valid format: NN.NNN, where NN.NNN = 84.903
GROUP: MIRROR_PARAMETERS	Error_Conversion_Factor	Static	float32	First half and second half scan mirror error measurement units in microseconds Valid format: N.NNNNNNNN, where N.NNNNNNNN = 0.18845000 (5.306437 megahertz [MHz])
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_SAM	Forward_Along_SME1_SAM	Dynamic	float64 array (6 values)	Fifth-order polynomial coefficients that describe the departure from linearity of forward along-scan mirror motion; Scan Angle Monitor (SAM) mode with Scan Mirror Electronics (SME) number 1 Valid format for each term: SN.NNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_SAM	Forward_Cross_SME1_SAM	Dynamic	float64 array (6 values)	Fifth-order polynomial coefficients that describe the deviation of forward cross-scan mirror motion from linear; SAM mode with SME number 1 Valid format for each term: SN.NNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_SAM	Forward_Angle1_SME1_SAM	Dynamic	float32	Angle in µrad from the start of the scan to the mid-scan point in the forward direction; SAM mode with SME number 1 Valid format NNNNN.N, where NNNNN.N = 67171.0
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_SAM	Forward_Angle2_SME1_SAM	Dynamic	float32	Angle in µrad from the mid-scan point to the end of the scan in the forward direction; SAM mode with SME number 1 Valid format NNNNN.N, where NNNNN.N = 67159.0
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_SAM	Reverse_Along_SME1_SAM	Dynamic	float64 array (6 values)	Fifth-order polynomial coefficients that describe the deviation of reverse along-scan mirror motion from linear; SAM mode with SME number 1 Valid format for each term: SN.NNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_SAM	Reverse_Cross_SME1_SAM	Dynamic	float64 array (6 values)	Fifth-order polynomial coefficients that describe the deviation of reverse cross-scan mirror motion from linear; SAM mode with SME number 1 Valid format for each term: SN.NNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_SAM	Reverse_Angle1_SME1_SAM	Dynamic	float32	Angle in µrad from the start of the scan to the mid-scan point in the reverse direction; SAM mode with SME number 1 Valid format NNNNN.N, where NNNNN.N = 67159.0

Parameter Groups	**Parameter Name**	**Value Type**	**Data Type**	**Description**
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_SAM	Reverse_Angle2_ SME1_SAM	Dynamic	float32	Angle in µrad from the mid-scan point to the end of the scan in the reverse direction; SAM mode with SME number 1 Valid format NNNNN.N, where NNNNN.N = 67171.0
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_SAM	Forward_Along_ SME2_SAM	Dynamic	float64 array (6 values)	Fifth-order polynomial coefficients that describe the deviation of forward along-scan mirror motion from linear; SAM mode with SME number 2 Valid format for each term: SN.NNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_SAM	Forward_Cross_ SME2_SAM	Dynamic	float64 array (6 values)	Fifth-order polynomial coefficients that describe the deviation of forward cross-scan mirror motion from linear; SAM mode with SME number 2 Valid format for each term: SN.NNNNENNSNN, where S = + or –, N = 0 to 9, and E = E
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_SAM	Forward_Angle1_ SME2_SAM	Dynamic	float32	Angle in µrad from the start of the scan to the mid-scan point in the forward direction; SAM mode with SME number 2 Valid format: NNNNN.N, where NNNNN.N = 67182.0
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_SAM	Forward_Angle2_ SME2_SAM	Dynamic	float32	Angle in µrad from the mid-scan point to the end of the scan in the forward direction; SAM mode with SME number 2 Valid format: NNNNN.N, where NNNNN.N = 67160.0
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_SAM	Reverse_Along_ SME2_SAM	Dynamic	float64 array (6 values)	Fifth-order polynomial coefficients that describe the deviation of the reverse along-scan mirror motion from linear; SAM mode with SME number 2 Valid format for each term: SN.NNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_SAM	Reverse_Cross_ SME2_SAM	Dynamic	float64 array (6 values)	Fifth-order polynomial coefficients that describe the deviation of the reverse cross-scan mirror motion from linear; SAM mode with SME number 2 Valid format for each term: SN.NNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_SAM	Reverse_Angle1_ SME2_SAM	Dynamic	float32	Angle in µrad from the start of the scan to the mid-scan point in the reverse direction; SAM mode with SME number 2 Valid format: NNNNN.N, where NNNNN.N = 67160.0
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_SAM	Reverse_Angle2_ SME2_SAM	Dynamic	float32	Angle in µrad from the mid-scan point to the end of the scan in the reverse direction; SAM mode with SME number 2 Valid format: NNNNN.N, where NNNNN.N = 67182.0
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_BUMP	Forward_Along_ SME1_Bump	Dynamic	float64 array (6 values)	Fifth-order polynomial coefficients that describe the deviation of forward along-scan mirror motion from linear; bumper mode with SME number 1 Valid format for each term: SN.NNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_BUMP	Forward_Cross_ SME1_Bump	Dynamic	float64 array (6 values)	Fifth-order polynomial coefficients that describe the deviation of forward cross-scan mirror motion from linear; bumper mode with SME number 1 Valid format for each term: SN.NNNNNNESNN, where S = + or –, N = 0 to 9, and E = E

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_BUMP	Forward_Angle1_SME1_Bump	**For CPFs with effective dates prior to March 1, 2002**		
		Static	float32	Angle in μrad from the start of the scan to the mid-scan point in the forward direction; bumper mode with SME number 1 Valid format: NNNNN.N, where NNNNN.N = 67882.3
		For CPFs with effective dates of March 1, 2002, and thereafter		
		Dynamic	float32 array of flexible length	Angle in μrad from the start of the scan to the mid-scan point in the forward direction; bumper mode with SME number 1; the array contains daily values over one CPF interval Valid format for each term: NNNNN.N, where N = 0 to 9
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_BUMP	Forward_Angle2_SME1_Bump	**For CPFs with effective dates prior to March 1, 2002**		
		Static	float32	Angle in μrad from the mid-scan point to the end of the scan in the forward direction; bumper mode with SME number 1 Valid format: NNNNN.N, where NNNNN.N = 69730.9
		For CPFs with effective dates of March 1, 2002, and thereafter		
		Dynamic	float32 array of flexible length	Angle in μrad from the mid-scan point to the end of the scan in the forward direction; bumper mode with SME number 1; the array contains daily values over one CPF interval Valid format for each term: NNNNN.N, where N = 0 to 9.
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_BUMP	Forward_FHSERR_SME1_Bump (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Dynamic	int16 array of flexible length	First-half error of the forward-scan angle; bumper mode with SME number 1; array contains daily values over one CPF interval Valid format for each term: SNNNN, where S = + or –, and N = 0 to 9
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_BUMP	Forward_SHSERR_SME1_Bump (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Dynamic	int16 array of flexible length	Second-half error of the forward-scan angle; bumper mode with SME number 1; array contains daily values over one CPF interval Valid format for each term: SNNNN, where S = + or – and N = 0 to 9
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_BUMP	Reverse_Along_SME1_Bump	Dynamic	float64 array (6 values)	Fifth-order polynomial coefficients that describe the deviation of reverse along-scan mirror motion from linear; bumper mode with SME number 1 Valid format: SN.NNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_BUMP	Reverse_Cross_SME1_Bump	Dynamic	float64 array (6 values)	Fifth-order polynomial coefficients that describe the deviation of reverse cross-scan mirror motion from linear; bumper mode with SME number 1 Valid format: SN.NNNNNNESNN, where S = + or –, N = 0 to 9, and E = E

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_BUMP	Reverse_Angle1_SME1_Bump	**For CPFs with effective dates prior to March 1, 2002**		
		Static	float32	Angle in µrad from the start of the scan to the mid-scan point in the reverse direction; bumper mode with SME number 1 Valid format: NNNNN.N, where NNNNN.N = 67988.5
		For CPFs with effective dates of March 1, 2002, and thereafter		
		Dynamic	float32 array of flexible length	Angle in µrad from the start of the scan to the mid-scan point in the reverse direction; bumper mode with SME number 1; array contains daily values over one CPF interval Valid format for each term: NNNNN.N, where N = 0 to 9
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_BUMP	Reverse_Angle2_SME1_Bump	**For CPFs with effective dates prior to March 1, 2002**		
		Static	float32	Angle in µrad from the mid-scan point to the end of the scan in the reverse direction; bumper mode with SME number 1 Valid format: NNNNN.N, where NNNNN.N = 69301.4
		For CPFs with effective dates of March 1, 2002, and thereafter		
		Dynamic	float32 array of flexible length	Angle in µrad from the mid-scan point to the end of the scan in the reverse direction; bumper mode with SME number 1; the array contains daily values over one CPF interval Valid format for each term: NNNNN.N, where N = 0 to 9
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_BUMP	Reverse_FHSERR_SME1_Bump (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Dynamic	int16 array of flexible length	First-half error of the reverse-scan angle; bumper mode with SME number 1; array contains daily values over one CPF interval Valid format for each term: SNNNN, where S = + or –, and N = 0 to 9
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_BUMP	Reverse_SHSERR_SME1_Bump (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Dynamic	int16 array of flexible length	Second-half error of the reverse-scan angle; bumper mode with SME number 1; array contains daily values over one CPF interval Valid format for each term: SNNNN, where S = + or –, and N = 0 to 9
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_BUMP	Forward_Along_SME2_Bump	Dynamic	float64 array (6 values)	Fifth-order polynomial coefficients that describe the deviation of forward along-scan mirror motion from linear; bumper mode with SME number 2 Valid format: SN.NNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_BUMP	Forward_Cross_SME2_Bump	Dynamic	float64 array (6 values)	Fifth-order polynomial coefficients that describe the deviation of forward cross-scan mirror motion from linear; bumper mode with SME number 2 Valid format: SN.NNNNNNESNN, where S = + or –, N = 0 to 9, and E = E

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_BUMP	Forward_Angle1_SME2_Bump	**For CPFs with effective dates prior to March 1, 2002**		
		Static	float32	Angle in µrad from the start of the scan to the mid-scan point in the forward direction; bumper mode with SME number 2 Valid format: NNNNN.N, where NNNNN.N = 67882.3
		For CPFs with effective dates of March 1, 2002, and thereafter		
		Dynamic	float32 array of flexible length	Angle in µrad from the start of the scan to the mid-scan point in the forward direction; bumper mode with SME number 2; the array contains daily values over one CPF interval Valid format for each term: NNNNN.N, where N = 0 to 9
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_BUMP	Forward_Angle2_SME2_Bump	**For CPFs with effective dates prior to March 1, 2002**		
		Static	float32	Angle in µrad from the mid-scan point to the end of the scan in the forward direction; bumper mode with SME number 2 Valid format: NNNNN.N, where NNNNN.N = 69730.9
		For CPFs with effective dates of March 1, 2002, and thereafter		
		Dynamic	float32 array of flexible length	Angle in µrad from the mid-scan point to the end of the scan in the forward direction; bumper mode with SME number 2; the array contains daily values over one CPF interval Valid format for each term: NNNNN.N, where N = 0 to 9
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_BUMP	Forward_FHSERR_SME2_Bump (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Dynamic	int16 array of flexible length	First-half error of the forward-scan angle; bumper mode with SME number 2; array contains daily values over one CPF interval Valid format for each term: SNNNN, where S = + or –, and N = 0 to 9
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_BUMP	Forward_SHSERR_SME2_Bump (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Dynamic	int16 array of flexible length	Second-half error of the forward-scan angle; bumper mode with SME number 2; array contains daily values over one CPF interval Valid format for each term: SNNNN, where S = + or –, and N = 0 to 9
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_BUMP	Reverse_Along_SME2_Bump	Dynamic	float64 array (6 values)	Fifth-order polynomial coefficients that describe the deviation of reverse along-scan mirror motion from linear; bumper mode with SME number 2 Valid format for each term: SN.NNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_BUMP	Reverse_Cross_SME2_Bump	Dynamic	float64 array (6 values)	Fifth-order polynomial coefficients that describe the deviation of reverse cross-scan mirror motion from linear; bumper mode with SME number 2 Valid format for each term: SN.NNNNNNESNN, where S = + or –, N = 0 to 9, and E = E

Parameter Groups	**Parameter Name**	**Value Type**	**Data Type**	**Description**
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_BUMP	Reverse_Angle1_SME2_Bump	**For CPFs with effective dates prior to March 1, 2002**		
		Static	float32	Angle in µrad from the start of the scan to the mid-scan point in the reverse direction; bumper mode with SME number 2 Valid format: NNNNN.N, where NNNNN.N = 67988.5
		For CPFs with effective dates of March 1, 2002, and thereafter		
		Dynamic	float32 array of flexible length	Angle in µrad from the start of the scan to the mid-scan point in the reverse direction; bumper mode with SME number 2; array contains daily values over one CPF interval Valid format for each term: NNNNN.N, where N = 0 to 9
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_BUMP	Reverse_Angle2_SME2_Bump	**For CPFs with effective dates prior to March 1, 2002**		
		Static	float32	Angle in µrad from the mid-scan point to the end of the scan in the reverse direction; bumper mode with SME number 2 Valid format: NNNNN.N, where NNNNN.N = 69301.4
		For CPFs with effective dates of March 1, 2002, and thereafter		
		Dynamic	float32 array of flexible length	Angle in µrad from the mid-scan point to the end of the scan in the reverse direction; bumper mode with SME number 2; the array contains daily values over one CPF interval Valid format for each term: NNNNN.N, where N = 0 to 9
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_BUMP	Reverse_FHSERR_SME2_Bump (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Dynamic	int16 array of flexible length	First-half error of the reverse-scan angle; bumper mode with SME number 2; array contains daily values over one CPF interval Valid format for each term: SNNNN, where S = + or –, and N = 0 to 9
GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_BUMP	Reverse_SHSERR_SME2_Bump (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Dynamic	int16 array of flexible length	Second-half error of the reverse-scan angle; bumper mode with SME number 2; array contains daily values over one CPF interval Valid format for each term: SNNNN, where S = + or –, and N = 0 to 9
GROUP: BUMPER_MODE_PARAMETERS	SME1_BumperA_Dwell_Time (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Dynamic	float32 array of flexible length	"Physical" bumper mode mirror model parameter—time from the bumper A pickoff signal to the start of the reverse-scan linear motion in microseconds; array contains daily values over one CPF interval Valid format for each term: NNNNN.NN, where N = 0 to 9
GROUP: BUMPER_MODE_PARAMETERS	SME1_BumperA_Pickoff_Time (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Dynamic	float32 array of flexible length	"Physical" bumper mode mirror model parameter—time from the end of the forward-scan linear motion to bumper A pickoff signal in microseconds; array contains daily values over one CPF interval Valid format for each term: NNNNN.NN, where N = 0 to 9
GROUP: BUMPER_MODE_PARAMETERS	SME1_BumperA_Offset_Time (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Static	float32	"Physical" bumper mode mirror model parameter—time from the bumper A pickoff signal to the start of the reverse active scan in microseconds Valid format: NNNNN.NN, where NNNNN.NN = 10110.00

Parameter Groups	**Parameter Name**	**Value Type**	**Data Type**	**Description**
GROUP: BUMPER_MODE_ PARAMETERS	SME1_BumperA_Angle (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Static	float32	"Physical" bumper mode mirror model parameter—mirror field angle at which linear scanning motion begins (reverse) and ends (forward) at bumper A in microradians Valid format: SNNNNN.N, where SNNNNN.N = –68665.0
GROUP: BUMPER_MODE_ PARAMETERS	SME1_BumperB_Dwell_ Time (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Dynamic	float32 array of flexible length	"Physical" bumper mode mirror model parameter—time from the bumper B pickoff signal to the start of the forward-scan linear motion in microseconds; array contains daily values over one CPF interval Valid format for each term: NNNNN.NN, where N = 0 to 9
GROUP: BUMPER_MODE_ PARAMETERS	SME1_BumperB_Pickoff_ Time (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Dynamic	float32 array of flexible length	"Physical" bumper mode mirror model parameter—time from the end of the reverse-scan linear motion to the bumper B pickoff signal in microseconds; array contains daily values over one CPF interval Valid format for each term: NNNNN.NN, where N = 0 to 9
GROUP: BUMPER_MODE_ PARAMETERS	SME1_BumperB_Offset_ Time (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Static	float32	"Physical" bumper mode mirror model parameter—time from the bumper B pickoff signal to the start of the forward active scan in microseconds Valid format: NNNNN.NN, where NNNNN.NN = 10110.00
GROUP: BUMPER_MODE_ PARAMETERS	SME1_BumperB_Angle (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Static	float32	"Physical" bumper mode mirror model parameter—mirror field angle at which linear scanning motion begins (forward) and ends (reverse) at bumper B in microradians Valid format: SNNNNN.N, where SNNNNN.N = 68607.0
GROUP: BUMPER_MODE_ PARAMETERS	SME2_BumperA_Dwell_ Time (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Dynamic	float32 array of flexible length	"Physical" bumper mode mirror model parameter—time from the bumper A pickoff signal to the start of the reverse-scan linear motion in microseconds; array contains daily values over one CPF interval Valid format for each term: NNNNN.NN, where N = 0 to 9
GROUP: BUMPER_MODE_ PARAMETERS	SME2_BumperA_Pickoff_ Time (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Dynamic	float32 array of flexible length	"Physical" bumper mode mirror model parameter—time from the end of the forward-scan linear motion to the bumper A pickoff signal in microseconds; array contains daily values over one CPF interval Valid format for each term: NNNNN.NN, where N = 0 to 9
GROUP: BUMPER_MODE_ PARAMETERS	SME2_BumperA_Offset_ Time (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Static	float32	"Physical" bumper mode mirror model parameter—time from the bumper A pickoff signal to the start of the reverse active scan in microseconds Valid format: NNNNN.NN, where NNNNN.NN = 10110.00
GROUP: BUMPER_MODE_ PARAMETERS	SME2_BumperA_Angle (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Static	float32	"Physical" bumper mode mirror model parameter—mirror field angle at which linear scanning motion begins (reverse) and ends (forward) at bumper A in microradians Valid format: SNNNNN.N, where SNNNNN.N = –68665.0

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: BUMPER_MODE_PARAMETERS	SME2_BumperB_Dwell_Time (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Dynamic	float32 array of flexible length	"Physical" bumper mode mirror model parameter—time from the bumper B pickoff signal to the start of the forward-scan linear motion in microseconds; array contains daily values over one CPF interval Valid format for each term: NNNNN.NN, where N = 0 to 9
GROUP: BUMPER_MODE_PARAMETERS	SME2_BumperB_Pickoff_Time (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Dynamic	float32 array of flexible length	"Physical" bumper mode mirror model parameter—time from the end of the reverse-scan linear motion to the bumper B pickoff signal in microseconds; array contains daily values over one CPF interval Valid format for each term: NNNNN.NN, where N = 0 to 9
GROUP: BUMPER_MODE_PARAMETERS	SME2_BumperB_Offset_Time (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Static	float32	"Physical" bumper mode mirror model parameter—time from the bumper B pickoff signal to the start of the forward active scan in microseconds Valid format: NNNNN.NN, where NNNNN.NN = 10110.00
GROUP: BUMPER_MODE_PARAMETERS	SME2_BumperB_Angle (available in all CPFs with effective dates of March 1, 2002, and thereafter)	Static	float32	"Physical" bumper mode mirror model parameter—mirror field angle at which linear scanning motion begins (forward) and ends (reverse) at bumper B in microradians Valid format: SNNNNN.N, where SNNNNN.N = 68607.0
GROUP: SCAN_LINE_CORRECTOR	Primary_Angular_Velocity	Static	float32	Angular velocity in radians per second of the primary scan line corrector Valid format: N.NNNNN, where N.NNNNN = 0.00966
GROUP: SCAN_LINE_CORRECTOR	Secondary_Angular_Velocity	Static	float32	Angular velocity in radians per second of the secondary scan line corrector Valid format: N.NNNNN, where N.NNNNN = 0.00960
GROUP: SCAN_LINE_CORRECTOR	Primary_Corrector_Motion	Static	float32 array (6 values)	Fifth-order polynomial coefficients that describe the motion of the primary scan line corrector Valid format for each term: N.NNNNN, where N = 0 to 9
GROUP: SCAN_LINE_CORRECTOR	Secondary_Corrector_Motion	Static	float32 array (6 values)	Fifth-order polynomial coefficients that describe the motion of the secondary scan line corrector Valid format for each term: N.NNNNN, where N = 0 to 9
GROUP: SCAN_LINE_CORRECTOR	Unpowered_Pointing_Bias	Dynamic	Float32	The best estimate of the pointing angle of the scan line corrector in its unpowered, “at-rest” pointing position Valid format: N.NNNNNNN, where N.NNNNNNN = 0.0000000
GROUP: FOCAL_PLANE_PARAMETERS GROUP: BAND_OFFSETS	Along_Scan_Band_Offsets	Static	float32 array (7 values)	Nominal displacement in µrad from the center of the focal plane to each Band's optical axis Valid format: SNNNN.NNN, where S = + or –, and N = 0 to 9
GROUP: FOCAL_PLANE_PARAMETERS GROUP: BAND_OFFSETS	Across_Scan_Band_Offsets	Static	float32 array (7 values)	Nominal displacement in µrad from the center of the focal plane to each band's scan motion axis Valid format: SNNNN.NNN, S = + or –, and N = 0 to 9
GROUP: FOCAL_PLANE_PARAMETERS GROUP: BAND_OFFSETS	Forward_Focal_Plane_Offsets	Static	float32 array (7 values)	Offset in Instrument Fields of View (IFOVs) for focal plane forward scans Valid format: SNNN.N, where S = + or –, and N = 0 to 9
GROUP: FOCAL_PLANE_PARAMETERS GROUP: BAND_OFFSETS	Reverse_Focal_Plane_Offsets	Static	float32 array (7 values)	Offset in IFOVs for focal plane reverse scans Valid format: SNNN.N, where S = + or –, and N = 0 to 9

Parameter Groups	**Parameter Name**	**Value Type**	**Data Type**	**Description**
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Forward_Along_ Scan_DO_B1	Static	float32 array (16 values)	Forward along-scan detector offsets in IFOV for each detector in Band 1 Valid format: N.NNN, where N = 0 to 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Reverse_Along_ Scan_DO_B1	Static	float32 array (16 values)	Reverse along-scan detector offsets in IFOV for each detector in Band 1 Valid format: N.NNN, where N = 0 to 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Forward_Along_ Scan_DO_B2	Static	float32 array (16 values)	Forward along-scan detector offsets in IFOV for each detector in Band 2 Valid format: N.NNN, where N = 0 to 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Reverse_Along_ Scan_DO_B2	Static	float32 array (16 values)	Reverse along-scan detector offsets in IFOV for each detector in Band 2 Valid format: N.NNN, where N = 0 to 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Forward_Along_ Scan_DO_B3	Static	float32 array (16 values)	Forward along-scan detector offsets in IFOV for each detector in Band 3 Valid format: N.NNN, where N = 0 to 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Reverse_Along_ Scan_DO_B3	Static	float32 array (16 values)	Reverse along-scan detector offsets in IFOV for each detector in Band 3 Valid format: N.NNN, where N = 0 to 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Forward_Along_ Scan_DO_B4	Static	float32 array (16 values)	Forward along-scan detector offsets in IFOV for each detector in Band 4 Valid format: N.NNN, where N = 0 to 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Reverse_Along_ Scan_DO_B4	Static	float32 array (16 values)	Reverse along-scan detector offsets in IFOV for each detector in Band 4 Valid format: N.NNN, where N = 0 to 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Forward_Along_ Scan_DO_B5	Static	float32 array (16 values)	Forward along-scan detector offsets in IFOV for each detector in Band 5 Valid format: N.NNN, where N = 0 to 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Reverse_Along_ Scan_DO_B5	Static	float32 array (16 values)	Reverse along-scan detector offsets in IFOV for each detector in Band 5 Valid format: N.NNN, where N = 0 to 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Forward_Along_ Scan_DO_B6	Static	float32 array (4 values)	Forward along-scan detector offsets in IFOV for each detector in Band 6 Valid format: N.NNN, where N = 0 to 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Reverse_Along_ Scan_DO_B6	Static	float32 array (4 values)	Reverse along-scan detector offsets in IFOV for each detector in Band 6 Valid format: N.NNN, where N = 0 to 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Forward_Along_ Scan_DO_B7	Static	float32 array (16 values)	Forward along-scan detector offsets in IFOV for each detector in Band 7 Valid format: N.NNN, where N = 0 to 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Reverse_Along_ Scan_DO_B7	Static	float32 array (16 values)	Reverse along-scan detector offsets in IFOV for each detector in Band 7 Valid format: N.NNN, where N = 0 to 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Forward_Across_ Scan_DO_B1	Static	float32 array (16 values)	Forward across-scan detector offsets in IFOV for each detector in Band 1 Valid format: N.NNN, where N = 0 TO 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Reverse_Across_ Scan_DO_B1	Static	float32 array (16 values)	Reverse across-scan detector offsets in IFOV for each detector in Band 1 Valid format: N.NNN, where N = 0 TO 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Forward_Across_ Scan_DO_B2	Static	float32 array (16 values)	Forward across-scan detector offsets in IFOV for each detector in Band 2 Valid format: N.NNN, where N = 0 TO 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Reverse_Across_ Scan_DO_B2	Static	float32 array (16 values)	Reverse across-scan detector offsets in IFOV for each detector in Band 2 Valid format: N.NNN, where N = 0 TO 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Forward_Across_ Scan_DO_B3	Static	float32 array (16 values)	Forward across-scan detector offsets in IFOV for each detector in Band 3 Valid format: N.NNN, where N = 0 TO 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Reverse_Across_ Scan_DO_B3	Static	float32 array (16 values)	Reverse across-scan detector offsets in IFOV for each detector in Band 3 Valid format: N.NNN, where N = 0 TO 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Forward_Across_ Scan_DO_B4	Static	float32 array (16 values)	Forward across-scan detector offsets in IFOV for each detector in Band 4 Valid format: N.NNN, where N = 0 TO 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Reverse_Across_ Scan_DO_B4	Static	float32 array (16 values)	Reverse across-scan detector offsets in IFOV for each detector in Band 4 Valid format: N.NNN, where N = 0 TO 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Forward_Across_ Scan_DO_B5	Static	float32 array (16 values)	Forward across-scan detector offsets in IFOV for each detector in Band 5 Valid format: N.NNN, where N = 0 TO 9
FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Reverse_Across_ Scan_DO_B5	Static	float32 array (16 values)	Reverse across-scan detector offsets in IFOV for each detector in Band 5 Valid format: N.NNN, where N = 0 TO 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Forward_Across_Scan_D O_B6	Static	float32 array (4 values)	Forward across-scan detector offsets in IFOV for each detector in Band 6 Valid format: N.NNN, where N = 0 TO 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Reverse_Across_ Scan_DO_B6	Static	float32 array (4 values)	Reverse across-scan detector offsets in IFOV for each detector in Band 6 Valid format: N.NNN, where N = 0 TO 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Forward_Across_ Scan_DO_B7	Static	float32 array (16 values)	Forward across-scan detector offsets in IFOV for each detector in Band 7 Valid format: N.NNN, where N = 0 TO 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS	Reverse_Across_ Scan_DO_B7	Static	float32 array (16 values)	Reverse across-scan detector offsets in IFOV for each detector in Band 7 Valid format: N.NNN, where N = 0 TO 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: ODD_EVEN_OFFSETS	Forward_Even_ Detector_Shift	Static	float32 array (7 values)	Adjustments in IFOVs to compensate for forward scan band offsets, even detector layout geometry, and multiplexer sampling for Bands 1 through 7 Valid format: NNN.N, where N = 0 TO 9

Parameter Groups	**Parameter Name**	**Value Type**	**Data Type**	**Description**
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: ODD_EVEN_OFFSETS	Forward_Odd_ Detector_Shift	Static	float32 array (7 values)	Adjustments in IFOVs to compensate for forward scan band offsets, odd detector layout geometry, and multiplexer sampling for Bands 1 through 7 Valid format: NNN.N, where N = 0 TO 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: ODD_EVEN_OFFSETS	Reverse_Even_ Detector_Shift	Static	float32 array (7 values)	Adjustments in IFOVs to compensate for reverse scan band offsets, even detector layout geometry, and multiplexer sampling for Bands 1 through 7 Valid format: NNN.N, where N = 0 TO 9
GROUP: FOCAL_PLANE_ PARAMETERS GROUP: ODD_EVEN_OFFSETS	Reverse_Odd_ Detector_Shift	Static	float32 array (7 values)	Adjustments in IFOVs to compensate for reverse scan band offsets, odd detector layout geometry, and multiplexer sampling for Bands 1 through 7 Valid format: NNN.N, where N = 0 TO 9
GROUP: ATTITUDE_PARAMETERS	Gyro_To_Attitude_ Matrix	Static	float32 array (9 values)	Matrix describing the relationship of the gyro axis to the attitude control reference axis Valid format: SN.NNNNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: ATTITUDE_PARAMETERS	ADSA_To_TM_ Matrix	Static	float32 array (9 values)	Matrix describing the relationship of the Attitude Displacement Sensor Assembly (ADSA) to the TM+ optical Axis Valid format: SN.NNNNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: ATTITUDE_PARAMETERS	Attitude_To_TM_ Matrix	Dynamic	float32 array (9 values)	Matrix describing the relationship of the attitude control reference axis to the TM optical axis Valid format: SN.NNNNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: ATTITUDE_PARAMETERS	Spacecraft_Roll_Bias	Static	float32	Spacecraft roll bias in radians Valid format: N.NNNNNNN, where N.NNNNNNN = 0.0000000
GROUP: ATTITUDE_PARAMETERS	Spacecraft_Pitch_ Bias	Static	float32	Spacecraft pitch bias in radians Valid format: N.NNNNNNN, where N.NNNNNNN = 0.0000000
GROUP: ATTITUDE_PARAMETERS	Spacecraft_Yaw_Bias	Static	float32	Spacecraft yaw bias in radians Valid format: N.NNNNNNN, where N.NNNNNNN = 0.0000000
GROUP: TIME_PARAMETERS	Scan_Time	Static	float32	Nominal scan time in microseconds Valid format: NNNNN.N, where NNNNN.N = 60743.0
GROUP: TIME_PARAMETERS	Forward_First_Half_ Time	Static	float32	Nominal forward first half scan time in microseconds Valid format: NNNNN.N, where NNNNN.N = 30371.4
GROUP: TIME_PARAMETERS	Forward_Second_ Half_Time	Static	float32	Nominal forward second half scan time in microseconds Valid format: NNNNN.N, where NNNNN.N = 30371.6
GROUP: TIME_PARAMETERS	Reverse_First_Half_ Time	Static	float32	Nominal reverse first half scan time in microseconds Valid format: NNNNN.N, where NNNNN.N = 30371.6
GROUP: TIME_PARAMETERS	Reverse_Second_ Half_Time	Static	float32	Nominal reverse second half scan time in microseconds Valid format: NNNNN.N, where NNNNN.N = 30371.4
GROUP: TRANSFER_FUNCTION GROUP: IMU	Fn	Static	float64	Inertial measurement unit transfer function resonant frequency (Hz) Valid format: N.NNNN, where N.NNNN = 2.2010
GROUP: TRANSFER_FUNCTION GROUP: IMU	Zeta	Static	float64	Inertial measurement unit transfer function damping coefficient Valid format: N.NNNN, where N.NNNN = 0.7022

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: TRANSFER_FUNCTION GROUP: IMU	Tau	Static	float64	Inertial measurement unit transfer function denominator time constant (seconds) Valid format: NN.NNNNESN, where NN.NNNNESN = 11.4468E-3
GROUP: TRANSFER_FUNCTION GROUP: IMU	P	Static	float64	Inertial measurement unit transfer function numerator time constant (seconds) Valid format: SN.NNNNESN, where SN.NNNNESN = –3.2590E-3
GROUP: TRANSFER_FUNCTION GROUP: IMU	Ak	Static	float64	Inertial measurement unit transfer function Direct Current (DC) gain Valid format: N.NNNNN, where N.NNNNN = 1.00518
GROUP: TRANSFER_FUNCTION GROUP: ADS	ADS_num	Static	float64 array (18 values)	Transfer function numerator coefficients in order a0, a1, a2, a3, a4, a5; one set of six coefficients for each of three ADS units; determined at 15 degrees C Valid format: N.NNNNEN, where N = 0 to 9, and E = E
GROUP: TRANSFER_FUNCTION GROUP: ADS	ADS_den	Static	float64 array (18 values)	Transfer function denominator coefficients in order b0, b1, b2, b3, b4, b5; one set of six coefficients for each of three ADS units; determined at 15 degrees C Valid format: N.NNNNEN, where N = 0 to 9, and E = E
GROUP: TRANSFER_FUNCTION GROUP: ADS	ADS_num_temp	Static	float64 array (18 values)	Temperature-dependent part of the ADS transfer function numerator coefficients in order da0, da1, da2, da3, da4, da5; one set of six coefficients for each of three ADS units; change per degree C Valid format: N.NNNNEN, where N = 0 to 9, and E = E
GROUP: TRANSFER_FUNCTION GROUP: ADS	ADS_den_temp	Static	float64 array (18 values)	Temperature-dependent part of the ADS transfer function denominator coefficients in order da0, da1, da2, da3, da4, da5; one set of six coefficients for each of three ADS units; change per degree C Valid format: N.NNNNEN, where N = 0 to 9, and E = E
GROUP: TRANSFER_FUNCTION GROUP: PREFILTER	ADSPre_W	Static	float64 array (5 values)	ADS prefilter transfer function quadratic term resonant periods (Note: Given as period instead of frequency so that the transfer function can be set to unity, if necessary, by setting all five values to zero.) Valid format: N.N, where N = 0 to 9
GROUP: TRANSFER_FUNCTION GROUP: PREFILTER	ADSPre_H	Static	float64 array (5 values)	ADS prefilter transfer function quadratic term damping coefficients Valid format: N.N, where N = 0 to 9
GROUP: TRANSFER_FUNCTION GROUP: PREFILTER	ADSPre_T	Static	float64 array (5 values)	ADS prefilter transfer function linear term time constants Valid format: N.N, where N = 0 to 9
GROUP: UT1_TIME_PARAMETERS	UT1_Year	Dynamic	int16 array (180 values)	Year of UT1 time correction prediction; values span 180 days Valid format: YYYY, where YYYY = 1984-2012
GROUP: UT1_TIME_PARAMETERS	UT1_Month	Dynamic	char8 array (180 values)	Month of UT1 time correction prediction; values span 180 days Valid format: MMM, where MMM = Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, or Dec
GROUP: UT1_TIME_PARAMETERS	UT1_Day	Dynamic	uint8 array (180 values)	Day of UT1 time correction prediction; values span 180 days Valid format: NN, where NN = 1-31

Parameter Groups	**Parameter Name**	**Value Type**	**Data Type**	**Description**
GROUP: UT1_TIME_PARAMETERS	UT1_Modified_Julian	Dynamic	int32 array (180 values)	Modified Julian day; values span 180 days; MJD = Julian day – 2 400 000.5; Julian date is a running day count starting 1 January 4713 B.C. Valid format: NNNNN, where NNNNN = e.g., 50234 (for May 31, 1996)
GROUP: UT1_TIME_PARAMETERS	UT1_X	Dynamic	float32 array (180 values)	X shift pole wander in arc seconds; values span 180 days Valid format: N.NNNNN, where N.NNNNN = e.g., 0.45431
GROUP: UT1_TIME_PARAMETERS	UT1_Y	Dynamic	float32 array (180 values)	Y shift pole wander in arc seconds; values span 180 days Valid format: N.NNNNN, where N.NNNNN = e.g., 0.13454
GROUP: UT1_TIME_PARAMETERS	UT1_UTC	Dynamic	float32 array (180 values)	UT1 – UTC time difference in seconds Values span 180 days Valid format: N.NNNNN, where SN.NNNNN = e.g., –0.44321
GROUP: TIME_SINCE_LAUNCH	Decimal_Years	Dynamic	float32 array of flexible length	Day since the satellite's launch expressed in decimal years; array contains daily values over a given CPF interval Valid format: NNNN.NNNN, where NNNN.NNNN = e.g., 1984.1667 (for March 1, 1984)
GROUP: TIME_SINCE_LAUNCH	Days_Since_Launch	Dynamic	int32 array of flexible length	Day since the satellite's launch with March 1, 1984, being equal to day 1; array contains daily values over a given CPF interval Valid format: NNNN, where N = 0 to 9
GROUP: TIME_SINCE_LAUNCH	Day_Of_Year	Dynamic	int16 array of flexible length	Day of the current year; array contains daily values over a given CPF interval Valid format: NNN, where N = 0 to 9

GROUP: DETECTOR_STATUS	Status_Band1	Dynamic	char8 array (16 values)	Health status of Band 1's 16 detectors
				Valid format: ABCDE, where
				A = 0 (live), 1 (dead), 2 (intermittent)
				B = 0 (noise in spec), 1 (noisy low signal), 2 (noisy high signal), 3 (both noisy signals)
				C = 0 (reserved)
				D = 0 (dynamic range in spec) 1 (fail, high end), 2 (fail, low end), 3 (fail, both ends)
				E = 0 (reserved)

GROUP: DETECTOR_STATUS	Status_Band2	Dynamic	char8 array (16 values)	Health status of Band 2's 16 detectors Valid format: same as above
GROUP: DETECTOR_STATUS	Status_Band3	Dynamic	char8 array (16 values)	Health status of Band 3's 16 detectors Valid format: same as above
GROUP: DETECTOR_STATUS	Status_Band4	Dynamic	char8 array (16 values)	Health status of Band 4's 16 detectors Valid format: same as above
GROUP: DETECTOR_STATUS	Status_Band5	Dynamic	char8 array (16 values)	Health status of Band 5's 16 detectors Valid format: as above
GROUP: DETECTOR_STATUS	Status_Band6	Dynamic	char8 array (4 values)	Health status of Band 6's 4 detectors Valid format: same as above
GROUP: DETECTOR_STATUS	Status_Band7	Dynamic	char8 array (16 values)	Health status of Band 7's 16 detectors Valid format: same as above
GROUP: DETECTOR_GAINS GROUP: GAIN_MODEL_PARAMETERS	Band_1_Normalized_ IC_Model_Coefficients	Dynamic	float32 array (4 values)	Normalized lifetime gain model coefficients derived from detector responses to the internal calibrator lamp 2 (lamp state [010]) for Band 1 Valid format: N.NNNNNNNN, where N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: DETECTOR_GAINS GROUP: GAIN_MODEL_PARAMETERS	Band_2_Normalized_ IC_Model_Coefficients	Dynamic	float32 array (4 values)	Normalized lifetime gain model coefficients derived from detector responses to the internal calibrator lamp 2 (lamp state [010]) for Band 1 Valid format: N.NNNNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: GAIN_MODEL_PARAMETERS	Band_3_Normalized_ IC_Model_Coefficients	Dynamic	float32 array (4 values)	Normalized lifetime gain model coefficients derived from detector responses to the internal calibrator lamp 2 (lamp state [010]) for Band 1 Valid format: N.NNNNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: GAIN_MODEL_PARAMETERS	Band_4_Normalized_ IC_Model_Coefficients	Dynamic	float32 array (4 values)	Normalized lifetime gain model coefficients derived from detector responses to the internal calibrator lamp 2 (lamp state [010]) for Band 1 Valid format: N.NNNNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: GAIN_MODEL_PARAMETERS	Band_5_Normalized_ IC_Model_Coefficients	Dynamic	float32 array (4 values)	Normalized lifetime gain model coefficients derived from detector responses to the internal calibrator lamp 2 (lamp state [010]) for Band 1 Valid format: N.NNNNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: GAIN_MODEL_PARAMETERS	Band_6_Normalized_ IC_Model_Coefficients	Dynamic	float32 array (4 values)	Normalized lifetime gain model coefficients derived from detector responses to the internal calibrator blackbody for Band 6 Valid format: N.NNNNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: GAIN_MODEL_PARAMETERS	Band_7_Normalized_ IC_Model_Coefficients	Dynamic	float32 array (4 values)	Normalized lifetime gain model coefficients derived from detector responses to the internal calibrator lamp 2 (lamp state [010]) for Band 1 Valid format: N.NNNNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: GAIN_MODEL_PARAMETERS	Time_Zero	Static	float32	Date in decimal years when the first scene used in derivation of the normalized lifetime models was acquired Valid format: NNNN.NNN, where NNNN.NNN = 1984.207
GROUP: DETECTOR_GAINS GROUP: GAIN_MODEL_PARAMETERS	Band_1_LT_Model_ Coefficients	Dynamic	float32 array (15 values)	Absolute radiometric gain model parameters for Band 1 Valid format: NN.NNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: GAIN_MODEL_PARAMETERS	Band_2_LT_Model_ Coefficients	Dynamic	float32 array (15 values)	Absolute radiometric gain model parameters for Band 2 Valid format: NN.NNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: GAIN_MODEL_PARAMETERS	Band_3_LT_Model_ Coefficients	Dynamic	float32 array (15 values)	Absolute radiometric gain model parameters for Band 3 Valid format: NN.NNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: GAIN_MODEL_PARAMETERS	Band_4_LT_Model_ Coefficients	Dynamic	float32 array (15 values)	Absolute radiometric gain model parameters for Band 4 Valid format: NN.NNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: GAIN_MODEL_PARAMETERS	Band_5_LT_Model_ Coefficients	Dynamic	float32 array (15 values)	Absolute radiometric gain model parameters for Band 5 Valid format: NN.NNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: GAIN_MODEL_PARAMETERS	Band_6_LT_Model_ Coefficients	Dynamic	float32 array (15 values)	Absolute radiometric gain model parameters for Band 6 Valid format: NN.NNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: GAIN_MODEL_PARAMETERS	Band_7_LT_Model_ Coefficients	Dynamic	float32 array (15 values)	Absolute radiometric gain model parameters for Band 7 Valid format: NN.NNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Outgassing_Events	Dynamic	int16 array (50 values)	Imaging start days, in days-since-launch, following the outgassing events Valid format: NNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_5_Film_Refractive_ Index_Part_1	Dynamic	float32 array (16 values)	Index of refraction for the contaminant, as used in the single outgassing cycle thin-film models for Band 5, for the time from the beginning of a calendar quarter until the next outgassing event; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_5_Film_Absorption_Index_Part_1	Dynamic	float32 array (16 values)	Index of absorption for the contaminant, as used in the single outgassing cycle thin-film models for Band 5, for the time from the beginning of a calendar quarter until the next outgassing event; array contains one value per detector Valid format: N.NNESN, where S = + or –, N = 0 to 9, and E = E
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_5_ARC_Refractive_Index_Part_1	Dynamic	float32 array (16 values)	Index of refraction for the antireflective coating, as used in the single outgassing cycle thin-film models for Band 5, for the time from the beginning of a calendar quarter until the next outgassing event; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_5_ARC_Thickness_Part_1	Dynamic	float32 array (16 values)	Thickness of the antireflective coating in nanometers (nm), as used in the single outgassing cycle thin-film models for Band 5, for the time from the beginning of a calendar quarter until the next outgassing event; array contains one value per detector Valid format: NNN.N, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_5_Oscillating_Period_Part_1	Dynamic	float32 array (16 values)	Period of gain oscillations in days, as used in the single outgassing cycle thin-film models for Band 5, for the time from the beginning of a calendar quarter until the next outgassing event; array contains one value per detector Valid format: NNN.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_7_Film_Refractive_Index_Part_1	Dynamic	float32 array (16 values)	Index of refraction for the contaminant, as used in the single outgassing cycle thin-film models for Band 7, for the time from the beginning of a calendar quarter until the next outgassing event; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_7_Film_Absorption_Index_Part_1	Dynamic	float32 array (16 values)	Index of absorption for the contaminant, as used in the single outgassing cycle thin-film models for Band 7, for the time from the beginning of a calendar quarter until the next outgassing event; array contains one value per detector Valid format: N.NNESN, where S = + or –, N = 0 to 9, and E = E
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_7_ARC_Refractive_Index_Part_1	Dynamic	float32 array (16 values)	Index of refraction for the antireflective coating, as used in the single outgassing cycle thin-film models for Band 7, for the time from the beginning of a calendar quarter until the next outgassing event; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_7_ARC_Thickness_Part_1	Dynamic	float32 array (16 values)	Thickness of the antireflective coating in nm, as used in the single outgassing cycle thin-film models for Band 7, for the time from the beginning of a calendar quarter until the next outgassing event; array contains one value per detector Valid format: NNN.N, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_7_Oscillating_Period_Part_1	Dynamic	float32 array (16 values)	Period of gain oscillations in days, as used in the single outgassing cycle thin-film models for Band 7, for the time from the beginning of a calendar quarter until the next outgassing event; array contains one value per detector Valid format: NNN.NN, where N = 0 to 9

Parameter Groups	**Parameter Name**	**Value Type**	**Data Type**	**Description**
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_5_Film_Refractive_ Index_Part_2	Dynamic	float32 array (16 values)	Index of refraction for the contaminant, as used in the single outgassing cycle thin-film models for Band 5, for the time from the first outgassing event that occurred in a given quarter to the next one; if no outgassing was performed in a given quarter, the values are the same as in Part 1; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_5_Film_Absorption_ Index_Part_2	Dynamic	float32 array (16 values)	Index of absorption for the contaminant, as used in the single outgassing cycle thin-film models for Band 5, for the time from the first outgassing event that occurred in a given quarter to the next one; if no outgassing was performed in a given quarter, the values are the same as in Part 1; array contains one value per detector Valid format: N.NNESN, where S = + or –, N = 0 to 9, and E = E
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_5_ARC_Refractive_ Index_Part_2	Dynamic	float32 array (16 values)	Index of refraction for the antireflective coating, as used in the single outgassing cycle thin-film models for Band 5, for the time from the first outgassing event that occurred in a given quarter to the next one; if no outgassing was performed in a given quarter, the values are the same as in Part 1; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_5_ARC_Thickness_ Part_2	Dynamic	float32 array (16 values)	Thickness of the antireflective coating in nm, as used in the single outgassing cycle thin-film models for Band 5, for the time from the first outgassing event that occurred in a given quarter to the next one; if no outgassing was performed in a given quarter, the values are the same as in Part 1; array contains one value per detector Valid format: NNN.N, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_5_Oscillating_ Period_Part_2	Dynamic	float32 array (16 values)	Period of gain oscillations in days, as used in the single outgassing cycle thin-film models for Band 5, for the time from the first outgassing event that occurred in a given quarter to the next one; if no outgassing was performed in a given quarter, the values are the same as in Part 1; array contains one value per detector Valid format: NNN.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_7_Film_Refractive_ Index_Part_2	Dynamic	float32 array (16 values)	Index of refraction for the contaminant, as used in the single outgassing cycle thin-film models for Band 7, for the time from the first outgassing event that occurred in a given quarter to the next one; if no outgassing was performed in a given quarter, the values are the same as in Part 1; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_7_Film_Absorption_ Index_Part_2	Dynamic	float32 array (16 values)	Index of absorption for the contaminant, as used in the single outgassing cycle thin-film models for Band 7, for the time from the first outgassing event that occurred in a given quarter to the next one; if no outgassing was performed in a given quarter, the values are the same as in Part 1; array contains one value per detectors Valid format: N.NNESN, where S = + or –, N = 0 to 9, and E = E

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_7_ARC_Refractive_ Index_Part_2	Dynamic	float32 array (16 values)	Index of refraction for the antireflective coating, as used in the single outgassing cycle thin-film models for Band 7, for the time from the first outgassing event that occurred in a given quarter to the next one; if no outgassing was performed in a given quarter, the values are the same as in Part 1; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_7_ARC_Thickness_ Part_2	Dynamic	float32 array (16 values)	Thickness of the antireflective coating in nm, as used in the single outgassing cycle thin-film models for Band 7, for the time from the first outgassing event that occurred in a given quarter to the next one; if no outgassing was performed in a given quarter, the values are the same as in Part 1; array contains one value per detector Valid format: NNN.N, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_7_Oscillating_ Period_Part_2	Dynamic	float32 array (16 values)	Period of gain oscillations in days, as used in the single outgassing cycle thin-film models for Band 7, for the time from the first outgassing event that occurred in a given quarter to the next one; if no outgassing was performed in a given quarter, the values are the same as in Part 1; array contains one value per detector Valid format: NNN.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_5_Film_Refractive_ Index_Part_3	Dynamic	float32 array (16 values)	Index of refraction for the contaminant, as used in the single outgassing cycle thin-film models for Band 5, for the time from the second outgassing event that occurred in a given quarter to the next one; if no second outgassing was performed in a given quarter, the values are the same as in part 2; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_5_Film_Absorption_ Index_Part_3	Dynamic	float32 array (16 values)	Index of absorption for the contaminant, as used in the single outgassing cycle thin-film models for Band 5, for the time from the second outgassing event that occurred in a given quarter to the next one; if no second outgassing was performed in a given quarter, the values are the same as in part 2; array contains one value per detector Valid format: N.NNESN, where S = + or –, N = 0 to 9, and E = E
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_5_ARC_Refractive_ Index_Part_3	Dynamic	float32 array (16 values)	Index of refraction for the antireflective coating, as used in the single outgassing cycle thin-film models for Band 5, for the time from the second outgassing event that occurred in a given quarter to the next one; if no second outgassing was performed in a given quarter, the values are the same as in part 2; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_5_ARC_Thickness_ Part_3	Dynamic	float32 array (16 values)	Thickness of the antireflective coating in nm, as used in the single outgassing cycle thin-film models for Band 5, for the time from the second outgassing event that occurred in a given quarter to the next one; if no second outgassing was performed in a given quarter, the values are the same as in part 2; array contains one value per detector Valid format: NNN.N, where N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_5_Oscillating_ Period_Part_3	Dynamic	float32 array (16 values)	Period of gain oscillations in days, as used in the single outgassing cycle thin-film models for Band 5, for the time from the second outgassing event that occurred in a given quarter to the next one; if no second outgassing was performed in a given quarter, the values are the same as in part 2; array contains one value per detector Valid format: NNN.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_7_Film_Refractive_ Index_Part_3	Dynamic	float32 array (16 values)	Index of refraction for the contaminant, as used in the single outgassing cycle thin-film models for Band 7, for the time from the second outgassing event that occurred in a given quarter to the next one; if no second outgassing was performed in a given quarter, the values are the same as in part 2; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_7_Film_Absorption_ Index_Part_3	Dynamic	float32 array (16 values)	Index of absorption for the contaminant, as used in the single outgassing cycle thin-film models for Band 7, for the time from the second outgassing event that occurred in a given quarter to the next one; if no second outgassing was performed in a given quarter, the values are the same as in part 2; array contains one value per detector Valid format: N.NNESN, where S = + or –, N = 0 to 9, and E = E
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_7_ARC_Refractive_ Index_Part_3	Dynamic	float32 array (16 values)	Index of refraction for the antireflective coating, as used in the single outgassing cycle thin-film models for Band 7, for the time from the second outgassing event that occurred in a given quarter to the next one; if no second outgassing was performed in a given quarter, the values are the same as in part 2; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_7_ARC_Thickness_ Part_3	Dynamic	float32 array (16 values)	Thickness of the antireflective coating in nm, as used in the single outgassing cycle thin-film models for Band 7, for the time from the second outgassing event that occurred in a given quarter to the next one; if no second outgassing was performed in a given quarter, the values are the same as in part 2; array contains one value per detector Valid format: NNN.N, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_7_Oscillating_ Period_Part_3	Dynamic	float32 array (16 values)	Period of gain oscillations in days, as used in the single outgassing cycle thin-film models for Band 7, for the time from the second outgassing event that occurred in a given quarter to the next one; if no second outgassing was performed in a given quarter, the values are the same as in part 2; array contains one value per detector Valid format: NNN.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_5_Period_LT_ Model_Exp_Scaling	Dynamic	float32 array (16 values)	Scaling factor in models of change in the period of gain oscillations in Band 5 over the TM's lifetime; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_5_Period_LT_ Model_Attenuation	Dynamic	float32 array (16 values)	Attenuation factor in models of change in the period of gain oscillations in Band 5 over the TM's lifetime; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_5_Period_LT_ Model_Slope	Dynamic	float32 array (16 values)	Slope in models of change in the period of gain oscillations in Band 5 over the TM's lifetime; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_5_Period_LT_ Model_Offset	Dynamic	float32 array (16 values)	Intercept in models of change in the period of gain oscillations in Band 5 over the TM's lifetime; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_7_Period_LT_ Model_Exp_Scaling	Dynamic	float32 array (16 values)	Scaling factor in models of change in the period of gain oscillations in Band 7 over the TM's lifetime; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_7_Period_LT_ Model_Attenuation	Dynamic	float32 array (16 values)	Attenuation factor in models of change in the period of gain oscillations in Band 7 over the TM's lifetime; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_7_Period_LT_ Model_Slope	Dynamic	float32 array (16 values)	Slope in models of change in the period of gain oscillations in Band 7 over the TM's lifetime; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: OUTGASSING_CORRECTION	Band_7_Period_LT_ Model_Offset	Dynamic	float32 array (16 values)	Intercept in models of change in the period of gain oscillations in Band 7 over the TM's lifetime; array contains one value per detector Valid format: N.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: BAND_AVERAGE_GAINS	Band_1_Average_Gain	Dynamic	float32 array of flexible length	Band 1 detector-averaged gain in counts/W/m^2-ster-µm; array contains daily values over a given CPF interval Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: BAND_AVERAGE_GAINS	Band_2_Average_Gain	Dynamic	float32 array of flexible length	Band 2 detector-averaged gain in counts/W/m^2-ster-µm; array contains daily values over a given CPF interval Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: BAND_AVERAGE_GAINS	Band_3_Average_Gain	Dynamic	float32 array of flexible length	Band 3 detector-averaged gain in counts/W/m^2-ster-µm; array contains daily values over a given CPF interval Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: BAND_AVERAGE_GAINS	Band_4_Average_Gain	Dynamic	float32 array of flexible length	Band 4 detector-averaged gain in counts/W/m^2-ster-µm; array contains daily values over a given CPF interval Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: BAND_AVERAGE_GAINS	Band_5_Average_Gain	Dynamic	float32 array of flexible length	Band 5 detector-averaged gain in counts/W/m^2-ster-µm.; array contains daily values over a given CPF interval Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: BAND_AVERAGE_GAINS	Band_6_Average_Gain	Dynamic	float32 array of flexible length	Band 6 detector-averaged gain in counts/W/m^2-ster-µm; array contains daily values over a given CPF interval Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: BAND_AVERAGE_GAINS	Band_7_Average_Gain	Dynamic	float32 array of flexible length	Band 7 detector-averaged gain in counts/W/m^2-ster-µm; array contains daily values over a given CPF interval Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: BAND_AVERAGE_GAINS	Band_5_Average_Gain_ No_OG_Cor	Dynamic	float32 array of flexible length	Band 5 detector-averaged gain without applied correction for the outgassing effects, in counts/W/m^2-ster-µm; array contains daily values over a given CPF interval Valid format: NNN.NNNN, where N = 0 to 9

Parameter Groups	**Parameter Name**	**Value Type**	**Data Type**	**Description**
GROUP: DETECTOR_GAINS GROUP: BAND_AVERAGE_GAINS	Band_7_Average_Gain_No_OG_Cor	Dynamic	float32 array of flexible length	Band 7 detector-averaged gain without applied correction for the outgassing effects, in counts/W/m^2-ster-μm; array contains daily values over a given CPF interval Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: BAND_AVERAGE_GAINS	Prelaunch_Average_Gains	Static	Float32 array (7 values)	Prelaunch average detector gain in counts/W/m^2-ster-μm; array contains one value per spectral band Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: PRELAUNCH_GAINS	Band_1_Prelaunch_Gain	Static	float32 array (16 values)	Band 1 prelaunch detector gains in counts/W/m^2-ster-μm; array contains one value per detector Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: PRELAUNCH_GAINS	Band_2_Prelaunch_Gain	Static	float32 array (16 values)	Band 2 prelaunch detector gains in counts/W/m^2-ster-μm; array contains one value per detector Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: PRELAUNCH_GAINS	Band_3_Prelaunch_Gain	Static	float32 array (16 values)	Band 3 prelaunch detector gains in counts/W/m^2-ster-μm; array contains one value per detector Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: PRELAUNCH_GAINS	Band_4_Prelaunch_Gain	Static	float32 array (16 values)	Band 4 prelaunch detector gains in counts/W/m^2-ster-μm; array contains one value per detector Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: PRELAUNCH_GAINS	Band_5_Prelaunch_Gain	Static	float32 array (16 values)	Band 5 prelaunch detector gains in counts/W/m^2-ster-μm; array contains one value per detector Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: PRELAUNCH_GAINS	Band_6_Prelaunch_Gain	Static	float32 array (16 values)	Band 6 prelaunch detector gains in counts/W/m^2-ster-μm; array contains one value per detector Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: PRELAUNCH_GAINS	Band_7_Prelaunch_Gain	Static	float32 array (16 values)	Band 7 prelaunch detector gains in counts/W/m^2-ster-μm; array contains one value per detector Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: PRELAUNCH_GAINS	Bandwidth	Static	Float32 array (7 values)	Spectral bandwidth in μm, used to calculate the above prelaunch gains; array contains one value per spectral band Valid format: N.NNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_GAIN_PARAMETERS	Band_1_Relative_Gain_Slope	Dynamic	float32 array (16 values)	Band 1 relative gain linear model slopes; array contains one value for each detector Valid format: SNN.NNNNNNESN, where S = + or –, N = 0 to 9, and E = E
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_GAIN_PARAMETERS	Band_2_Relative_Gain_Slope	Dynamic	float32 array (16 values)	Band 2 relative gain linear model slopes; array contains one value for each detector Valid format: SNN.NNNNNNESN, where S = + or –, N = 0 to 9, and E = E
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_GAIN_PARAMETERS	Band_3_Relative_Gain_Slope	Dynamic	float32 array (16 values)	Band 3 relative gain linear model slopes; array contains one value for each detector Valid format: SNN.NNNNNNESN, where S = + or –, N = 0 to 9, and E = E
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_GAIN_PARAMETERS	Band_4_Relative_Gain_Slope	Dynamic	float32 array (16 values)	Band 4 relative gain linear model slopes; array contains one value for each detector Valid format: SNN.NNNNNNESN, where S = + or –, N = 0 to 9, and E = E
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_GAIN_PARAMETERS	Band_5_Relative_Gain_Slope	Dynamic	float32 array (16 values)	Band 5 relative gain linear model slopes; array contains one value for each detector Valid format: SNN.NNNNNNESN, where S = + or –, N = 0 to 9, and E = E

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_6_Relative_Gain_ Slope	Dynamic	float32 array (4 values)	Band 6 relative gain linear model slopes; array contains one value for each detector Valid format: SNN.NNNNNNESN, where S = + or –, N = 0 to 9, and E = E
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_7_Relative_Gain_ Slope	Dynamic	float32 array (16 values)	Band 7 relative gain linear model slopes; array contains one value for each detector Valid format: SNN.NNNNNNESN, where S = + or –, N = 0 to 9, and E = E
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_1_Relative_Gain_ Intercept	Dynamic	float32 array (16 values)	Band 1 relative gain linear model intercepts; array contains one value for each detector Valid format: N.NNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_2_Relative_Gain_ Intercept	Dynamic	float32 array (16 values)	Band 2 relative gain linear model intercepts; array contains one value for each detector Valid format: N.NNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_3_Relative_Gain_ Intercept	Dynamic	float32 array (16 values)	Band 3 relative gain linear model intercepts; array contains one value for each detector Valid format: N.NNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_4_Relative_Gain_ Intercept	Dynamic	float32 array (16 values)	Band 4 relative gain linear model intercepts; array contains one value for each detector Valid format: N.NNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_5_Relative_Gain_ Intercept	Dynamic	float32 array (16 values)	Band 5 relative gain linear model intercepts; array contains one value for each detector Valid format: N.NNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_6_Relative_Gain_ Intercept	Dynamic	float32 array (4 values)	Band 6 relative gain linear model intercepts; array contains one value for each detector Valid format: N.NNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_7_Relative_Gain_ Intercept	Dynamic	float32 array (16 values)	Band 7 relative gain linear model intercepts; array contains one value for each detector Valid format: N.NNNNNN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_1_RG_ExpPar1	Dynamic	float32 array (16 values)	Reserved Band 1 relative gain exponential model coefficient 1; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_2_RG_ExpPar1	Dynamic	float32 array (16 values)	Reserved Band 2 relative gain exponential model coefficient 1; array contains one value for each detector Valid format: N.NN, , where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_3_RG_ExpPar1	Dynamic	float32 array (16 values)	Reserved Band 3 relative gain exponential model coefficient 1; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_4_RG_ExpPar1	Dynamic	float32 array (16 values)	Reserved Band 4 relative gain exponential model coefficient 1; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_5_RG_ExpPar1	Dynamic	float32 array (16 values)	Reserved Band 5 relative gain exponential model coefficient 1; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_6_RG_ExpPar1	Dynamic	float32 array (4 values)	Reserved Band 6 relative gain exponential model coefficient 1; array contains one value for each detector Valid format: N.NN, where N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_7_RG_ExpPar1	Dynamic	float32 array (16 values)	Reserved Band 7 relative gain exponential model coefficient 2; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_1_RG_ExpPar2	Dynamic	float32 array (16 values)	Reserved Band 1 relative gain exponential model coefficient 2; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_2_RG_ExpPar2	Dynamic	float32 array (16 values)	Reserved Band 2 relative gain exponential model coefficient 2; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_3_RG_ExpPar2	Dynamic	float32 array (16 values)	Reserved Band 3 relative gain exponential model coefficient 2; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_4_RG_ExpPar2	Dynamic	float32 array (16 values)	Reserved Band 4 relative gain exponential model coefficient 2; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_5_RG_ExpPar2	Dynamic	float32 array (16 values)	Reserved Band 5 relative gain exponential model coefficient 2; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_6_RG_ExpPar2	Dynamic	float32 array (4 values)	Reserved Band 6 relative gain exponential model coefficient 2; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_7_RG_ExpPar2	Dynamic	float32 array (16 values)	Reserved Band 7 relative gain exponential model coefficient 2; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_1_RG_AddPar1	Dynamic	float32 array (16 values)	Reserved additional Band 1 relative gain model coefficient 1; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_2_RG_AddPar1	Dynamic	float32 array (16 values)	Reserved additional Band 2 relative gain model coefficient 1; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_3_RG_AddPar1	Dynamic	float32 array (16 values)	Reserved additional Band 3 relative gain model parameter 1; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_4_RG_AddPar1	Dynamic	float32 array (16 values)	Reserved additional Band 4 relative gain model parameter 1; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_5_RG_AddPar1	Dynamic	float32 array (16 values)	Reserved additional Band 5 relative gain model parameter 1; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_6_RG_AddPar1	Dynamic	float32 array (4 values)	Reserved additional Band 6 relative gain model parameter 1; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_7_RG_AddPar1	Dynamic	float32 array (16 values)	Reserved additional Band 7 relative gain model parameter Automated Cloud Cover Assessment (ACCA); array contains one value for each detector Valid format: N.NN, where N = 0 to 9

Parameter Groups	**Parameter Name**	**Value Type**	**Data Type**	**Description**
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_1_RG_AddPar2	Dynamic	float32 array (16 values)	Reserved additional Band 1 relative gain model coefficient 2; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_2_RG_AddPar2	Dynamic	float32 array (16 values)	Reserved additional Band 2 relative gain model coefficient 2; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_3_RG_AddPar2	Dynamic	float32 array (16 values)	Reserved additional Band 3 relative gain model parameter 2; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_4_RG_AddPar2	Dynamic	float32 array (16 values)	Reserved additional Band 4 relative gain model parameter 2; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_5_RG_AddPar2	Dynamic	float32 array (16 values)	Reserved additional Band 5 relative gain model parameter 2; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_6_RG_AddPar2	Dynamic	float32 array (4 values)	Reserved additional Band 6 relative gain model parameter 2; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: DETECTOR_GAINS GROUP: DETECTOR_RELATIVE_ GAIN_PARAMETERS	Band_7_RG_AddPar2	Dynamic	float32 array (16 values)	Reserved additional Band 7 relative gain model parameter 2; array contains one value for each detector Valid format: N.NN, where N = 0 to 9
GROUP: BIAS_LOCATIONS	Forward_Bias_ Location_30	Dynamic	int16	Offset, per line, in pixels, from the beginning of the data (left-hand offset) to the bias location starting point (start of DC restore) for Bands 1-5 and 7 Valid format: NNN, where NNN = to be supplied (TBS)
GROUP: BIAS_LOCATIONS	Forward_Bias_ Length_30	Dynamic	int16	Number of pixels to use, per line, in calculating bias for Bands 1-5 and 7 Valid format: NNN, where NNN = TBS
GROUP: BIAS_LOCATIONS	Forward_IC_ Region_30	Dynamic	int16	Length of useable Internal Calibrator (IC) region, in pixels, from the start of the bias region (DC restore) to the end of the calibration pulse region for Bands 1-5 and 7 Valid format: NNN, where NNN = TBS
GROUP: BIAS_LOCATIONS	Reverse_Bias_ Location_30	Dynamic	int16	Offset, per line, in pixels, from the beginning of the data (right-hand offset) to the bias location starting point (start of DC restore) for Bands 1-5 and 7 Valid format: NNN, where NNN = TBS
GROUP: BIAS_LOCATIONS	Reverse_Bias_ Length_30	Dynamic	int16	Number of pixels to use per line in calculating the bias for Bands 1-5 and 7 Valid format: NNN, where NNN = TBS
GROUP: BIAS_LOCATIONS	Reverse_IC_ Region_30	Dynamic	int16	Length of useable IC region, in pixels, from the start of the bias region (DC restore) to the end of the calibration pulse region for Bands 1-5 and 7 Valid format: NNN, where NNN = TBS
GROUP: BIAS_LOCATIONS	Forward_Bias_ Location_120	Dynamic	int16	Offset, per-line, in pixels, from the beginning of the data (left-hand offset) to the bias location starting point (start of DC restore) for Band 6 Valid format: NNN, where NNN = TBS
GROUP: BIAS_LOCATIONS	Forward_Bias_ Length_120	Dynamic	int16	Number of pixels to use per line, in calculating the bias for Band 6 Valid format: NNN, where NNN = TBS

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: BIAS_LOCATIONS	Forward_IC_Region_120	Dynamic	int16	Length of the useable IC region, in pixels, from the start of the bias region (DC restore) to the end of the calibration pulse region for Band 6 Valid format: NNN, where NNN = TBS
GROUP: BIAS_LOCATIONS	Reverse_Bias_Location_120	Dynamic	int16	Offset, per line, in pixels, from the beginning of the data (right-hand offset) to the bias location starting point (start of DC restore) for Band 6 Valid format: NNN, where NNN = TBS
GROUP: BIAS_LOCATIONS	Reverse_Bias_Length_120	Dynamic	int16	Number of pixels to use, per line, in calculating the bias for Band 6 Valid format: NNN, where NNN = TBS
GROUP: BIAS_LOCATIONS	Reverse_IC_Region_120	Dynamic	int16	Length of useable IC region, in pixels, from the start of the bias region (DC restore) to the end of the calibration pulse region for Band 6 Valid format: NNN, where NNN = TBS
GROUP: DETECTOR_BIASES	Band_1_Detector_Bias	Static	float32 array (16 values)	Band 1 prelaunch average on-orbit detector bias in digital counts; array contains one value per detector Valid format: N.N, where N = 0 to 9
GROUP: DETECTOR_BIASES	Band_2_Detector_Bias	Static	float32 array (16 values)	Band 2 average on-orbit prelaunch detector bias in digital counts; array contains one value per detector Valid format: N.N, where N = 0 to 9
GROUP: DETECTOR_BIASES	Band_3_Detector_Bias	Static	float32 array (16 values)	Band 3 average on-orbit prelaunch detector bias in digital counts; array contains one value per detector Valid format: N.N, where N = 0 to 9
GROUP: DETECTOR_BIASES	Band_4_Detector_Bias	Static	float32 array (16 values)	Band 4 average on-orbit prelaunch detector bias in digital counts; array contains one value per detector Valid format: N.N, where N = 0 to 9
GROUP: DETECTOR_BIASES	Band_5_Detector_Bias	Static	float32 array (16 values)	Band 5 average on-orbit prelaunch detector bias in digital counts; array contains one value per detector Valid format: N.N, where N = 0 to 9
GROUP: DETECTOR_BIASES	Band_6_Detector_Bias	Static	float32 array (4 values)	Band 6 average on-orbit prelaunch detector bias in digital counts; array contains one value per detector Valid format: N.N, where N = 0 to 9
GROUP: DETECTOR_BIASES	Band_7_Detector_Bias	Static	float32 array (16 values)	Band 7 average on-orbit prelaunch detector bias in digital counts; array contains one value per detector Valid format: N.N, where N = 0 to 9
GROUP: DETECTOR_BIASES GROUP: BIAS_LIMITS	Band_1_Lower_Limit	Static	float32 array (16 values)	Valid lower limit for bias in digital counts for Band 1; array contains one value per detector Valid format: N.N, where N.N = 0.5
GROUP: DETECTOR_BIASES GROUP: BIAS_LIMITS	Band_2_Lower_Limit	Static	float32 array (16 values)	Valid lower limit for bias in digital counts for Band 2; array contains one value per detector Valid format: N.N, where N.N = 0.5
GROUP: DETECTOR_BIASES GROUP: BIAS_LIMITS	Band_3_Lower_Limit	Static	float32 array (16 values)	Valid lower limit for bias in digital counts for Band 3; array contains one value per detector Valid format: N.N, where N.N = 0.5
GROUP: DETECTOR_BIASES GROUP: BIAS_LIMITS	Band_4_Lower_Limit	Static	float32 array (16 values)	Valid lower limit for bias in digital counts for Band 4; array contains one value per detector Valid format: N.N, where N.N = 0.5
GROUP: DETECTOR_BIASES GROUP: BIAS_LIMITS	Band_5_Lower_Limit	Static	float32 array (16 values)	Valid lower limit for bias in digital counts for Band 5; array contains one value per detector Valid format: N.N, where N.N = 0.5

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: DETECTOR_BIASES GROUP: BIAS_LIMITS	Band_7_Lower_Limit	Static	float32 array (16 values)	Valid lower limit for bias in digital counts for Band 7; array contains one value per detector Valid format: N.N, where N.N = 0.5
GROUP: DETECTOR_BIASES GROUP: BIAS_LIMITS	Band_1_Lower_Limit	Static	float32 array (16 values)	Valid upper limit for bias in digital counts for Band 1; array contains one value per detector Valid format: N.N, where N.N = 6.0
GROUP: DETECTOR_BIASES GROUP: BIAS_LIMITS	Band_2_Upper_Limit	Static	float32 array (16 values)	Valid upper limit for bias in digital counts for Band 2; array contains one value per detector Valid format: N.N, where N.N = 6.0
GROUP: DETECTOR_BIASES GROUP: BIAS_LIMITS	Band_3_Upper_Limit	Static	float32 array (16 values)	Valid upper limit for bias in digital counts for Band 3; array contains one value per detector Valid format: N.N, where N.N = 6.0
GROUP: DETECTOR_BIASES GROUP: BIAS_LIMITS	Band_4_Upper_Limit	Static	float32 array (16 values)	Valid upper limit for bias in digital counts for Band 4; array contains one value per detector Valid format: N.N, where N.N = 6.0
GROUP: DETECTOR_BIASES GROUP: BIAS_LIMITS	Band_5_Upper_Limit	Static	float32 array (16 values)	Valid upper limit for bias in digital counts for Band 5; array contains one value per detector Valid format: N.N, where N.N = 3.5
GROUP: DETECTOR_BIASES GROUP: BIAS_LIMITS	Band_7_Upper_Limit	Static	float32 array (16 values)	Valid upper limit for bias in digital counts for Band 7; array contains one value per detector Valid Format: N.N, where N.N = 3.5
GROUP: DETECTOR_BIASES GROUP: PRELAUNCH_BIASES	Band_1_Prelaunch_Bias	Static	float32 array (16 values)	Band 1 prelaunch detector bias in digital counts; array contains one value per detector Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_BIASES GROUP: PRELAUNCH_BIASES	Band_2_ Prelaunch _Bias	Static	float32 array (16 values)	Band 2 prelaunch detector bias in digital counts; array contains one value per detector Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_BIASES GROUP: PRELAUNCH_BIASES	Band_3_ Prelaunch _Bias	Static	float32 array (16 values)	Band 3 prelaunch detector bias in digital counts; array contains one value per detector Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_BIASES GROUP: PRELAUNCH_BIASES	Band_4_ Prelaunch _Bias	Static	float32 array (16 values)	Band 4 prelaunch detector bias in digital counts; array contains one value per detector Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_BIASES GROUP: PRELAUNCH_BIASES	Band_5_ Prelaunch _Bias	Static	float32 array (16 values)	Band 5 prelaunch detector bias in digital counts; array contains one value per detector Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_BIASES GROUP: PRELAUNCH_BIASES	Band_6_ Prelaunch _Bias	Static	float32 array (4 values)	Band 6 prelaunch detector bias in digital counts; array contains one value per detector Valid format: NNN.NNNN, where N = 0 to 9
GROUP: DETECTOR_BIASES GROUP: PRELAUNCH_BIASES	Band_7_ Prelaunch _Bias	Static	float32 array (16 values)	Band 7 prelaunch detector bias in digital counts; array contains one value per detector Valid format: NNN.NNNN, where N = 0 to 9
GROUP: ACCA_BIASES	B1_ACCA_BIAS	Dynamic	float32 array (16 values)	Band 1 ACCA bias in digital count; array contains one value per each detector Valid format: NN.NN, where NN.NN = TBS
GROUP: ACCA_BIASES	B2_ACCA_BIAS	Dynamic	float32 array (16 values)	Band 2 ACCA bias in digital counts; array contains one value per each detector Valid format: NN.NN, where NN.NN = TBS
GROUP: ACCA_BIASES	B3_ACCA_BIAS	Dynamic	float32 array (16 values)	Band 3 ACCA bias in digital counts; array contains one value per each detector Valid format: NN.NN, where NN.NN = TBS
GROUP: ACCA_BIASES	B4_ACCA_BIAS	Dynamic	float32 array (16 values)	Band 4 ACCA bias in digital counts; array contains one value per each detector Valid format: NN.NN, where NN.NN = TBS
GROUP: ACCA_BIASES	B5_ACCA_BIAS	Dynamic	float32 array (16 values)	Band 5 ACCA bias in digital counts; array contains one value per each detector Valid format: NN.NN, where NN.NN = TBS

Parameter Groups	**Parameter Name**	**Value Type**	**Data Type**	**Description**
GROUP: ACCA_BIASES	B6_ACCA_BIAS	Dynamic	float32 array (4 values)	Band 6 ACCA bias in digital counts; array contains one value per each detector Valid format: NN.NN, where NN.NN = TBS
GROUP: ACCA_BIASES	B7_ACCA_BIAS	Dynamic	float32 array (16 values)	Band 7 ACCA bias in digital counts; array contains one value per each detector Valid format: NN.NN, where NN.NN = TBS
GROUP: ACCA_THRESHOLDS	Thresh_B3	Dynamic	float32	Band 3 ACCA threshold Valid format: N.NNNN, where N.NNNN = TBS
GROUP: ACCA_THRESHOLDS	Thresh_B3_Lower	Dynamic	float32	Band 3 land reflectance threshold Valid format: NN.NN, where NN.NN = TBS
GROUP: ACCA_THRESHOLDS	Thresh_B56	Dynamic	float32	Band 5-6 composite threshold Valid format: NNN.NNN, where NNN.NNN = TBS
GROUP: ACCA_THRESHOLDS	Thresh_B6	Dynamic	float32	Band 6 threshold – maximum cloud temperature Valid format: NNN.NNN, where NNN.NNN = TBS
GROUP: ACCA_THRESHOLDS	Thresh_B45_Ratio	Dynamic	float32	Band 4-5 ratio threshold Valid format: N.NNNN, where N.NNNN = TBS
GROUP: ACCA_THRESHOLDS	Thresh_B42_Ratio	Dynamic	float32	Band 4-2 ratio threshold Valid format: N.NNNN, where N.NNNN = TBS
GROUP: ACCA_THRESHOLDS	Thresh_B43_Ratio	Dynamic	float32	Band 4-3 ratio threshold Valid format: N.NNNN, where N.NNNN = TBS
GROUP: ACCA_THRESHOLDS	Thresh_NDSI_Max	Dynamic	float32	Normalized Difference Snow Index (NDSI) ceiling Valid format: N.NNNN, where N.NNNN = TBS
GROUP: ACCA_THRESHOLDS	Thresh_NDSI_Min	Dynamic	float32	NDSI floor Valid format: N.NNNN, where N.NNNN = TBS
GROUP: ACCA_THRESHOLDS	Thresh_NDSI_Snow	Dynamic	float32	NDSI threshold used to identify snow Valid format: NN.NNNN, where NN.NNNN = TBS
GROUP: ACCA_THRESHOLDS	Cloud_Percent_Min	Dynamic	float32	Minimum cloud cover percentage required for pass two Valid format: N.NNNN, where N.NNNN = TBS
GROUP: ACCA_THRESHOLDS	Desert_Index	Dynamic	float32	Desert Index (Thresh_45_Ratio / Thresh_42_Ratio) Valid format: N.NNNN, where N.NNNN = TBS
GROUP: ACCA_THRESHOLDS	Thresh_Snow_Percent	Dynamic	float32	Maximum snow cover percentage allowed to use looser cloud properties for pass two Valid format: N.NNN, where N.NNN = TBS
GROUP: ACCA_THRESHOLDS	Thermal_Effect_High	Dynamic	float32	Maximum allowable pass 2 percentage cloud cover increase allowed using looser cloud properties Valid format: NNN.NNN, where NNN.NNN = TBS
GROUP: ACCA_THRESHOLDS	Thermal_Effect_Low	Dynamic	float32	Maximum allowable pass 2 percentage cloud cover increase allowed using narrower cloud properties Valid format: NNN.NNN, where NNN.NNN = TBS
GROUP: ACCA_THRESHOLDS	B6Max_Maxthresh_Diff	Dynamic	float32	Minimum difference allowed between maximum cloud temperature and maximum thermal threshold Valid format: NN.NNN, where NN.NNN = TBS
GROUP: SOLAR_SPECTRAL_ IRRADIANCES	B1_Solar_Irradiance	Static	float32	Mean solar exoatmospheric irradiance for Band 1 in W/m^2-ster-μm Valid format: NNNN.NNN, where NNNN.NNN = 1957.000

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: SOLAR_SPECTRAL_ IRRADIANCES	B2_Solar_Irradiance	Static	float32	Mean solar exoatmospheric irradiance for Band 2 in W/m^2-ster-μm Valid format: NNNN.NNN, where NNNN.NNN = 1826.000
GROUP: SOLAR_SPECTRAL_ IRRADIANCES	B3_Solar_Irradiance	Static	float32	Mean solar exoatmospheric irradiance for Band 3 in W/m^2-ster-μm Valid format: NNNN.NNN, where NNNN.NNN = 1554.000
GROUP: SOLAR_SPECTRAL_ IRRADIANCES	B4_Solar_Irradiance	Static	float32	Mean solar exoatmospheric irradiance for Band 4 in W/m^2-ster-μm Valid format: NNNN.NNN, where NNNN.NNN = 1036.000
GROUP: SOLAR_SPECTRAL_ IRRADIANCES	B5_Solar_Irradiance	Static	float32	Mean solar exoatmospheric irradiance for Band 5 in W/m^2-ster-μm Valid format: NNNN.NNN, where NNNN.NNN = 215.000
GROUP: SOLAR_SPECTRAL_ IRRADIANCES	B7_Solar_Irradiance	Static	float32	Mean solar exoatmospheric irradiance for Band 7 in W/m^2-ster-μm Valid format: NNNN.NNN, where NNNN.NNN = 80.670
GROUP: BAND_6_CALIBRATION_ COEFFICIENTS	Temp_To_Rad	Static	float64 array (3 values)	Coefficients used to extract the effective spectral radiance of the calibration shutter and the blackbody, expressed in W/m^{2}srμm Valid format: N.NNNNESN, where S = + or –, N = 0 to 9, and E = E
GROUP: BAND_6_CALIBRATION_ COEFFICIENTS	a	Static	float32 array (4 values)	Constants used to estimate the thermal detector gains Valid format: N.NN, where N = 0 to 9
GROUP: BAND_6_CALIBRATION_COE FFICIENTS	b	Static	float32 array (4 values)	Constants used to estimate the thermal detector biases Valid format: N.NNN, where N = 0 to 9
GROUP: BAND_6_CALIBRATION_ COEFFICIENTS	c	Static	float32 array (4 values)	Constants used to estimate the thermal detector biases Valid format: N.NNN, where N = 0 to 9
GROUP: THERMAL_CONSTANTS	K1_Constant	Static	float32	Thermal calibration constant 1 in W/m^2-ster-μm Valid format: NNNNN.NNN, where NNNNN.NNN = 607.760
GROUP: THERMAL_CONSTANTS	K2_Constant	Static	float32	Thermal calibration constant 2 in degrees Kelvin Valid format: NNNNN.NNN, where NNNNN.NNN = 1260.560
GROUP: SCALING_PARAMETERS	B1_Lmin_Lmax	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors for Band 1, in W/m^2-ster-μm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SCALING_PARAMETERS	B2_Lmin_Lmax	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors for Band 2, in W/m^2-ster-μm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SCALING_PARAMETERS	B3_Lmin_Lmax	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors for Band 3, in W/m^2-ster-μm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SCALING_PARAMETERS	B4_Lmin_Lmax	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors for Band 4, in W/m^2-ster-μm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SCALING_PARAMETERS	B5_Lmin_Lmax	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors for Band 5, in W/m^2-ster-μm Valid format: SNNN.NNNN, where S = + or –. and N = 0 to 9

Parameter Groups	**Parameter Name**	**Value Type**	**Data Type**	**Description**
GROUP: SCALING_PARAMETERS	B6_Lmin_Lmax	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors for Band 6, in W/m^2-ster-µm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SCALING_PARAMETERS	B7_Lmin_Lmax	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors for Band 7, in W/m^2-ster-µm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SCALING_PARAMETERS	B1_Lmin_Lmax_LUT03	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors used in radiometric processing based on the application of the LUT 03 gain model for Band 1, in W/m^2-ster-µm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SCALING_PARAMETERS	B2_Lmin_Lmax_LUT03	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors used in radiometric processing based on the application of the LUT 03 gain model for Band 2, in W/m^2-ster-µm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SCALING_PARAMETERS	B3_Lmin_Lmax_LUT03	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors used in radiometric processing based on the application of the LUT 03 gain model for Band 3, in W/m^2-ster-µm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SCALING_PARAMETERS	B4_Lmin_Lmax_LUT03	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors used in radiometric processing based on the application of the LUT 03 gain model for Band 4, in W/m^2-ster-µm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SCALING_PARAMETERS	B5_Lmin_Lmax_LUT03	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors used in radiometric processing based on the application of the LUT 03 gain model for Band 5, in W/m^2-ster-µm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SCALING_PARAMETERS	B6_Lmin_Lmax_LUT03	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors used in radiometric processing based on the application of the LUT 03 gain model for Band 6, in W/m^2-ster-µm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SCALING_PARAMETERS	B7_Lmin_Lmax_LUT03	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors used in radiometric processing based on the application of the LUT 03 gain model for Band 7, in W/m^2-ster-µm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SCALING_PARAMETERS	B1_Lmin_Lmax_IC	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors used in processing based on the Internal Calibrator gains for Band 1, in W/m^2-ster-µm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SCALING_PARAMETERS	B2_Lmin_Lmax_IC	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors used in processing based on the Internal Calibrator gains for Band 2, in W/m^2-ster-µm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: SCALING_PARAMETERS	B3_Lmin_Lmax_IC	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors used in processing based on the Internal Calibrator gains for Band 3, in W/m^2-ster-µm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SCALING_PARAMETERS	B4_Lmin_Lmax_IC	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors used in processing based on the Internal Calibrator gains for Band 4, in W/m^2-ster-µm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SCALING_PARAMETERS	B5_Lmin_Lmax_IC	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors used in processing based on the Internal Calibrator gains for Band 5, in W/m^2-ster-µm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SCALING_PARAMETERS	B6_Lmin_Lmax_IC	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors used in processing based on the Internal Calibrator gains for Band 6, in W/m^2-ster-µm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SCALING_PARAMETERS	B7_Lmin_Lmax_IC	Static	float32 array (2 values)	Postcalibration 8-bit dynamic range scaling factors used in processing based on the Internal Calibrator gains for Band 7, in W/m^2-ster-µm Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: MTF_COMPENSATION	B1_weights_along	Dynamic	float64 array (5 values)	Weighting function coefficients used to compute along-scan Modulation Transfer Function Compensation (MTFC) for Band 1 Valid format: SNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: MTF_COMPENSATION	B1_weights_across	Dynamic	float64 array (5 values)	Weighting function coefficients used to compute across-scan MTFC for Band 1 Valid format: SNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: MTF_COMPENSATION	B2_weights_along	Dynamic	float64 array (5 values)	Weighting function coefficients used to compute along-scan MTFC for Band 2 Valid format: SNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: MTF_COMPENSATION	B2_weights_across	Dynamic	float64 array (5 values)	Weighting function coefficients used to compute across-scan MTFC for Band 2 Valid format: SNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: MTF_COMPENSATION	B3_weights_along	Dynamic	float64 array (5 values)	Weighting function coefficients used to compute along-scan MTFC for Band 3 Valid format: SNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: MTF_COMPENSATION	B3_weights_across	Dynamic	float64 array (5 values)	Weighting function coefficients used to compute across-scan MTFC for Band 3 Valid format: SNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: MTF_COMPENSATION	B4_weights_along	Dynamic	float64 array (5 values)	Weighting function coefficients used to compute along-scan MTFC for Band 4 Valid format: SNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: MTF_COMPENSATION	B4_weights_across	Dynamic	float64 array (5 values)	Weighting function coefficients used to compute across-scan MTFC for Band 4 Valid format: SNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: MTF_COMPENSATION	B5_weights_along	Dynamic	float64 array (5 values)	Weighting function coefficients used to compute along-scan MTFC for Band 5 Valid format: SNN.NNNN, where S = + or –, and N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: MTF_COMPENSATION	B5_weights_across	Dynamic	float64 array (5 values)	Weighting function coefficients used to compute across-scan MTFC for Band 5 Valid format: SNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: MTF_COMPENSATION	B6_weights_along	Dynamic	float64 array (5 values)	Weighting function coefficients used to compute along-scan MTFC for Band 6 Valid format: SNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: MTF_COMPENSATION	B6_weights_across	Dynamic	float64 array (5 values)	Weighting function coefficients used to compute across-scan MTFC for Band 6 Valid format: SNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: MTF_COMPENSATION	B7_weights_along	Dynamic	float64 array (5 values)	Weighting function coefficients used to compute along-scan MTFC for Band 7 Valid format: SNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: MTF_COMPENSATION	B7_weights_across	Dynamic	float64 array (5 values)	Weighting function coefficients used to compute across-scan MTFC for Band 7 Valid format: SNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: MEMORY_EFFECT GROUP: ME_MAGNITUDES	B1_kME_Magnitude	Dynamic	Float64 array (16 values)	Band 1 memory effect magnitude measured in Digital Numbers (DNs); array contains one value per detector Valid format: SN.NNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: MEMORY_EFFECT GROUP: ME_MAGNITUDES	B2_kME_Magnitude	Dynamic	float64 array (16 values)	Band 2 memory effect magnitude measured in DNs; array contains one value per detector Valid format: SN.NNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: MEMORY_EFFECT GROUP: ME_MAGNITUDES	B3_kME_Magnitude	Dynamic	float64 array (16 values)	Band 3 memory effect magnitude measured in DNs; array contains one value per detector Valid format: SN.NNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: MEMORY_EFFECT GROUP: ME_MAGNITUDES	B4_kME_Magnitude	Dynamic	float64 array (16 values)	Band 4 memory effect magnitude measured in DNs; array contains one value per detector Valid format: SN.NNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: MEMORY_EFFECT GROUP: ME_MAGNITUDES	B5_kME_Magnitude	Dynamic	float64 array (16 values)	Band 5 memory effect magnitude measured in DNs; array contains one value per detector Valid format: SN.NNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: MEMORY_EFFECT GROUP: ME_MAGNITUDES	B6_kME_Magnitude	Dynamic	float64 array (4 values)	Band 6 memory effect magnitude measured in DNs; array contains one value per detector Valid format: SN.NNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: MEMORY_EFFECT GROUP: ME_MAGNITUDES	B7_kME_Magnitude	Dynamic	float64 array (16 values)	Band 7 memory effect magnitude measured in DNs; array contains one value per detector Valid format: SN.NNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: MEMORY_EFFECT GROUP: ME_SCALING	B1_ME_Scal_Factor	Dynamic	Float32 array (16 values)	Band 1 memory effect magnitude scaling factor; array contains one value per detector Valid format: N.N, where N = 0 to 9
GROUP: MEMORY_EFFECT GROUP: ME_SCALING	B2_ME_Scal_Factor	Dynamic	Float32 array (16 values)	Band 2 memory effect magnitude scaling factor; array contains one value per detector Valid format: N.N, where N = 0 to 9
GROUP: MEMORY_EFFECT GROUP: ME_SCALING	B3_ME_Scal_Factor	Dynamic	Float32 array (16 values)	Band 3 memory effect magnitude scaling factor; array contains one value per detector Valid format: N.N, where N = 0 to 9
GROUP: MEMORY_EFFECT GROUP: ME_SCALING	B4_ME_Scal_Factor	Dynamic	Float32 array (16 values)	Band 4 memory effect magnitude scaling factor; array contains one value per detector Valid format: N.N, where N = 0 to 9

Parameter Groups	**Parameter Name**	**Value Type**	**Data Type**	**Description**
GROUP: MEMORY_EFFECT GROUP: ME_SCALING	B5_ME_Scal_Factor	Dynamic	Float32 array (16 values)	Band 5 memory effect magnitude scaling factor; array contains one value per detector Valid format: N.N, where N = 0 to 9
GROUP: MEMORY_EFFECT GROUP: ME_SCALING	B6_ME_Scal_Factor	Dynamic	Float32 array (4 values)	Band 6 memory effect magnitude scaling factor; array contains one value per detector Valid format: N.N, where N = 0 to 9.
GROUP: MEMORY_EFFECT GROUP: ME_SCALING	B7_ME_Scal_Factor	Dynamic	Float32 array (16 values)	Band 7 memory effect magnitude scaling factor; array contains one value per detector Valid format: N.N, where N = 0 to 9
GROUP: MEMORY_EFFECT GROUP: ME_TIME_CONSTANTS	B1_ME_Time_Constant	Dynamic	float32 array (16 values)	Band 1 time constant measured in minor frames; array contains one value per detector Valid format: NNNN.NN, where N = 0 to 9
GROUP: ME_TIME_CONSTANTS	B2_ME_Time_Constant	Dynamic	float32 array (16 values)	Band 2 time constant measured in minor frames; array contains one value per detector Valid format: NNNN.NN, where N = 0 to 9
GROUP: MEMORY_EFFECT GROUP: ME_TIME_CONSTANTS	B3_ME_Time_Constant	Dynamic	float32 array (16 values)	Band 3 time constant measured in minor frames; array contains one value per detector Valid format: NNNN.NN, where N = 0 to 9
GROUP: MEMORY_EFFECT GROUP: ME_TIME_CONSTANTS	B4_ME_Time_Constant	Dynamic	float32 array (16 values)	Band 4 time constant measured in minor frames; array contains one value per detector Valid format: NNNN.NN, where N = 0 to 9
GROUP: MEMORY_EFFECT GROUP: ME_TIME_CONSTANTS	B5_ME_Time_Constant	Dynamic	float32 array (16 values)	Band 5 time constant measured in minor frames; array contains one value per detector Valid format: NNNN.NN, where N = 0 to 9
GROUP: MEMORY_EFFECT GROUP: ME_TIME_CONSTANTS	B6_ME_Time_Constant	Dynamic	float32 array (4 values)	Band 6 time constant measured in minor frames; array contains one value per detector Valid format: NNNN.NN, where N = 0 to 9
GROUP: MEMORY_EFFECT GROUP: ME_TIME_CONSTANTS	B7_ME_Time_Constant	Dynamic	float32 array (16 values)	Band 7 time constant measured in minor frames; array contains one value per detector Valid format: NNNN.NN, where N = 0 to 9
GROUP: MEMORY_EFFECT GROUP: ME_FILTER_PARAMETERS	ME_Filter_Widths	Dynamic	int16 array (30 values)	Convolution 30-step filter widths Valid format: NNN, where N = 0 to 9
GROUP: GHOST_PULSE	Ghost_Pulse_ Endpoints	Dynamic	float32 array (2 values)	Beginning and ending fractional minor frames that bound IC ghost pulse Valid format: NNNN.NNNN, where N = 0 to 9
GROUP: SCAN_CORRELATED_SHIFT	SCS_Reference_Detector _1	Dynamic	uint8 array (3 values)	Default scan correlated shift reference detector; array contains band number, detector number, and phase relative to Det. 1 in Band 2 (1-in phase, 0-out of phase), respectively, identifying a single detector used for correction in all bands Valid format: NN, where N = 0 to 9
GROUP: SCAN_CORRELATED_SHIFT	SCS_Reference_Detector _2	Dynamic	uint8 array (3 values)	Optional Scan-Correlated Shift (SCS) reference detector; array contains band number, detector number, and phase relative to Det. 1 in Band 2 (1-in phase, 0-out of phase), respectively, identifying a single detector used for correction in all bands Valid format: NN, where N = 0 to 9
GROUP: SCAN_CORRELATED_SHIFT	SCS_Reference_Detector _3	Dynamic	uint8 array (3 values)	Optional SCS reference detector; array contains band number, detector number, and phase relative to Det. 1 in Band 2 (1-in phase, 0-out of phase), respectively, identifying a single detector used for correction in all bands Valid format: NN, where N = 0 to 9

Parameter Groups	**Parameter Name**	**Value Type**	**Data Type**	**Description**
GROUP: SCAN_CORRELATED_SHIFT	SCS_State_Mask_ Parameters	Dynamic	float32 array (6 values)	SCS state mask parameters Valid format: (A, B, C, D, E), where A = slope of the linear model for the bias state B = day-since-launch when 27 scenes of the 1985 night data set were acquired C = intercept of the linear model for the bias state D = high delta value used to determine the upper limit for detector bias levels E = lower delta value used to determine the upper limit for detector bias levels
GROUP: SCAN_CORRELATED_SHIFT	B1_SCS_Additive_ Correction_Factors	Dynamic	float64 array (16 values)	Magnitude of Band 1 shift in digital numbers; array contains one value per detector Valid format: SN.NNNNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: SCAN_CORRELATED_SHIFT	B2_SCS_Additive_ Correction_Factors	Dynamic	float64 array (16 values)	Magnitude of Band 2 shift in digital numbers; array contains one value per detector Valid format: SN.NNNNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: SCAN_CORRELATED_SHIFT	B3_SCS_Additive_ Correction_Factors	Dynamic	float64 array (16 values)	Magnitude of Band 3 shift in digital numbers; array contains one value per detector Valid format: SN.NNNNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: SCAN_CORRELATED_SHIFT	B4_SCS_Additive_ Correction_Factors	Dynamic	float64 array (16 values)	Magnitude of Band 4 shift in digital numbers; array contains one value per detector Valid format: SN.NNNNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: SCAN_CORRELATED_SHIFT	B5_SCS_Additive_ Correction_Factors	Dynamic	float64 array (16 values)	Magnitude of Band 5 shift in digital numbers; array contains one value per detector Valid format: SN.NNNNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: SCAN_CORRELATED_SHIFT	B6_SCS_Additive_ Correction_Factors	Dynamic	float64 array (4 values)	Magnitude of Band 6 shift in digital numbers; array contains one value per detector Valid format: SN.NNNNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: SCAN_CORRELATED_SHIFT	B7_SCS_Additive_ Correction_Factors	Dynamic	float64 array (16 values)	Magnitude of Band 7 shift in digital numbers; array contains one value per detector Valid format: SN.NNNNNNNNESNN, where S = + or –, N = 0 to 9, and E = E
GROUP: STRIPING	Correction_ Reference_B1	Static	uint8	Striping correction methodology flag, relative to the band average or reference detector, for Band 1 Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction)
GROUP: STRIPING	Correction_ Reference_B2	Static	uint8	Striping correction methodology flag, relative to the band average or reference detector, for Band 2 Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction)
GROUP: STRIPING	Correction_ Reference_B3	Static	uint8	Striping correction methodology flag, relative to the band average or reference detector, for Band 3 Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction)
GROUP: STRIPING	Correction_ Reference_B4	Static	uint8	Striping correction methodology flag, relative to the band average or reference detector, for Band 4 Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction)

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: STRIPING	Correction_ Reference_B5	Static	uint8	Striping correction methodology flag, relative to the band average or reference detector, for Band 5 Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction)
GROUP: STRIPING	Correction_ Reference_B6	Static	uint8	Striping correction methodology flag, relative to the band average or reference detector, for Band 6 Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction)
GROUP: STRIPING	Correction_ Reference_B7	Static	uint8	Striping correction methodology flag, relative to the band average or reference detector, for Band 7 Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction)
GROUP: HISTOGRAM GROUP: DETECTOR_NOISE	Detector_Noise_ Level_B1	Dynamic	float32 array (16 values)	Standard deviation of the image region data for each detector of Band 1 Valid format: NN.NNNN, where N = 0 to 9
GROUP: HISTOGRAM GROUP: DETECTOR_NOISE	Detector_Noise_ Level_B2	Dynamic	float32 array (16 values)	Standard deviation of the image region data for each detector of Band 2 Valid format: NN.NNNN, where N = 0 to 9
GROUP: HISTOGRAM GROUP: DETECTOR_NOISE	Detector_Noise_ Level_B3	Dynamic	float32 array (16 values)	Standard deviation of the image region data for each detector of Band 3 Valid format: NN.NNNN, where N = 0 to 9
GROUP: HISTOGRAM GROUP: DETECTOR_NOISE	Detector_Noise_ Level_B4	Dynamic	float32 array (16 values)	Standard deviation of the image region data for each detector of Band 4 Valid format: NN.NNNN, where N = 0 to 9
GROUP: HISTOGRAM GROUP: DETECTOR_NOISE	Detector_Noise_ Level_B5	Dynamic	float32 array (16 values)	Standard deviation of the image region data for each detector of Band 5 Valid format: NN.NNNN, where N = 0 to 9
GROUP: HISTOGRAM GROUP: DETECTOR_NOISE	Detector_Noise_ Level_B6	Dynamic	float32 array (4 values)	Standard deviation of the image region data for each detector of Band 6 Valid format: NN.NNNN, where N = 0 to 9
GROUP: HISTOGRAM GROUP: DETECTOR_NOISE	Detector_Noise_ Level_B7	Dynamic	float32 array (16 values)	Standard deviation of the image region data for each detector of Band 7 Valid format: NN.NNNN, where N = 0 to 9
GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE	Det_Shutter_Noise_ Level_B1	Dynamic	float32 array (16 values)	Standard deviation of the shutter region data for each detector of Band 1 Valid format: NN.NNNN, where N = 0 to 9
GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE	Det_Shutter_Noise_ Level_B2	Dynamic	float32 array (16 values)	Standard deviation of the shutter region data for each detector of Band 2 Valid format: NN.NNNN, where N = 0 to 9
GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE	Det_Shutter_Noise_ Level_B3	Dynamic	float32 array (16 values)	Standard deviation of the shutter region data for each detector of Band 3 Valid format: NN.NNNN, where N = 0 to 9
GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE	Det_Shutter_Noise_ Level_B4	Dynamic	float32 array (16 values)	Standard deviation of the shutter region data for each detector of Band 4 Valid format: NN.NNNN, where N = 0 to 9
GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE	Det_Shutter_Noise_ Level_B5	Dynamic	float32 array (16 values)	Standard deviation of the shutter region data for each detector of Band 5 Valid format: NN.NNNN, where N = 0 to 9
GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE	Det_Shutter_Noise_ Level_B6	Dynamic	float32 array (4 values)	Standard deviation of the shutter region data for each detector of Band 6 Valid format: NN.NNNN, where N = 0 to 9
GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE	Det_Shutter_Noise_ Level_B7	Dynamic	float32 array (16 values)	Standard deviation of the shutter region data for each detector of Band 7 Valid format: NN.NNNN, where N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: HISTOGRAM GROUP: REFERENCE_DETECTORS	Reference_Detector_B1	Dynamic	uint8	Detector used as a reference when computing relative detector gains and biases (least noisy), Band 1 Valid format: NN, where NN = TBS
GROUP: HISTOGRAM GROUP: REFERENCE_DETECTORS	Reference_Detector_B2	Dynamic	uint8	Detector used as a reference when computing relative detector gains and biases (least noisy), Band 2 Valid format: NN, where NN = TBS
GROUP: HISTOGRAM GROUP: REFERENCE_DETECTORS	Reference_Detector_B3	Dynamic	uint8	Detector used as a reference when computing relative detector gains and biases (least noisy), Band 3 Valid format: NN, where NN = TBS
GROUP: HISTOGRAM GROUP: REFERENCE_DETECTORS	Reference_Detector_B4	Dynamic	uint8	Detector used as a reference when computing relative detector gains and biases (least noisy), Band 4 Valid format: NN, where NN = TBS
GROUP: HISTOGRAM GROUP: REFERENCE_DETECTORS	Reference_Detector_B5	Dynamic	uint8	Detector used as a reference when computing relative detector gains and biases (least noisy), Band 5 Valid format: NN, where NN = TBS
GROUP: HISTOGRAM GROUP: REFERENCE_DETECTORS	Reference_Detector_B6	Dynamic	uint8	Detector used as a reference when computing relative detector gains and biases (least noisy), Band 6 Valid format: NN, where NN = TBS
GROUP: HISTOGRAM GROUP: REFERENCE_DETECTORS	Reference_Detector_B7	Dynamic	uint8	Detector used as a reference when computing relative detector gains and biases (least noisy), Band 7 Valid format: NN, where NN = TBS
GROUP: HISTOGRAM GROUP: SATURATION_ THRESHOLDS	Saturation_Bin_ Threshold_B1	Dynamic	uint16	Number of pixels that a bin must have to be tested as a saturation bin, Band 1 Valid format: NNNNN, where NNNNN = 10000 (default)
GROUP: HISTOGRAM GROUP: SATURATION_ THRESHOLDS	Saturation_Bin_ Threshold_B2	Dynamic	uint16	Number of pixels that a bin must have to be tested as a saturation bin, Band 2 Valid format: NNNNN, where NNNNN = 10000 (default)
GROUP: HISTOGRAM GROUP: SATURATION_ THRESHOLDS	Saturation_Bin_ Threshold_B3	Dynamic	uint16	Number of pixels that a bin must have to be tested as a saturation bin, Band 3 Valid format: NNNNN, where NNNNN = 10000 (default)
GROUP: HISTOGRAM GROUP: SATURATION_ THRESHOLDS	Saturation_Bin_ Threshold_B4	Dynamic	uint16	Number of pixels that a bin must have to be tested as a saturation bin, Band 4 Valid format: NNNNN, where NNNNN = 10000 (default)
GROUP: HISTOGRAM GROUP: SATURATION_ THRESHOLDS	Saturation_Bin_ Threshold_B5	Dynamic	uint16	Number of pixels that a bin must have to be tested as a saturation bin, Band 5 Valid format: NNNNN, where NNNNN = 10000 (default)
GROUP: HISTOGRAM GROUP: SATURATION_ THRESHOLDS	Saturation_Bin_ Threshold_B6	Dynamic	uint16	Number of pixels that a bin must have to be tested as a saturation bin, Band 6 Valid format: NNNNN, where NNNNN = 10000 (default)
GROUP: HISTOGRAM GROUP: SATURATION_ THRESHOLDS	Saturation_Bin_ Threshold_B7	Dynamic	uint16	Number of pixels that a bin must have to be tested as a saturation bin, Band 7 Valid format: NNNNN, where NNNNN = 10000 (default)
GROUP: HISTOGRAM GROUP: ADJACENT_BINS GROUP: BIN_NUMBER	Adjacent_Bin_ Number_B1	Dynamic	uint8	Bins adjacent to a possible saturation bin that must have fewer pixels than "adjacent bin threshold" to declare a possible bin as a saturation bin, Band 1 Valid format: N, where N = 2 (default)

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: HISTOGRAM GROUP: ADJACENT_BINS GROUP: BIN_NUMBER	Adjacent_Bin_ Number_B2	Dynamic	uint8	Bins adjacent to a possible saturation bin that must have fewer pixels than "adjacent bin threshold" to declare a possible bin as a saturation bin, Band 2 Valid format: N, where N = 2 (default)
GROUP: HISTOGRAM GROUP: ADJACENT_BINS GROUP: BIN_NUMBER	Adjacent_Bin_ Number_B3	Dynamic	uint8	Bins adjacent to a possible saturation bin that must have fewer pixels than "adjacent bin threshold" to declare a possible bin as a saturation bin, Band 3 Valid format: N, where N = 2 (default)
GROUP: HISTOGRAM GROUP: ADJACENT_BINS GROUP: BIN_NUMBER	Adjacent_Bin_ Number_B4	Dynamic	uint8	Bins adjacent to a possible saturation bin that must have fewer pixels than "adjacent bin threshold" to declare a possible bin as a saturation bin, Band 4 Valid format: N, where N = 2 (default)
GROUP: HISTOGRAM GROUP: ADJACENT_BINS GROUP: BIN_NUMBER	Adjacent_Bin_ Number_B5	Dynamic	uint8	Bins adjacent to a possible saturation bin that must have fewer pixels than "adjacent bin threshold" to declare a possible bin as a saturation bin, Band 5 Valid format: N, where N = 2 (default)
GROUP: HISTOGRAM GROUP: ADJACENT_BINS GROUP: BIN_NUMBER	Adjacent_Bin_ Number_B6	Dynamic	uint8	Bins adjacent to a possible saturation bin that must have fewer pixels than "adjacent bin threshold" to declare a possible bin as a saturation bin, Band 6 Valid format: N, where N = 3 (default)
GROUP: HISTOGRAM GROUP: ADJACENT_BINS GROUP: BIN_NUMBER	Adjacent_Bin_ Number_B7	Dynamic	uint8	Bins adjacent to a possible saturation bin that must have fewer pixels than "adjacent bin threshold" to declare a possible bin as a saturation bin, Band 7 Valid format: N, where N = 2 (default)
GROUP: HISTOGRAM GROUP: ADJACENT_BINS GROUP: BIN_THRESHOLD	Adjacent_Bin_ Threshold_B1	Dynamic	uint8	Number of adjacent bin pixels that cannot be exceeded for the Band 1 candidate saturation bin to be a valid saturation bin Valid format: NNNN, where NN = 1000 (default)
GROUP: HISTOGRAM GROUP: ADJACENT_BINS GROUP: BIN_THRESHOLD	Adjacent_Bin_ Threshold_B2	Dynamic	uint8	Number of adjacent bin pixels that cannot be exceeded for the Band 2 candidate saturation bin to be a valid saturation bin Valid format: NNNN, where NN = 1000 (default)
GROUP: HISTOGRAM GROUP: ADJACENT_BINS GROUP: BIN_THRESHOLD	Adjacent_Bin_ Threshold_B3	Dynamic	uint8	Number of adjacent bin pixels that cannot be exceeded for the Band 3 candidate saturation bin to be a valid saturation bin Valid format: NNNN, where NN = 1000 (default)
GROUP: HISTOGRAM GROUP: ADJACENT_BINS GROUP: BIN_THRESHOLD	Adjacent_Bin_ Threshold_B4	Dynamic	uint8	Number of adjacent bin pixels that cannot be exceeded for the Band 4 candidate saturation bin to be a valid saturation bin Valid format: NNNN, where NN = 1000 (default)
GROUP: HISTOGRAM GROUP: ADJACENT_BINS GROUP: BIN_THRESHOLD	Adjacent_Bin_ Threshold_B5	Dynamic	uint8	Number of adjacent bin pixels that cannot be exceeded for the Band 5 candidate saturation bin to be a valid saturation bin Valid format: NNNN, where NN = 1000 (default)
GROUP: HISTOGRAM GROUP: ADJACENT_BINS GROUP: BIN_THRESHOLD	Adjacent_Bin_ Threshold_B6	Dynamic	uint8	Number of adjacent bin pixels that cannot be exceeded for the Band 6 candidate saturation bin to be a valid saturation bin Valid format: NNNN, where NN = 1000 (default)
GROUP: HISTOGRAM GROUP: ADJACENT_BINS GROUP: BIN_THRESHOLD	Adjacent_Bin_ Threshold_B7	Dynamic	uint8	Number of adjacent bin pixels that cannot be exceeded for the Band 7 candidate saturation bin to be a valid saturation bin Valid format: NNNN, where NN = 1000 (default)
GROUP: HISTOGRAM GROUP: STARTING_PIXEL	Start_pixel_B1	Dynamic	uint8	Left-most pixel in the window to be tested, Band 1 Valid format: NNNN, where NNNN = (TBS)

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: HISTOGRAM GROUP: STARTING_PIXEL	Start_pixel_B2	Dynamic	uint8	Left-most pixel in the window to be tested, Band 2 Valid format: NNNN, where NNNN = (TBS)
GROUP: HISTOGRAM GROUP: STARTING_PIXEL	Start_pixel_B3	Dynamic	uint8	Left-most pixel in the window to be tested, Band 3 Valid format: NNNN, where NNNN = (TBS)
GROUP: HISTOGRAM GROUP: STARTING_PIXEL	Start_pixel_B4	Dynamic	uint8	Left-most pixel in the window to be tested, Band 4 Valid format: NNNN, where NNNN = (TBS)
GROUP: HISTOGRAM GROUP: STARTING_PIXEL	Start_pixel_B5	Dynamic	uint8	Left-most pixel in the window to be tested, Band 5 Valid format: NNNN, where NNNN = (TBS)
GROUP: HISTOGRAM GROUP: STARTING_PIXEL	Start_pixel_B6	Dynamic	uint8	Left-most pixel in the window to be tested, Band 6 Valid format: NNNN, where NNNN = (TBS)
GROUP: HISTOGRAM GROUP: STARTING_PIXEL	Start_pixel_B7	Dynamic	uint8	Left-most pixel in the window to be tested, Band 7 Valid format: NNNN, where NNNN = (TBS)
GROUP: HISTOGRAM GROUP: WINDOW_WIDTH	Window_Samples_B1	Dynamic	uint8	Width of the window, in pixels, to be tested, Band 1 Valid format: NNNN, where NNNN = (TBS)
GROUP: HISTOGRAM GROUP: WINDOW_WIDTH	Window_Samples_B2	Dynamic	uint8	Width of the window, in pixels, to be tested, Band 2 Valid format: NNN, where NNNN = (TBS)
GROUP: HISTOGRAM GROUP: WINDOW_WIDTH	Window_Samples_B3	Dynamic	uint8	Width of the window, in pixels, to be tested, Band 3 Valid format: NNNN, where NNNN = (TBS)
GROUP: HISTOGRAM GROUP: WINDOW_WIDTH	Window_Samples_B4	Dynamic	uint8	Width of the window, in pixels, to be tested, Band 4 Valid format: NNNN, where NNNN = (TBS)
GROUP: HISTOGRAM GROUP: WINDOW_WIDTH	Window_Samples_B5	Dynamic	uint8	Width of the window, in pixels, to be tested, Band 5 Valid format: NNNN, where NNNN = (TBS)
GROUP: HISTOGRAM GROUP: WINDOW_WIDTH	Window_Samples_B6	Dynamic	uint8	Width of the window, in pixels, to be tested, Band 6 Valid format: NNNN, where NNNN = (TBS)
GROUP: HISTOGRAM GROUP: WINDOW_WIDTH	Window_Samples_B7	Dynamic	uint8	Width of the window, in pixels, to be tested, Band 7 Valid format: NNNN, where NNNN = (TBS)
GROUP: HISTOGRAM GROUP: WINDOW_LENGTH	Window_Scans_B1	Dynamic	uint8	Number of scans in window to be tested, Band 1 Valid format: NNN, where NNN = (TBS)
GROUP: HISTOGRAM GROUP: WINDOW_LENGTH	Window_Scans_B2	Dynamic	uint8	Number of scans in window to be tested, Band 2 Valid format: NNN, where NNN = (TBS)
GROUP: HISTOGRAM GROUP: WINDOW_LENGTH	Window_Scans_B3	Dynamic	uint8	Number of scans in the window to be tested, Band 3 Valid format: NNN, where NNN = (TBS)
GROUP: HISTOGRAM GROUP: WINDOW_LENGTH	Window_Scans_B4	Dynamic	uint8	Number of scans in the window to be tested, Band 4 Valid format: NNN, where NNN = (TBS)
GROUP: HISTOGRAM GROUP: WINDOW_LENGTH	Window_Scans_B5	Dynamic	uint8	Number of scans in the window to be tested, Band 5 Valid format: NNN, where NNN = (TBS)
GROUP: HISTOGRAM GROUP: WINDOW_LENGTH	Window_Scans_B6	Dynamic	uint8	Number of scans in the window to be tested, Band 6 Valid format: NNN, where NNN = (TBS)
GROUP: HISTOGRAM GROUP: WINDOW_LENGTH	Window_Scans_B7	Dynamic	uint8	Number of scans in the window to be tested, Band 7 Valid format: NNN, where NNN = (TBS)

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: HISTOGRAM GROUP: OVERLAPPING_ SCANS	Overlap_Scans_B1	Dynamic	uint8	Number of overlapping scans between the windows to be tested, Band 1 Valid format: NNN, where NNN = (TBS)
GROUP: HISTOGRAM GROUP: OVERLAPPING_ SCANS	Overlap_Scans_B2	Dynamic	uint8	Number of overlapping scans between the windows to be tested, Band 2 Valid format: NNN, where NNN = (TBS)
GROUP: HISTOGRAM GROUP: OVERLAPPING_ SCANS	Overlap_Scans_B3	Dynamic	uint8	Number of overlapping scans between the windows to be tested, Band 3 Valid format: NNN, where NNN = (TBS)
GROUP: HISTOGRAM GROUP: OVERLAPPING_ SCANS	Overlap_Scans_B4	Dynamic	uint8	Number of overlapping scans between the windows to be tested, Band 4 Valid format: NNN, where NNN = (TBS)
GROUP: HISTOGRAM GROUP: OVERLAPPING_ SCANS	Overlap_Scans_B5	Dynamic	uint8	Number of overlapping scans between the windows to be tested, Band 5 Valid format: NNN, where NNN = (TBS)
GROUP: HISTOGRAM GROUP: OVERLAPPING_ SCANS	Overlap_Scans_B6	Dynamic	uint8	Number of overlapping scans between the windows to be tested, Band 6 Valid format: NNN, where NNN = (TBS)
GROUP: HISTOGRAM GROUP: OVERLAPPING_ SCANS	Overlap_Scans_B7	Dynamic	uint8	Number of overlapping scans between the windows to be tested, Band 7 Valid format: NNN, where NNN = (TBS)
GROUP: IMPULSE_NOISE	Median_Filter_Width	Static	uint8	Width of median filter Valid format: N, where N = 3
GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD	B1_Threshold	Dynamic	float32 array (16 values)	Band 1 noise threshold for an unequal case Valid format: NN.NNNNNN,where N = 0 to 9
GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD	B2_Threshold	Dynamic	float32 array (16 values)	Band 2 noise threshold for an unequal case Valid format: NN.NNNNNNN, where N = 0 to 9
GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD	B3_Threshold	Dynamic	float32 array (16 values)	Band 3 noise threshold for an unequal case Valid format: NN.NNNNNN, where N = 0 to 9
GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD	B4_Threshold	Dynamic	float32 array (16 values)	Band 4 noise threshold for an unequal case Valid format: NN.NNNNNN, where N = 0 to 9
GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD	B5_Threshold	Dynamic	float32 array (16 values)	Band 5 noise threshold for an unequal case Valid format: NN.NNNNNN, where N = 0 to 9
GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD	B6_Threshold	Dynamic	float32 array (4 values)	Band 6 noise threshold for an unequal case Valid format: NN.NNNNNN, where N = 0 to 9
GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD	B7_Threshold	Dynamic	float32 array (16 values)	Band 7 noise threshold for an unequal case Valid format: NN.NNNNNN, where N = 0 to 9
GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD	B1_Sigma_Threshold	Dynamic	float32 array (16 values)	Band 1 noise threshold for an equal case Valid format: NN.NNNNNN, where N = 0 to 9
GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD	B2_Sigma_Threshold	Dynamic	float32 array (16 values)	Band 2 noise threshold for an equal case Valid format: NN.NNNNNN, where N = 0 to 9
GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD	B3_Sigma_Threshold	Dynamic	float32 array (16 values)	Band 3 noise threshold for an equal case Valid format: NN.NNNNNN, where N = 0 to 9
GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD	B4_Sigma_Threshold	Dynamic	float32 array (16 values)	Band 4 noise threshold for an equal case Valid format: NN.NNNNNN, where N = 0 to 9
GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD	B5_Sigma_Threshold	Dynamic	float32 array (16 values)	Band 5 noise threshold for equal case Valid format: NN.NNNNNN, where N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD	B6_Sigma_Threshold	Dynamic	float32 array (4 values)	Band 6 noise threshold for an equal case Valid format: NN.NNNNNN, where N = 0 to 9
GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD	B7_Sigma_Threshold	Dynamic	float32 array (16 values)	Band 7 noise threshold for an equal case Valid format: NN.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE	Frequency_Components	Dynamic	uint8	Number of frequency components derived during waveform analysis for coherent noise correction Valid format: NN, where NN = 10
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_MEANS	B1_Frequency_Mean	Dynamic	float32 array (10 values)	Band 1 frequency means measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_MEANS	B2_Frequency_Mean	Dynamic	float32 array (10 values)	Band 2 frequency means measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_MEANS	B3_Frequency_Mean	Dynamic	float32 array (10 values)	Band 3 frequency means measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_MEANS	B4_Frequency_Mean	Dynamic	float32 array (10 values)	Band 4 frequency means measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_MEANS	B5_Frequency_Mean	Dynamic	float32 array (10 values)	Band 5 frequency means measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_MEANS	B6_Frequency_Mean	Dynamic	float32 array (10 values)	Band 6 frequency means measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_MEANS	B7_Frequency_Mean	Dynamic	float32 array (10 values)	Band 7 frequency means measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_SIGMAS	B1_Frequency_Sigma	Dynamic	float32 array (10 values)	Band 1 frequency sigmas measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_SIGMAS	B2_Frequency_Sigma	Dynamic	float32 array (10 values)	Band 2 frequency sigmas measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_SIGMAS	B3_Frequency_Sigma	Dynamic	float32 array (10 values)	Band 3 frequency sigmas measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_SIGMAS	B4_Frequency_Sigma	Dynamic	float32 array (10 values)	Band 4 frequency sigmas measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_SIGMAS	B5_Frequency_Sigma	Dynamic	float32 array (10 values)	Band 5 frequency sigmas measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_SIGMAS	B6_Frequency_Sigma	Dynamic	float32 array (10 values)	Band 6 frequency sigmas measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_SIGMAS	B7_Frequency_Sigma	Dynamic	float32 array (10 values)	Band 7 frequency sigmas measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_ MINIMUMS	B1_Frequency_Min	Dynamic	float32 array (10 values)	Band 1 frequency minimums measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_ MINIMUMS	B2_Frequency_Min	Dynamic	float32 array (10 values)	Band 2 frequency minimums measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_ MINIMUMS	B3_Frequency_Min	Dynamic	float32 array (10 values)	Band 3 frequency minimums measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_ MINIMUMS	B4_Frequency_Min	Dynamic	float32 array (10 values)	Band 4 frequency minimums measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_ MINIMUMS	B5_Frequency_Min	Dynamic	float32 array (10 values)	Band 5 frequency minimums measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_ MINIMUMS	B6_Frequency_Min	Dynamic	float32 array (10 values)	Band 6 frequency minimums measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_ MINIMUMS	B7_Frequency_Min	Dynamic	float32 array (10 values)	Band 7 frequency minimums measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_ MAXIMUMS	B1_Frequency_Max	Dynamic	float32 array (10 values)	Band 1 frequency maximums measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_ MAXIMUMS	B2_Frequency_Max	Dynamic	float32 array (10 values)	Band 2 frequency maximums measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_ MAXIMUMS	B3_Frequency_Max	Dynamic	float32 array (10 values)	Band 3 frequency maximums measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_ MAXIMUMS	B4_Frequency_Max	Dynamic	float32 array (10 values)	Band 4 frequency maximums measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_ MAXIMUMS	B5_Frequency_Max	Dynamic	float32 array (10 values)	Band 5 frequency maximums measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_ MAXIMUMS	B6_Frequency_Max	Dynamic	float32 array (10 values)	Band 6 frequency maximums measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_ MAXIMUMS	B7_Frequency_Max	Dynamic	float32 array (10 values)	Band 7 frequency maximums measured in inverse minor frames Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MEANS	B1_Phase_Mean	Dynamic	float32 array (10 values)	Band 1 phase means measured in radians Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MEANS	B2_Phase_Mean	Dynamic	float32 array (10 values)	Band 2 phase means measured in radians Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MEANS	B3_Phase_Mean	Dynamic	float32 array (10 values)	Band 3 phase means measured in radians Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MEANS	B4_Phase_Mean	Dynamic	float32 array (10 values)	Band 4 phase means measured in radians Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MEANS	B5_Phase_Mean	Dynamic	float32 array (10 values)	Band 5 phase means measured in radians Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MEANS	B6_Phase_Mean	Dynamic	float32 array (10 values)	Band 6 phase means measured in radians Valid format: N.NNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MEANS	B7_Phase_Mean	Dynamic	float32 array (10 values)	Band 7 phase means measured in radians Valid format: N.NNNNNN, where N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_PARAMETERS GROUP: MAGNITUDE_MEANS	B1_Magnitude_Mean	Dynamic	float32 array (10 values)	Band 1 magnitude means measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9.
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_PARAMETERS GROUP: MAGNITUDE_MEANS	B2_Magnitude_Mean	Dynamic	float32 array (10 values)	Band 2 magnitude means measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_PARAMETERS GROUP: MAGNITUDE_MEANS	B3_Magnitude_Mean	Dynamic	float32 array (10 values)	Band 3 magnitude means measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_PARAMETERS GROUP: MAGNITUDE_MEANS	B4_Magnitude_Mean	Dynamic	float32 array (10 values)	Band 4 magnitude means measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_PARAMETERS GROUP: MAGNITUDE_MEANS	B5_Magnitude_Mean	Dynamic	float32 array (10 values)	Band 5 magnitude means measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_PARAMETERS GROUP: MAGNITUDE_MEANS	B6_Magnitude_Mean	Dynamic	float32 array (10 values)	Band 6 magnitude means measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_PARAMETERS GROUP: MAGNITUDE_MEANS	B7_Magnitude_Mean	Dynamic	float32 array (10 values)	Band 7 magnitude means measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_PARAMETERS GROUP: MAGNITUDE_SIGMAS	B1_Magnitude_Sigma	Dynamic	float32 array (10 values)	Band 1 magnitude sigmas measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_PARAMETERS GROUP: MAGNITUDE_SIGMAS	B2_Magnitude_Sigma	Dynamic	float32 array (10 values)	Band 2 magnitude sigmas measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_PARAMETERS GROUP: MAGNITUDE_SIGMAS	B3_Magnitude_Sigma	Dynamic	float32 array (10 values)	Band 3 magnitude sigmas measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_PARAMETERS GROUP: MAGNITUDE_SIGMAS	B4_Magnitude_Sigma	Dynamic	float32 array (10 values)	Band 4 magnitude sigmas measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_PARAMETERS GROUP: MAGNITUDE_SIGMAS	B5_Magnitude_Sigma	Dynamic	float32 array (10 values)	Band 5 magnitude sigmas measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_SIGMAS	B6_Magnitude_Sigma	Dynamic	float32 array (10 values)	Band 6 magnitude sigmas measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_SIGMAS	B7_Magnitude_Sigma	Dynamic	float32 array (10 values)	Band 7 magnitude sigmas measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MINIMUMS	B1_Magnitude_Min	Dynamic	float32 array (10 values)	Band 1 magnitude minimums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MINIMUMS	B2_Magnitude_Min	Dynamic	float32 array (10 values)	Band 2 magnitude minimums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MINIMUMS	B3_Magnitude_Min	Dynamic	float32 array (10 values)	Band 3 magnitude minimums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MINIMUMS	B4_Magnitude_Min	Dynamic	float32 array (10 values)	Band 4 magnitude minimums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MINIMUMS	B5_Magnitude_Min	Dynamic	float32 array (10 values)	Band 5 magnitude minimums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MINIMUMS	B6_Magnitude_Min	Dynamic	float32 array (10 values)	Band 6 magnitude minimums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MINIMUMS	B7_Magnitude_Min	Dynamic	float32 array (10 values)	Band 7 magnitude minimums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MAXIMUMS	B1_Magnitude_Max	Dynamic	float32 array (10 values)	Band 1 magnitude maximums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MAXIMUMS	B2_Magnitude_Max	Dynamic	float32 array (10 values)	Band 2 magnitude maximums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MAXIMUMS	B3_Magnitude_Max	Dynamic	float32 array (10 values)	Band 3 magnitude maximums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MAXIMUMS	B4_Magnitude_Max	Dynamic	float32 array (10 values)	Band 4 magnitude maximums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MAXIMUMS	B5_Magnitude_Max	Dynamic	float32 array (10 values)	Band 5 magnitude maximums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MAXIMUMS	B6_Magnitude_Max	Dynamic	float32 array (10 values)	Band 6 magnitude maximums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MAXIMUMS	B7_Magnitude_Max	Dynamic	float32 array (10 values)	Band 7 magnitude maximums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9
GROUP: CHANNEL_SATURATION	High_Level_B1	Dynamic	uint8 array (16 values)	Digital count at which the channel saturates at the high end in Band 1; array contains one value per detector Valid format: NNN, where NNN = 255 (default)
GROUP: CHANNEL_SATURATION	High_Level_B2	Dynamic	uint8 array (16 values)	Digital count at which the channel saturates at the high end in Band 2; array contains one value per detector Valid format: NNN, where NNN = 255 (default)
GROUP: CHANNEL_SATURATION	High_Level_B3	Dynamic	uint8 array (16 values)	Digital count at which the channel saturates at the high end in Band 3; array contains one value per detector Valid format: NNN, where NNN = 255 (default)
GROUP: CHANNEL_SATURATION	High_Level_B4	Dynamic	uint8 array (16 values)	Digital count at which the channel saturates at the high end in Band 4; array contains one value per detector Valid format: NNN, where NNN = 255 (default)
GROUP: CHANNEL_SATURATION	High_Level_B5	Dynamic	uint8 array (16 values)	Digital count at which the channel saturates at the high end in Band 5; array contains one value per detector Valid format: NNN, where NNN = 255 (default)
GROUP: CHANNEL_SATURATION	High_Level_B6	Dynamic	uint8 array (4 values)	Digital count at which the channel saturates at the high end in Band 6; array contains one value per detector Valid format: NNN, where NNN = 255 (default)
GROUP: CHANNEL_SATURATION	High_Level_B7	Dynamic	uint8 array (16 values)	Digital count at which the channel saturates at the high end in Band 7; array contains one value per detector Valid format: NNN, where NNN = 255 (default)
GROUP: CHANNEL_SATURATION	Low_Level_B1	Dynamic	uint8 array (16 values)	Digital count at which the channel saturates at the low end in Band 1; array contains one value per detector Valid format: NNN, where NNN = 000 (default)
GROUP: CHANNEL_SATURATION	Low_Level_B2	Dynamic	uint8 array (16 values)	Digital count at which the channel saturates at the low end in Band 2; array contains one value per detector Valid format: NNN, where NNN = 000 (default)
GROUP: CHANNEL_SATURATION	Low_Level_B3	Dynamic	uint8 array (16 values)	Digital count at which the channel saturates at the low end in Band 3; array contains one value per detector Valid format: NNN, where NNN = 000 (default)

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: CHANNEL_SATURATION	Low_Level_B4	Dynamic	uint8 array (16 values)	Digital count at which the channel saturates at the low end in Band 4; array contains one value per detector Valid format: NNN, where NNN = 000 (default)
GROUP: CHANNEL_SATURATION	Low_Level_B5	Dynamic	uint8 array (16 values)	Digital count at which the channel saturates at the low end in Band 5; array contains one value per detector Valid format: NNN, where NNN = 000 (default)
GROUP: CHANNEL_SATURATION	Low_Level_B6	Dynamic	uint8 array (4 values)	Digital count at which the channel saturates at the low end in Band 6; array contains one value per detector Valid format: NNN, where NNN = 000 (default)
GROUP: CHANNEL_SATURATION	Low_Level_B7	Dynamic	uint8 array (16 values)	Digital count at which the channel saturates at the low end in Band 7; array contains one value per detector Valid format: NNN, where NNN = 000 (default)
GROUP: REFERENCE_ TEMPERATURES	B1_RTemp	Static	float64	Band 1 calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 23.000
GROUP: REFERENCE_ TEMPERATURES	B2_RTemp	Static	float64	Band 2 calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 23.00
GROUP: REFERENCE_ TEMPERATURES	B3_RTemp	Static	float64	Band 3 calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 23.000
GROUP: REFERENCE_ TEMPERATURES	B4_RTemp	Static	float64	Band 4 calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 23.000
GROUP: REFERENCE_ TEMPERATURES	B5_RTemp	Static	float64	Band 5 calibration reference temperature in degrees C Valid format: SNNN.NN, where SNNN.NN = -181.00
GROUP: REFERENCE_ TEMPERATURES	B6_RTemp	Static	float64	Band 6 calibration reference temperature in degrees C Valid format: SNNN.NN, where SNNN.NN = -181.00
GROUP: REFERENCE_ TEMPERATURES	B7_RTemp	Static	float64	Band 7 calibration reference temperature in degrees C Valid format: SNNN.NN, where SNNN.NN = -181.00
GROUP: SENSITIVITY_ TEMPERATURES	B1_SCoeff	Dynamic	float64 array (16 values)	Band 1 calibration temperature sensitivity coefficient; array contains one value per detector Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SENSITIVITY_ TEMPERATURES	B2_SCoeff	Dynamic	float64 array (16 values)	Band 2 calibration temperature sensitivity coefficient; array contains one value per detector Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SENSITIVITY_ TEMPERATURES	B3_SCoeff	Dynamic	float64 array (16 values)	Band 3 calibration temperature sensitivity coefficient; array contains one value per detector Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SENSITIVITY_ TEMPERATURES	B4_SCoeff	Dynamic	float64 array (16 values)	Band 4 calibration temperature sensitivity coefficient; array contains one value per detector Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9

Parameter Groups	**Parameter Name**	**Value Type**	**Data Type**	**Description**
GROUP: SENSITIVITY_ TEMPERATURES	B5_SCoeff	Dynamic	float64 array (16 values)	Band 5 calibration temperature sensitivity coefficient; array contains one value per detector Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SENSITIVITY_ TEMPERATURES	B6_SCoeff	Dynamic	float64 array (4 values)	Band 6 calibration temperature sensitivity coefficient; array contains one value per detector Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SENSITIVITY_ TEMPERATURES	B6_SCoeff_Off	Dynamic	float64 array (4 values)	Band 6 offset calibration temperature sensitivity coefficient; array contains one value per detector Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: SENSITIVITY_ TEMPERATURES	B7_SCoeff	Dynamic	float64 array (16 values)	Band 7 calibration temperature sensitivity coefficient; array contains one value per detector Valid format: SNNN.NNNN, where S = + or –, and N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: TRENDING_COEFFS	Lamp1_Coeffs	Static	float32 array (2 values)	Time since launch trending coefficients for Lamp 1 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: TRENDING_COEFFS	Lamp2_Coeffs	Static	float32 array (2 values)	Time since launch trending coefficients for Lamp 2 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: TRENDING_COEFFS	Lamp3_Coeffs	Static	float32 array (2 values)	Time since launch trending coefficients for Lamp 3 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_000_RADIANCE	B1_Rad_State_000	Static	float32 array (16 values)	Band 1 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-μm State 000: Off-Off-Off Valid format: SNNN.NN, where S = + or –, and N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_000_RADIANCE	B2_Rad_State_000	Static	float32 array (16 values)	Band 2 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-μm. State 000: Off-Off-Off Valid format: SNNN.NN, where S = "+" or "-" and N = 0 to 9.
GROUP: LAMP_RADIANCE GROUP: STATE_000_RADIANCE	B3_Rad_State_000	Static	float32 array (16 values)	Band 3 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-μm State 000: Off-Off-Off Valid format: SNNN.NN, where S = + or –, and N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_000_RADIANCE	B4_Rad_State_000	Static	float32 array (16 values)	Band 4 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-μm State 000: Off-Off-Off Valid format: SNNN.NN, where S = + or –, and N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_000_RADIANCE	B5_Rad_State_000	Static	float32 array (16 values)	Band 5 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-μm State 000: Off-Off-Off Valid format: SNNN.NN, where S = + or –, and N = 0 to 9

Parameter Groups	**Parameter Name**	**Value Type**	**Data Type**	**Description**
GROUP: LAMP_RADIANCE GROUP: STATE_000_RADIANCE	B7_Rad_State_000	Static	float32 array (16 values)	Band 7 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 000: Off-Off-Off Valid format: SNNN.NN, where S = + or –, and N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_001_RADIANCE	B1_Rad_State_001	Static	float32 array (16 values)	Band 1 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 001: Off-Off-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_001_RADIANCE	B2_Rad_State_001	Static	float32 array (16 values)	Band 2 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 001: Off-Off-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_001_RADIANCE	B3_Rad_State_001	Static	float32 array (16 values)	Band 3 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 001: Off-Off-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_001_RADIANCE	B4_Rad_State_001	Static	float32 array (16 values)	Band 4 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 001: Off-Off-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_001_RADIANCE	B5_Rad_State_001	Static	float32 array (16 values)	Band 5 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 001: Off-Off-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_001_RADIANCE	B7_Rad_State_001	Static	float32 array (16 values)	Band 7 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 001: Off-Off-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_010_RADIANCE	B1_Rad_State_010	Static	float32 array (16 values)	Band 1 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 010: Off-On-Off Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_010_RADIANCE	B2_Rad_State_010	Static	float32 array (16 values)	Band 2 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 010: Off-On-Off Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_010_RADIANCE	B3_Rad_State_010	Static	float32 array (16 values)	Band 3 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 010: Off-On-Off Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_010_RADIANCE	B4_Rad_State_010	Static	float32 array (16 values)	Band 4 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 010: Off-On-Off Valid format: NNN.NN, where N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: LAMP_RADIANCE GROUP: STATE_010_RADIANCE	B5_Rad_State_010	Static	float32 array (16 values)	Band 5 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-μm State 010: Off-On-Off Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_010_RADIANCE	B7_Rad_State_010	Static	float32 array (16 values)	Band 7 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-μm State 010: Off-On-Off Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_011_RADIANCE	B1_Rad_State_011	Static	float32 array (16 values)	Band 1 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-μm State 011: Off-On-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_011_RADIANCE	B2_Rad_State_011	Static	float32 array (16 values)	Band 2 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-μm State 011: Off-On-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_011_RADIANCE	B3_Rad_State_011	Static	float32 array (16 values)	Band 3 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-μm State 011: Off-On-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_011_RADIANCE	B4_Rad_State_011	Static	float32 array (16 values)	Band 4 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-μm State 011: Off-On-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_011_RADIANCE	B5_Rad_State_011	Static	float32 array (16 values)	Band 5 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-μm State 011: Off-On-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_011_RADIANCE	B7_Rad_State_011	Static	float32 array (16 values)	Band 7 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-μm State 011: Off-On-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_100_RADIANCE	B1_Rad_State_100	Static	float32 array (16 values)	Band 1 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-μm State 100: On-Off-Off Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_100_RADIANCE	B2_Rad_State_100	Static	float32 array (16 values)	Band 2 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-μm State 100: On-Off-Off Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_100_RADIANCE	B3_Rad_State_100	Static	float32 array (16 values)	Band 3 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-μm State 100: On-Off-Off Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_100_RADIANCE	B4_Rad_State_100	Static	float32 array (16 values)	Band 4 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-μm State 100: On-Off-Off Valid format: NNN.NN, where N = 0 to 9

Parameter Groups	**Parameter Name**	**Value Type**	**Data Type**	**Description**
GROUP: LAMP_RADIANCE GROUP: STATE_100_RADIANCE	B5_Rad_State_100	Static	float32 array (16 values)	Band 5 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 100: On-Off-Off Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_100_RADIANCE	B7_Rad_State_100	Static	float32 array (16 values)	Band 7 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 100: On-Off-Off Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_101_RADIANCE	B1_Rad_State_101	Static	float32 array (16 values)	Band 1 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 101: On-Off-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_101_RADIANCE	B2_Rad_State_101	Static	float32 array (16 values)	Band 2 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µ State 101: On-Off-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_101_RADIANCE	B3_Rad_State_101	Static	float32 array (16 values)	Band 3 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 101: On-Off-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_101_RADIANCE	B4_Rad_State_101	Static	float32 array (16 values)	Band 4 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 101: On-Off-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_101_RADIANCE	B5_Rad_State_101	Static	float32 array (16 values)	Band 5 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 101: On-Off-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_101_RADIANCE	B7_Rad_State_101	Static	float32 array (16 values)	Band 7 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 101: On-Off-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_110_RADIANCE	B1_Rad_State_110	Static	float32 array (16 values)	Band 1 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 110: Off-On-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_110_RADIANCE	B2_Rad_State_110	Static	float32 array (16 values)	Band 2 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 110: On-On-Off Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_110_RADIANCE	B3_Rad_State_110	Static	float32 array (16 values)	Band 3 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 110: On-On-Off Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_110_RADIANCE	B4_Rad_State_110	Static	float32 array (16 values)	Band 4 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 110: On-On-Off Valid format: NNN.NN, where N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: LAMP_RADIANCE GROUP: STATE_110_RADIANCE	B5_Rad_State_110	Static	float32 array (16 values)	Band 5 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 110: On-On-Off Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_110_RADIANCE	B7_Rad_State_110	Static	float32 array (16 values)	Band 7 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 110: On-On-Off Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_111_RADIANCE	B1_Rad_State_111	Static	float32 array (16 values)	Band 1 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 111: On-On-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_111_RADIANCE	B2_Rad_State_111	Static	float32 array (16 values)	Band 2 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 111: On-On-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_111_RADIANCE	B3_Rad_State_111	Static	float32 array (16 values)	Band 3 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 111: On-On-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_111_RADIANCE	B4_Rad_State_111	Static	float32 array (16 values)	Band 4 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 111: On-On-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_111_RADIANCE	B5_Rad_State_111	Static	float32 array (16 values)	Band 5 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 111: On-On-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_RADIANCE GROUP: STATE_111_RADIANCE	B7_Rad_State_111	Static	float32 array (16 values)	Band 7 internal calibrator prelaunch lamp effective per-detector spectral radiance in W/m^2-ster-µm State 111: On-On-On Valid format: NNN.NN, where N = 0 to 9
GROUP: LAMP_REFERENCE	Lmp_Rtemp	Static	float32 array (10 values)	Internal calibrator lamp radiance reference temperatures in degrees C Valid format: SNNN.N, where S = + or –, and NNN.N = TBS T1 = Blackbody temp T2 = Silicon focal plane array temp T3 = Cal shutter flag temp T4 = Baffle temp T5 = Cold stage focal plane array temp T6 = Scan line corrector temp T7 = Cal shutter hub temp T8 = Relay optics temp T9 = Primary mirror temp T10 = Secondary mirror temp
GROUP: REFLECTIVE_IC_ COEFFS	B1_Coefficients_ Detector1	Dynamic	float32 array (14 values)	IC coefficients for Band 1, detector 1 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B1_Coefficients_ Detector2	Dynamic	float32 array (14 values)	IC coefficients for Band 1, detector 2 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: REFLECTIVE_IC_COEFFS	B1_Coefficients_Detector3	Dynamic	float32 array (14 values)	IC coefficients for Band 1, detector 3 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_COEFFS	B1_Coefficients_Detector4	Dynamic	float32 array (14 values)	IC coefficients for Band 1, detector 4 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_COEFFS	B1_Coefficients_Detector5	Dynamic	float32 array (14 values)	IC coefficients for Band 1, detector 5 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_COEFFS	B1_Coefficients_Detector6	Dynamic	float32 array (14 values)	IC coefficients for Band 1, detector 6 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_COEFFS	B1_Coefficients_Detector7	Dynamic	float32 array (14 values)	IC coefficients for Band 1, detector 7 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_COEFFS	B1_Coefficients_Detector8	Dynamic	float32 array (14 values)	IC coefficients for Band 1, detector 8 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_COEFFS	B1_Coefficients_Detector9	Dynamic	float32 array (14 values)	IC coefficients for Band 1, detector 9 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_COEFFS	B1_Coefficients_Detector10	Dynamic	float32 array (14 values)	IC coefficients for Band 1, detector 10 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_COEFFS	B1_Coefficients_Detector11	Dynamic	float32 array (14 values)	IC coefficients for Band 1, detector 11 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_COEFFS	B1_Coefficients_Detector12	Dynamic	float32 array (14 values)	IC coefficients for Band 1, detector 12 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_COEFFS	B1_Coefficients_Detector13	Dynamic	float32 array (14 values)	IC coefficients for Band 1, detector 13 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_COEFFS	B1_Coefficients_Detector14	Dynamic	float32 array (14 values)	IC coefficients for Band 1, detector 14 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_COEFFS	B1_Coefficients_Detector15	Dynamic	float32 array (14 values)	IC coefficients for Band 1, detector 15 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_COEFFS	B1_Coefficients_Detector16	Dynamic	float32 array (14 values)	IC coefficients for Band 1, detector 16 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_COEFFS	B2_Coefficients_Detector1	Dynamic	float32 array (14 values)	IC coefficients for Band 2, detector 1 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_COEFFS	B2_Coefficients_Detector2	Dynamic	float32 array (14 values)	IC coefficients for Band 2, detector 2 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_COEFFS	B2_Coefficients_Detector3	Dynamic	float32 array (14 values)	IC coefficients for Band 2, detector 3 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_COEFFS	B2_Coefficients_Detector4	Dynamic	float32 array (14 values)	IC coefficients for Band 2, detector 4 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_COEFFS	B2_Coefficients_Detector5	Dynamic	float32 array (14 values)	IC coefficients for Band 2, detector 5 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: REFLECTIVE_IC_ COEFFS	B2_Coefficients_ Detector6	Dynamic	float32 array (14 values)	IC coefficients for Band 2, detector 6 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B2_Coefficients_ Detector7	Dynamic	float32 array (14 values)	IC coefficients for Band 2, detector 7 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B2_Coefficients_ Detector8	Dynamic	float32 array (14 values)	IC coefficients for Band 2, detector 8 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B2_Coefficients_ Detector9	Dynamic	float32 array (14 values)	IC coefficients for Band 2, detector 9 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B2_Coefficients_ Detector10	Dynamic	float32 array (14 values)	IC coefficients for Band 2, detector 10 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B2_Coefficients_ Detector11	Dynamic	float32 array (14 values)	IC coefficients for Band 2, detector 11 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B2_Coefficients_ Detector12	Dynamic	float32 array (14 values)	IC coefficients for Band 2, detector 12 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B2_Coefficients_ Detector13	Dynamic	float32 array (14 values)	IC coefficients for Band 2, detector 13 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B2_Coefficients_ Detector14	Dynamic	float32 array (14 values)	IC coefficients for Band 2, detector 14 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B2_Coefficients_ Detector15	Dynamic	float32 array (14 values)	IC coefficients for Band 2, detector 15 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B2_Coefficients_ Detector16	Dynamic	float32 array (14 values)	IC coefficients for Band 2, detector 16 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B3_Coefficients_ Detector1	Dynamic	float32 array (14 values)	IC coefficients for Band 3, detector 1 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B3_Coefficients_ Detector2	Dynamic	float32 array (14 values)	IC coefficients for Band 3, detector 2 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B3_Coefficients_ Detector3	Dynamic	float32 array (14 values)	IC coefficients for Band 3, detector 3 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B3_Coefficients_ Detector4	Dynamic	float32 array (14 values)	IC coefficients for Band 3, detector 4 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B3_Coefficients_ Detector5	Dynamic	float32 array (14 values)	IC coefficients for Band 3, detector 5 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B3_Coefficients_ Detector6	Dynamic	float32 array (14 values)	IC coefficients for Band 3, detector 6 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B3_Coefficients_ Detector7	Dynamic	float32 array (14 values)	IC coefficients for Band 3, detector 7 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B3_Coefficients_ Detector8	Dynamic	float32 array (14 values)	IC coefficients for Band 3, detector 8 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: REFLECTIVE_IC_ COEFFS	B3_Coefficients_ Detector9	Dynamic	float32 array (14 values)	IC coefficients for Band 3, detector 9 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B3_Coefficients_ Detector10	Dynamic	float32 array (14 values)	IC coefficients for Band 3, detector 10 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B3_Coefficients_ Detector11	Dynamic	float32 array (14 values)	IC coefficients for Band 3, detector 11 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B3_Coefficients_ Detector12	Dynamic	float32 array (14 values)	IC coefficients for Band 3, detector 12 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B3_Coefficients_ Detector13	Dynamic	float32 array (14 values)	IC coefficients for Band 3, detector 13 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B3_Coefficients_ Detector14	Dynamic	float32 array (14 values)	IC coefficients for Band 3, detector 14 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B3_Coefficients_ Detector15	Dynamic	float32 array (14 values)	IC coefficients for Band 3, detector 15 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B3_Coefficients_ Detector16	Dynamic	float32 array (14 values)	IC coefficients for Band 3, detector 16 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B4_Coefficients_ Detector1	Dynamic	float32 array (14 values)	IC coefficients for Band 4, detector 1 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B4_Coefficients_ Detector2	Dynamic	float32 array (14 values)	IC coefficients for Band 4, detector 2 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B4_Coefficients_ Detector3	Dynamic	float32 array (14 values)	IC coefficients for Band 4, detector 3 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B4_Coefficients_ Detector4	Dynamic	float32 array (14 values)	IC coefficients for Band 4, detector 4 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B4_Coefficients_ Detector5	Dynamic	float32 array (14 values)	IC coefficients for Band 4, detector 5 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B4_Coefficients_ Detector6	Dynamic	float32 array (14 values)	IC coefficients for Band 4, detector 6 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B4_Coefficients_ Detector7	Dynamic	float32 array (14 values)	IC coefficients for Band 4, detector 7 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B4_Coefficients_ Detector8	Dynamic	float32 array (14 values)	IC coefficients for Band 4, detector 8 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B4_Coefficients_ Detector9	Dynamic	float32 array (14 values)	IC coefficients for Band 4, detector 9 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B4_Coefficients_ Detector10	Dynamic	float32 array (14 values)	IC coefficients for Band 4, detector 10 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B4_Coefficients_ Detector11	Dynamic	float32 array (14 values)	IC coefficients for Band 4, detector 11 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: REFLECTIVE_IC_ COEFFS	B4_Coefficients_ Detector12	Dynamic	float32 array (14 values)	IC coefficients for Band 4, detector 12 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B4_Coefficients_ Detector13	Dynamic	float32 array (14 values)	IC coefficients for Band 4, detector 13 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B4_Coefficients_ Detector14	Dynamic	float32 array (14 values)	IC coefficients for Band 4, detector 14 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B4_Coefficients_ Detector15	Dynamic	float32 array (14 values)	IC coefficients for Band 4, detector 15 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B4_Coefficients_ Detector16	Dynamic	float32 array (14 values)	IC coefficients for Band 4, detector 16 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B5_Coefficients_ Detector1	Dynamic	float32 array (14 values)	IC coefficients for Band 5, detector 1 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B5_Coefficients_ Detector2	Dynamic	float32 array (14 values)	IC coefficients for Band 5, detector 2 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B5_Coefficients_ Detector3	Dynamic	float32 array (14 values)	IC coefficients for Band 5, detector 3 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B5_Coefficients_ Detector4	Dynamic	float32 array (14 values)	IC coefficients for Band 5, detector 4 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B5_Coefficients_ Detector5	Dynamic	float32 array (14 values)	IC coefficients for Band 5, detector 5 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B5_Coefficients_ Detector6	Dynamic	float32 array (14 values)	IC coefficients for Band 5, detector 6 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B5_Coefficients_ Detector7	Dynamic	float32 array (14 values)	IC coefficients for Band 5, detector 7 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B5_Coefficients_ Detector8	Dynamic	float32 array (14 values)	IC coefficients for Band 5, detector 8 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B5_Coefficients_ Detector9	Dynamic	float32 array (14 values)	IC coefficients for Band 5, detector 9 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B5_Coefficients_ Detector10	Dynamic	float32 array (14 values)	IC coefficients for Band 5, detector 10 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B5_Coefficients_ Detector11	Dynamic	float32 array (14 values)	IC coefficients for Band 5, detector 11 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B5_Coefficients_ Detector12	Dynamic	float32 array (14 values)	IC coefficients for Band 5, detector 12 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B5_Coefficients_ Detector13	Dynamic	float32 array (14 values)	IC coefficients for Band 5, detector 13 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B5_Coefficients_ Detector14	Dynamic	float32 array (14 values)	IC coefficients for Band 5, detector 14 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: REFLECTIVE_IC_ COEFFS	B5_Coefficients_ Detector15	Dynamic	float32 array (14 values)	IC coefficients for Band 5, detector 15 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B5_Coefficients_ Detector16	Dynamic	float32 array (14 values)	IC coefficients for Band 5, detector 16 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B7_Coefficients_ Detector1	Dynamic	float32 array (14 values)	IC coefficients for Band 7, detector 1 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B7_Coefficients_ Detector2	Dynamic	float32 array (14 values)	IC coefficients for Band 7, detector 2 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B7_Coefficients_ Detector3	Dynamic	float32 array (14 values)	IC coefficients for Band 7, detector 3 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B7_Coefficients_ Detector4	Dynamic	float32 array (14 values)	IC coefficients for Band 7, detector 4 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B7_Coefficients_ Detector5	Dynamic	float32 array (14 values)	IC coefficients for Band 7, detector 5 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B7_Coefficients_ Detector6	Dynamic	float32 array (14 values)	IC coefficients for Band 7, detector 6 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B7_Coefficients_ Detector7	Dynamic	float32 array (14 values)	IC coefficients for Band 7, detector 7 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B7_Coefficients_ Detector8	Dynamic	float32 array (14 values)	IC coefficients for Band 7, detector 8 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B7_Coefficients_ Detector9	Dynamic	float32 array (14 values)	IC coefficients for Band 7, detector 9 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B7_Coefficients_ Detector10	Dynamic	float32 array (14 values)	IC coefficients for Band 7, detector 10 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B7_Coefficients_ Detector11	Dynamic	float32 array (14 values)	IC coefficients for Band 7, detector 11 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B7_Coefficients_ Detector12	Dynamic	float32 array (14 values)	IC coefficients for Band 7, detector 12 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B7_Coefficients_ Detector13	Dynamic	float32 array (14 values)	IC coefficients for Band 7, detector 13 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B7_Coefficients_ Detector14	Dynamic	float32 array (14 values)	IC coefficients for Band 7, detector 14 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B7_Coefficients_ Detector15	Dynamic	float32 array (14 values)	IC coefficients for Band 7, detector 15 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: REFLECTIVE_IC_ COEFFS	B7_Coefficients_ Detector16	Dynamic	float32 array (14 values)	IC coefficients for Band 7, detector 16 Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: THERMISTOR_ COEFFS	Black_Body_Temp	Static	float32 array (6 values)	Housekeeping telemetry conversion coefficients for raw blackbody temperature Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9

Parameter Groups	Parameter Name	Value Type	Data Type	Description
GROUP: THERMISTOR_ COEFFS	Silicon_FP_Array_ Temp	Static	float32 array (6 values)	Housekeeping telemetry conversion coefficients for raw primary focal plane temperature Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: THERMISTOR_ COEFFS	Cal_Shutter_Flag_Temp	Static	float32 array (6 values)	Housekeeping telemetry conversion coefficients for raw calibration shutter flag temperature Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: THERMISTOR_ COEFFS	Baffle_Temp	Static	float32 array (6 values)	Housekeeping telemetry conversion coefficients for raw baffle temperature Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: THERMISTOR_ COEFFS	Cold_Stage_FP_Array_Temp	Static	float32 array (6 values)	Housekeeping telemetry conversion coefficients for raw cold focal plane temperature Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: THERMISTOR_ COEFFS	Scan_Line_Corrector_ Temp	Static	float32 array (6 values)	Housekeeping telemetry conversion coefficients for raw scan-line corrector temperature Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: THERMISTOR_ COEFFS	Cal_Shutter_Hub_ Temp	Static	float32 array (6 values)	Housekeeping telemetry conversion coefficients for raw calibration shutter hub temperature Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: THERMISTOR_ COEFFS	Relay_Optics_Temp	Static	float32 array (6 values)	Housekeeping telemetry conversion coefficients for raw relay optics temperature Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: THERMISTOR_ COEFFS	Primary_Mirror_Temp	Static	float32 array (6 values)	Housekeeping telemetry conversion coefficients for raw primary mirror temperature Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: THERMISTOR_ COEFFS	Secondary_Mirror_ Temp	Static	float32 array (6 values)	Housekeeping telemetry conversion coefficients for raw secondary mirror temperature Valid format: SNNN.NNNNNNN, where S = + or –, and N = 0 to 9
GROUP: FILL_PATTERNS	Band_Fill_Pattern	Static	uint8 array (2 values)	Fill pattern used by LPS for filling erroneous or missing image data minor frames Valid format: NNN, where NNN = (0, 255) (alternating 0, 255s)

Table 3-1. Landsat 5 TM CPF Parameters

Section 4 CPF ODL

4.1 Introduction to ODL Syntax

The ODL syntax employs the following conventions:

- The parameter definition is in the form of parameter = value.
- The value can be either a scalar or an array. Array values are enclosed in parentheses and are separated by commas.
- Parameter arrays can and do exist on multiple lines.
- A carriage return <CR> and line feed <LF> end each line in the file.
- Blank spaces and lines are ignored.
- Each line of comments must begin with /* and end with */, including comments embedded on the same line as a parameter definition.
- Quotation marks are required for values that are text strings, including single characters. The exceptions to this rule are the GROUP and END_GROUP identifiers or values, which do not use quotation marks. The third and fourth parameters in the file, Effective_Date_Begin and Effective_Date_End, also do not have quotation marks. ODL recognizes dates if they follow prescribed formats.
- In general, case is not significant for the ODL. For the CPF, however, the case is significant for keyword and group names of the CPF. Group names are all uppercase letters and keywords are mixed case.
- Indentation is not significant but is used for readability.
- The reserve word END concludes the file.

Unavailable parameter values are denoted by TBS (to be supplied). During the initial phases of TM functionality of IAS development, many of the parameters will be TBS. Once full TM IAS functionality is in place, the parameters will be populated, and the CPFs will be distributed appropriately.

4.2 Sample TM CPF ODL File

The following is a prototype of a CPF file that contains valid parameter values for the third calendar quarter of 2005:

```
GROUP = FILE_ATTRIBUTES
  Spacecraft_Name = "Landsat_5"
  Sensor_Name = "Thematic_Mapper"
  Effective_Date_Begin = 2005-07-01
  Effective_Date_End = 2005-09-30
  CPF_File_Name = "L5CPF20050701_20050930.03"
END_GROUP = FILE_ATTRIBUTES

GROUP = EARTH_CONSTANTS
  Ellipsoid_Name = "WGS84"
  Semi_Major_Axis = 6378137.0000
  Semi_Minor_Axis = 6356752.3142
  Ellipticity = 0.00335281066474
  Eccentricity = 0.00669437999013
  Earth_Spin_Rate = 72.921158553E-06
  Gravity_Constant = 3.986005E14
  J2_Earth_Model_Term = 1082.63E-06
```

```
END_GROUP = EARTH_CONSTANTS

GROUP = ORBIT_PARAMETERS
  WRS_Cycle_Days = 16
  WRS_Cycle_Orbits = 233
  Scenes_Per_Orbit = 248
  Orbital_Period = 5933.0472
  Angular_Momentum = 53.136250E9
  Orbit_Radius = 7083.4457
  Orbit_Semimajor_Axis = 7083.4457
  Orbit_Semiminor_Axis = 7083.4408
  Orbit_Eccentricity = 0.00117604
  Inclination_Angle = 98.2096
  Argument_Of_Perigee = 90.0
  Descending_Node_Row = 60
  Long_Path1_Row60 = -64.6
  Descending_Node_Time_Min = "09:10"
  Descending_Node_Time_Max = "10:15"
  Nodal_Regression_Rate = 0.985647366
END_GROUP = ORBIT_PARAMETERS

GROUP = SCANNER_PARAMETERS
  Lines_Per_Scan_30 = 16
  Lines_Per_Scan_120 = 4
  Scans_Per_Scene = 374
  Swath_Angle = 0.26861
  Scan_Rate = 2.21095
  Dwell_Time_30 = 9.6106302
  Dwell_Time_120 = 38.4425208
  IC_Line_Length_30 = 1100
  IC_Line_Length_120 = 275
  Scan_Line_Length_30 = 6320
  Scan_Line_Length_120 = 1580
  Filter_Frequency_30 = 52.02
  Filter_Frequency_120 = 13.005
  IFOV_B1234 = 42.5000
  IFOV_B57_along_scan = 42.5
  IFOV_B57_across_scan = 42.5
  IFOV_B6 = 170.0
  Scan_Period = 142.922000
  Scan_Frequency = 6.9968
  Active_Scan_Time = 60743.013
  Turn_Around_Time = 10.719
END_GROUP = SCANNER_PARAMETERS

GROUP = SPACECRAFT_PARAMETERS
  ADS_Interval = 2.0
  ADS_Roll_Offset = 0.375
  ADS_Pitch_Offset = 0.875
  ADS_Yaw_Offset = 1.375
  Data_Rate = 84.903
END_GROUP = SPACECRAFT_PARAMETERS

GROUP = MIRROR_PARAMETERS
  Error_Conversion_Factor = 0.18845000
  GROUP = ANGLES_SME1_SAM
    Forward_Along_SME1_SAM = (+0.0000E0, -2.0846E-3, +2.4365E-1, -1.1042E1, +2.1349E2, -1.4560E3)
    Forward_Cross_SME1_SAM = (+0.0000E0, -1.2639E-4, +3.5312E-3, -4.8660E-2, +5.4476E-1, -2.2077E0)
    Forward_Angle1_SME1_SAM = 67171.0
    Forward_Angle2_SME1_SAM = 67159.0
    Reverse_Along_SME1_SAM = (+0.0000E0, +2.5179E-3, -3.0669E-1, +1.3025E1, -2.3212E2, +1.4747E3)
    Reverse_Cross_SME1_SAM = (+0.0000E0, -9.9308E-5, +2.6935E-3, -6.8859E-2, +1.4509E0, -9.9468E0)
    Reverse_Angle1_SME1_SAM = 67159.0
    Reverse_Angle2_SME1_SAM = 67171.0
  END_GROUP = ANGLES_SME1_SAM
  GROUP = ANGLES_SME2_SAM
    Forward_Along_SME2_SAM = (+0.0000E0, -1.6484E-3, +2.4464E-1, -1.1422E1, +2.1987E2, -1.4945E3)
    Forward_Cross_SME2_SAM = (+0.0000E0, -1.2101E-4, +2.9221E-3, -2.9348E-2, +3.3941E-1, -1.7827E0)
    Forward_Angle1_SME2_SAM = 67182.0
    Forward_Angle2_SME2_SAM = 67160.0
    Reverse_Along_SME2_SAM = (+0.0000E0, +3.1143E-3, -3.2331E-1, +1.3313E1, -2.3650E2, +1.4991E3)
    Reverse_Cross_SME2_SAM = (+0.0000E0, -9.0740E-5, +1.5799E-3, -1.3242E-2, +2.9615E-1, -1.6706E0)
    Reverse_Angle1_SME2_SAM = 67160.0
    Reverse_Angle2_SME2_SAM = 67182.0
  END_GROUP = ANGLES_SME2_SAM
  GROUP = ANGLES_SME1_BUMP
    Forward_Along_SME1_Bump = (1.251220E-11, -9.068689E-03, 4.031291E-01, -1.339203E+01, 2.606205E+02, -1.793570E+03)
    Forward_Cross_SME1_Bump = (3.714633E-05, -3.501001E-04, -1.098810E-02, 4.363837E-01, -4.996861E+00, 1.733860E+01)
    Forward_Angle1_SME1_Bump =
(68060.7,68060.7,68060.3,68060.2,68060.2,68060.1,68060.1,68060.0,68060.0,68059.9,68059.8,68059.9,68059.8,68059.5,68059.5
,68059.5,68059.5,68059.5,68059.5,68059.0,68059.0,68058.9,68059.0,68058.8,68058.7,68058.5,68058.3,68058.3,68058.2,68058.1
,68058.1,68058.0,68057.9,68057.8,68058.0,68058.0,68057.8,68057.7,68057.7,68057.6,68057.7,68057.6,68057.4,68056.9,68056.9
,68056.9,68056.8,68056.8,68056.8,68056.8,68056.9,68056.8,68056.8,68056.8,68056.8,68056.7,68056.7,68056.7,68056.6,68056.6
,68056.6,68056.6,68056.7,68056.7,68056.7,68056.7,68056.7,68056.7,68056.8,68056.8,68056.9,68057.0,68057.0,68057.1,68057.1
,68057.5,68057.6,68057.6,68057.7,68058.0,68058.2,68058.3,68058.2,68058.3,68058.6,68058.7,68058.8,68059.0,68059.1,68059.2
,68059.4,68059.5)
    Forward_Angle2_SME1_Bump =
(70088.3,70088.7,70088.3,70088.8,70088.9,70089.3,70089.7,70090.1,70090.3,70090.5,70090.7,70091.1,70091.3,70091.5,70091.6
,70091.7,70091.8,70091.9,70091.9,70093.3,70093.6,70093.7,70093.9,70094.0,70094.2,70094.3,70094.5,70094.4,70094.9,70095.4
,70095.9,70096.4,70096.9,70097.4,70097.4,70097.9,70098.0,70098.5,70099.1,70099.6,70100.3,70100.7,70101.0,70102.1,70102.2
,70102.3,70102.4,70102.4,70102.5,70102.6,70102.9,70103.2,70103.4,70103.6,70103.8,70104.0,70104.3,70104.5,70104.7,70104.6
,70105.0,70105.3,70105.7,70106.0,70106.4,70106.7,70106.4,70106.6,70106.9,70107.1,70107.5,70107.8,70108.2,70108.5,70108.8
,70109.3,70109.5,70109.8,70110.6,70111.0,70111.3,70111.5,70111.9,70112.3,70112.4,70112.8,70113.2,70113.6,70114.0,70114.4
,70114.8,70114.7)
    Forward_FHSERR_SME1_Bump =
(2147,2147,2147,2148,2148,2148,2149,2149,2150,2150,2150,2151,2151,2152,2152,2152,2152,2152,2152,2154,2155,2155,2155,2
155,2156,2156,2156,2156,2157,2158,2158,2159,2160,2160,2160,2161,2161,2162,2162,2163,2164,2164,2165,2167,2167,2167,21
67,2167,2167,2168,2168,2168,2168,2169,2169,2169,2170,2170,2170,2170,2171,2171,2171,2172,2172,2172,2172,2172,2173,217
3,2173,2173,2174,2174,2174,2174,2175,2175,2176,2176,2176,2176,2177,2177,2177,2177,2177,2178,2178,2178,2178,2178)
    Forward_SHSERR_SME1_Bump = (-2101,-2101,-2101,-2102,-2102,-2103,-2103,-2104,-2104,-2104,-2105,-2105,-2105,-2106,-
2106,-2106,-2106,-2106,-2106,-2108,-2109,-2109,-2109,-2109,-2110,-2110,-2111,-2111,-2111,-2112,-2113,-2113,-2114,-2115,-
2114,-2115,-2115,-2116,-2117,-2117,-2118,-2119,-2119,-2121,-2121,-2121,-2121,-2122,-2122,-2122,-2122,-2122,-2123,-2123,-
2123,-2124,-2124,-2124,-2124,-2124,-2125,-2125,-2125,-2126,-2126,-2127,-2126,-2127,-2127,-2127,-2127,-2128,-2128,-2128,-
2129,-2129,-2129,-2129,-2130,-2130,-2130,-2130,-2131,-2131,-2131,-2131,-2132,-2132,-2132,-2132,-2133,-2132)
    Reverse_Along_SME1_Bump = (-7.338293E-12, -2.992738E-03, -2.619761E-01, 1.526341E+01, -2.732670E+02,
1.750636E+03)
    Reverse_Cross_SME1_Bump = (1.779879E-05, 3.316772E-04, -1.641136E-02, 6.902699E-01, -9.899752E+00, 5.043054E+01)
    Reverse_Angle1_SME1_Bump =
(68348.9,68349.1,68348.4,68348.5,68348.5,68348.5,68348.6,68348.7,68348.6,68348.6,68348.5,68348.8,68348.7,68348.7,68348.5
,68348.4,68348.3,68348.2,68348.1,68348.8,68348.9,68348.9,68348.8,68348.7,68348.6,68348.5,68348.5,68347.8,68348.0,68348.2
,68348.4,68348.5,68348.7,68348.9,68349.0,68349.2,68349.3,68349.4,68349.5,68349.7,68350.1,68350.1,68350.0,68350.5,68350.6
,68350.6,68350.6,68350.6,68350.7,68350.7,68350.8,68350.8,68350.8,68350.8,68350.8,68350.8,68350.8,68350.8,68350.8,68351.0
,68351.0,68351.1,68351.2,68351.3,68351.3,68351.4,68351.6,68351.8,68352.0,68352.2,68352.4,68352.6,68352.9,68353.1,68353.2
,68354.4,68354.7,68355.0,68354.8,68355.3,68355.7,68356.1,68355.4,68355.8,68356.3,68356.9,68357.4,68357.9,68358.9,68359.3
,68359.8,68360.4)
    Reverse_Angle2_SME1_Bump =
(69377.0,69377.2,69377.2,69377.3,69377.5,69377.7,69377.8,69378.0,69378.1,69378.1,69378.2,69378.6,69378.7,69378.6,69378.8
,69378.9,69379.0,69379.2,69379.3,69379.2,69379.4,69379.5,69379.7,69379.8,69379.8,69379.8,69379.9,69379.8,69379.9,69380.0
,69380.1,69380.2,69380.3,69380.3,69380.4,69380.6,69380.7,69380.8,69380.9,69381.0,69381.4,69381.4,69381.5,69381.0,69381.1
,69381.2,69381.3,69381.4,69381.5,69381.5,69381.6,69381.7,69381.9,69382.0,69382.2,69382.3,69382.5,69382.6,69382.8,69382.7
,69382.9,69383.1,69383.3,69383.6,69383.8,69384.0,69383.7,69383.9,69384.0,69384.1,69384.3,69384.5,69384.7,69384.8,69385.0
,69385.8,69385.9,69385.9,69386.3,69386.6,69386.8,69387.0,69387.0,69387.1,69387.4,69387.6,69387.8,69388.0,69388.1,69388.3
,69388.6,69388.6)
    Reverse_FHSERR_SME1_Bump =
(1442,1442,1443,1443,1443,1444,1444,1444,1444,1444,1444,1444,1444,1445,1445,1445,1445,1446,1446,1445,1445,1445,1446,1
446,1446,1446,1446,1447,1447,1447,1447,1446,1446,1446,1446,1446,1446,1446,1446,1446,1446,1446,1446,1445,1445,1445,14
45,1445,1445,1446,1445,1446,1446,1446,1446,1446,1446,1447,1447,1446,1447,1447,1447,1447,1447,1448,1447,1447,1447,144
7,1447,1447,1447,1447,1446,1446,1446,1446,1446,1446,1446,1445,1446,1446,1446,1445,1445,1444,1443,1443,1443,1442)
    Reverse_SHSERR_SME1_Bump = (-1397,-1397,-1397,-1397,-1398,-1398,-1398,-1398,-1398,-1398,-1398,-1398,-1399,-1399,-
1399,-1399,-1400,-1400,-1400,-1399,-1399,-1400,-1400,-1400,-1400,-1400,-1400,-1401,-1401,-1401,-1401,-1401,-1401,-1400,-
1400,-1400,-1400,-1400,-1400,-1400,-1400,-1400,-1400,-1399,-1399,-1399,-1400,-1400,-1400,-1400,-1400,-1400,-1400,-1400,-
1400,-1401,-1401,-1401,-1401,-1401,-1401,-1401,-1401,-1401,-1402,-1402,-1401,-1401,-1401,-1401,-1401,-1401,-1401,-1401,-
1401,-1400,-1400,-1400,-1400,-1400,-1400,-1400,-1400,-1400,-1400,-1399,-1399,-1399,-1398,-1397,-1397,-1396)
  END_GROUP = ANGLES_SME1_BUMP
  GROUP = ANGLES_SME2_BUMP
    Forward_Along_SME2_Bump = (1.251220E-11, -9.068689E-03, 4.031291E-01, -1.339203E+01, 2.606205E+02, -1.793570E+03)
    Forward_Cross_SME2_Bump = (3.714633E-05, -3.501001E-04, -1.098810E-02, 4.363837E-01, -4.996861E+00, 1.733860E+01)
    Forward_Angle1_SME2_Bump =
(68060.7,68060.7,68060.3,68060.2,68060.2,68060.1,68060.1,68060.0,68060.0,68059.9,68059.8,68059.9,68059.8,68059.5,68059.5
,68059.5,68059.5,68059.5,68059.5,68059.0,68059.0,68058.9,68059.0,68058.8,68058.7,68058.5,68058.3,68058.3,68058.2,68058.1
```

```
,68058.1,68058.0,68057.9,68057.8,68058.0,68058.0,68057.8,68057.7,68057.7,68057.6,68057.7,68057.6,68057.4,68056.9,68056.9
,68056.9,68056.8,68056.8,68056.8,68056.8,68056.9,68056.8,68056.8,68056.8,68056.8,68056.7,68056.7,68056.7,68056.6,68056.6
,68056.6,68056.6,68056.7,68056.7,68056.7,68056.7,68056.7,68056.7,68056.8,68056.8,68056.9,68057.0,68057.0,68057.1,68057.1
,68057.5,68057.6,68057.6,68057.7,68058.0,68058.2,68058.3,68058.2,68058.3,68058.6,68058.7,68058.8,68059.0,68059.1,68059.2
,68059.4,68059.5)
   Forward_Angle2_SME2_Bump =
(70088.3,70088.7,70088.3,70088.8,70088.9,70089.3,70089.7,70090.1,70090.3,70090.5,70090.7,70091.1,70091.3,70091.5,70091.6
,70091.7,70091.8,70091.9,70091.9,70093.3,70093.6,70093.7,70093.9,70094.0,70094.2,70094.3,70094.5,70094.4,70094.9,70095.4
,70095.9,70096.4,70096.9,70097.4,70097.4,70097.9,70098.0,70098.5,70099.1,70099.6,70100.3,70100.7,70101.0,70102.1,70102.2
,70102.3,70102.4,70102.4,70102.5,70102.6,70102.9,70103.2,70103.4,70103.6,70103.8,70104.0,70104.3,70104.5,70104.7,70104.6
,70105.0,70105.3,70105.7,70106.0,70106.4,70106.7,70106.4,70106.6,70106.9,70107.1,70107.5,70107.8,70108.2,70108.5,70108.8
,70109.3,70109.5,70109.8,70110.6,70111.0,70111.3,70111.5,70111.9,70112.3,70112.4,70112.8,70113.2,70113.6,70114.0,70114.4
,70114.8,70114.7)
   Forward_FHSERR_SME2_Bump =
(2147,2147,2147,2148,2148,2148,2149,2149,2150,2150,2150,2151,2151,2152,2152,2152,2152,2152,2152,2154,2155,2155,2155,2
155,2156,2156,2156,2156,2157,2158,2158,2159,2160,2160,2160,2161,2161,2162,2162,2163,2164,2164,2165,2167,2167,2167,21
67,2167,2167,2168,2168,2168,2168,2169,2169,2169,2170,2170,2170,2170,2171,2171,2171,2172,2172,2172,2172,2172,2173,217
3,2173,2173,2174,2174,2174,2174,2175,2175,2176,2176,2176,2176,2177,2177,2177,2177,2177,2178,2178,2178,2178,2178)
   Forward_SHSERR_SME2_Bump = (-2101,-2101,-2101,-2102,-2102,-2103,-2103,-2104,-2104,-2104,-2105,-2105,-2105,-2106,-
2106,-2106,-2106,-2106,-2106,-2108,-2109,-2109,-2109,-2109,-2110,-2110,-2111,-2111,-2111,-2112,-2113,-2113,-2114,-2115,-
2114,-2115,-2115,-2116,-2117,-2117,-2118,-2119,-2119,-2121,-2121,-2121,-2121,-2122,-2122,-2122,-2122,-2122,-2123,-2123,-
2123,-2124,-2124,-2124,-2124,-2124,-2125,-2125,-2125,-2126,-2126,-2127,-2126,-2127,-2127,-2127,-2127,-2128,-2128,-2128,-
2129,-2129,-2129,-2129,-2130,-2130,-2130,-2130,-2131,-2131,-2131,-2131,-2132,-2132,-2132,-2132,-2133,-2132)
   Reverse_Along_SME2_Bump = (-8.034324E-12,-3.029990E-03,-2.581620E-01,1.511157E+01,-2.706933E+02,1.735135E+03)
   Reverse_Cross_SME2_Bump = (1.783806E-05,3.467927E-04,-1.882727E-02,8.256714E-01,-1.270779E+01,6.979331E+01)
   Reverse_Angle1_SME2_Bump =
(68348.9,68349.1,68348.4,68348.5,68348.5,68348.5,68348.6,68348.7,68348.6,68348.6,68348.5,68348.8,68348.7,68348.7,68348.5
,68348.4,68348.3,68348.2,68348.1,68348.8,68348.9,68348.9,68348.8,68348.7,68348.6,68348.5,68348.5,68347.8,68348.0,68348.2
,68348.4,68348.5,68348.7,68348.9,68349.0,68349.2,68349.3,68349.4,68349.5,68349.7,68350.1,68350.1,68350.0,68350.5,68350.6
,68350.6,68350.6,68350.6,68350.7,68350.7,68350.8,68350.8,68350.8,68350.8,68350.8,68350.8,68350.8,68350.8,68350.8,68351.0
,68351.0,68351.1,68351.2,68351.3,68351.3,68351.4,68351.6,68351.8,68352.0,68352.2,68352.4,68352.6,68352.9,68353.1,68353.2
,68354.4,68354.7,68355.0,68354.8,68355.3,68355.7,68356.1,68355.4,68355.8,68356.3,68356.9,68357.4,68357.9,68358.9,68359.3
,68359.8,68360.4)
   Reverse_Angle2_SME2_Bump =
(69377.0,69377.2,69377.2,69377.3,69377.5,69377.7,69377.8,69378.0,69378.1,69378.1,69378.2,69378.6,69378.7,69378.6,69378.8
,69378.9,69379.0,69379.2,69379.3,69379.2,69379.4,69379.5,69379.7,69379.8,69379.8,69379.8,69379.9,69379.8,69379.9,69380.0
,69380.1,69380.2,69380.3,69380.3,69380.4,69380.6,69380.7,69380.8,69380.9,69381.0,69381.4,69381.4,69381.5,69381.0,69381.1
,69381.2,69381.3,69381.4,69381.5,69381.5,69381.6,69381.7,69381.9,69382.0,69382.2,69382.3,69382.5,69382.6,69382.8,69382.7
,69382.9,69383.1,69383.3,69383.6,69383.8,69384.0,69383.7,69383.9,69384.0,69384.1,69384.3,69384.5,69384.7,69384.8,69385.0
,69385.8,69385.9,69385.9,69386.3,69386.6,69386.8,69387.0,69387.0,69387.1,69387.4,69387.6,69387.8,69388.0,69388.1,69388.3
,69388.6,69388.6)
   Reverse_FHSERR_SME2_Bump =
(1442,1442,1443,1443,1443,1444,1444,1444,1444,1444,1444,1444,1444,1445,1445,1445,1445,1446,1446,1445,1445,1445,1446,1
446,1446,1446,1446,1447,1447,1447,1447,1446,1446,1446,1446,1446,1446,1446,1446,1446,1446,1446,1446,1445,1445,1445,14
45,1445,1445,1446,1445,1446,1446,1446,1446,1446,1446,1447,1447,1446,1447,1447,1447,1447,1447,1448,1447,1447,1447,144
7,1447,1447,1447,1447,1446,1446,1446,1446,1446,1446,1446,1445,1446,1446,1446,1445,1445,1444,1443,1443,1443,1442)
   Reverse_SHSERR_SME2_Bump = (-1397,-1397,-1397,-1397,-1398,-1398,-1398,-1398,-1398,-1398,-1398,-1398,-1399,-1399,-
1399,-1399,-1400,-1400,-1400,-1399,-1399,-1400,-1400,-1400,-1400,-1400,-1400,-1401,-1401,-1401,-1401,-1401,-1401,-1400,-
1400,-1400,-1400,-1400,-1400,-1400,-1400,-1400,-1399,-1399,-1399,-1400,-1400,-1400,-1400,-1400,-1400,-1400,-1400,-
1400,-1401,-1401,-1401,-1401,-1401,-1401,-1401,-1401,-1401,-1402,-1402,-1401,-1401,-1401,-1401,-1401,-1401,-1401,-1401,-
1401,-1400,-1400,-1400,-1400,-1400,-1400,-1400,-1400,-1400,-1400,-1399,-1399,-1399,-1398,-1397,-1397,-1396)
  END_GROUP = ANGLES_SME2_BUMP
END_GROUP = MIRROR_PARAMETERS

GROUP = BUMPER_MODE_PARAMETERS
  SME1_BumperA_Dwell_Time =
(9970.61,9970.71,9970.37,9970.45,9970.41,9970.44,9970.47,9970.50,9970.50,9970.47,9970.44,9970.58,9970.54,9970.50,9970.45
,9970.40,9970.35,9970.31,9970.25,9970.57,9970.62,9970.59,9970.57,9970.53,9970.49,9970.45,9970.41,9970.12,9970.20,9970.29
,9970.37,9970.45,9970.54,9970.62,9970.65,9970.74,9970.77,9970.83,9970.89,9970.94,9971.13,9971.12,9971.11,9971.34,9971.35
,9971.36,9971.37,9971.38,9971.39,9971.40,9971.44,9971.44,9971.44,9971.44,9971.44,9971.44,9971.44,9971.44,9971.44,9971.53
,9971.56,9971.59,9971.62,9971.65,9971.68,9971.71,9971.81,9971.90,9971.99,9972.07,9972.16,9972.24,9972.37,9972.45,9972.53
,9973.06,9973.20,9973.32,9973.23,9973.42,9973.60,9973.78,9973.49,9973.68,9973.90,9974.13,9974.36,9974.60,9975.03,9975.23
,9975.42,9975.71)
  SME1_BumperA_Pickoff_Time =
(1124.01,1124.19,1124.00,1124.21,1124.28,1124.45,1124.63,1124.80,1124.87,1124.96,1125.05,1125.23,1125.28,1125.39,1125.43
,1125.47,1125.51,1125.55,1125.58,1126.17,1126.31,1126.37,1126.43,1126.49,1126.56,1126.62,1126.69,1126.68,1126.89,1127.11
,1127.32,1127.54,1127.75,1127.97,1127.96,1128.19,1128.25,1128.47,1128.69,1128.91,1129.24,1129.39,1129.54,1130.02,1130.05
,1130.09,1130.13,1130.16,1130.20,1130.24,1130.38,1130.47,1130.57,1130.67,1130.76,1130.86,1130.96,1131.05,1131.15,1131.12
,1131.27,1131.42,1131.57,1131.73,1131.88,1132.03,1131.87,1131.99,1132.10,1132.21,1132.37,1132.52,1132.66,1132.80,1132.95
```

```
,1133.16,1133.25,1133.38,1133.70,1133.90,1133.99,1134.09,1134.29,1134.47,1134.49,1134.65,1134.82,1134.99,1135.20,1135.36
,1135.52,1135.48)
  SME1_BumperA_Offset_Time = 10110.00
  SME1_BumperA_Angle = -68665.0
  SME1_BumperB_Dwell_Time =
(9869.81,9869.84,9869.65,9869.62,9869.60,9869.58,9869.55,9869.52,9869.50,9869.46,9869.42,9869.47,9869.44,9869.32,9869.32
,9869.32,9869.32,9869.31,9869.31,9869.10,9869.10,9869.07,9869.09,9869.02,9868.95,9868.87,9868.80,9868.78,9868.75,9868.72
,9868.68,9868.65,9868.61,9868.58,9868.67,9868.64,9868.57,9868.54,9868.51,9868.48,9868.51,9868.46,9868.42,9868.19,9868.18
,9868.17,9868.15,9868.14,9868.13,9868.11,9868.16,9868.15,9868.14,9868.13,9868.12,9868.10,9868.09,9868.08,9868.07,9868.03
,9868.05,9868.06,9868.08,9868.09,9868.10,9868.12,9868.07,9868.10,9868.12,9868.15,9868.19,9868.21,9868.24,9868.27,9868.30
,9868.45,9868.51,9868.50,9868.56,9868.68,9868.76,9868.83,9868.77,9868.82,9868.93,9868.99,9869.05,9869.12,9869.17,9869.23
,9869.30,9869.35)
  SME1_BumperB_Pickoff_Time =
(837.81,837.93,837.89,837.98,838.05,838.12,838.18,838.25,838.30,838.33,838.36,838.51,838.55,838.53,838.60,838.66,838.72,83
8.78,838.84,838.78,838.88,838.93,839.02,839.04,839.05,839.07,839.08,839.06,839.10,839.14,839.18,839.22,839.25,839.29,839.3
4,839.38,839.43,839.47,839.51,839.56,839.76,839.77,839.78,839.59,839.63,839.67,839.70,839.74,839.78,839.81,839.82,839.89,8
39.95,840.02,840.08,840.15,840.22,840.28,840.35,840.30,840.40,840.50,840.60,840.69,840.79,840.89,840.76,840.82,840.88,840.
94,841.04,841.10,841.18,841.24,841.31,841.67,841.73,841.73,841.89,842.02,842.10,842.19,842.17,842.25,842.35,842.44,842.53,
842.62,842.67,842.77,842.87,842.89)
  SME1_BumperB_Offset_Time = 10110.00
  SME1_BumperB_Angle = 68607.0
  SME2_BumperA_Dwell_Time =
(9970.61,9970.71,9970.37,9970.45,9970.41,9970.44,9970.47,9970.50,9970.50,9970.47,9970.44,9970.58,9970.54,9970.50,9970.45
,9970.40,9970.35,9970.31,9970.25,9970.57,9970.62,9970.59,9970.57,9970.53,9970.49,9970.45,9970.41,9970.12,9970.20,9970.29
,9970.37,9970.45,9970.54,9970.62,9970.65,9970.74,9970.77,9970.83,9970.89,9970.94,9971.13,9971.12,9971.11,9971.34,9971.35
,9971.36,9971.37,9971.38,9971.39,9971.40,9971.44,9971.44,9971.44,9971.44,9971.44,9971.44,9971.44,9971.44,9971.44,9971.53
,9971.56,9971.59,9971.62,9971.65,9971.68,9971.71,9971.81,9971.90,9971.99,9972.07,9972.16,9972.24,9972.37,9972.45,9972.53
,9973.06,9973.20,9973.32,9973.23,9973.42,9973.60,9973.78,9973.49,9973.68,9973.90,9974.13,9974.36,9974.60,9975.03,9975.23
,9975.42,9975.71)
  SME2_BumperA_Pickoff_Time =
(1124.01,1124.19,1124.00,1124.21,1124.28,1124.45,1124.63,1124.80,1124.87,1124.96,1125.05,1125.23,1125.28,1125.39,1125.43
,1125.47,1125.51,1125.55,1125.58,1126.17,1126.31,1126.37,1126.43,1126.49,1126.56,1126.62,1126.69,1126.68,1126.89,1127.11
,1127.32,1127.54,1127.75,1127.97,1127.96,1128.19,1128.25,1128.47,1128.69,1128.91,1129.24,1129.39,1129.54,1130.02,1130.05
,1130.09,1130.13,1130.16,1130.20,1130.24,1130.38,1130.47,1130.57,1130.67,1130.76,1130.86,1130.96,1131.05,1131.15,1131.12
,1131.27,1131.42,1131.57,1131.73,1131.88,1132.03,1131.87,1131.99,1132.10,1132.21,1132.37,1132.52,1132.66,1132.80,1132.95
,1133.16,1133.25,1133.38,1133.70,1133.90,1133.99,1134.09,1134.29,1134.47,1134.49,1134.65,1134.82,1134.99,1135.20,1135.36
,1135.52,1135.48)
  SME2_BumperA_Offset_Time = 10110.00
  SME2_BumperA_Angle = -68665.0
  SME2_BumperB_Dwell_Time =
(9869.81,9869.84,9869.65,9869.62,9869.60,9869.58,9869.55,9869.52,9869.50,9869.46,9869.42,9869.47,9869.44,9869.32,9869.32
,9869.32,9869.32,9869.31,9869.31,9869.10,9869.10,9869.07,9869.09,9869.02,9868.95,9868.87,9868.80,9868.78,9868.75,9868.72
,9868.68,9868.65,9868.61,9868.58,9868.67,9868.64,9868.57,9868.54,9868.51,9868.48,9868.51,9868.46,9868.42,9868.19,9868.18
,9868.17,9868.15,9868.14,9868.13,9868.11,9868.16,9868.15,9868.14,9868.13,9868.12,9868.10,9868.09,9868.08,9868.07,9868.03
,9868.05,9868.06,9868.08,9868.09,9868.10,9868.12,9868.07,9868.10,9868.12,9868.15,9868.19,9868.21,9868.24,9868.27,9868.30
,9868.45,9868.51,9868.50,9868.56,9868.68,9868.76,9868.83,9868.77,9868.82,9868.93,9868.99,9869.05,9869.12,9869.17,9869.23
,9869.30,9869.35)
  SME2_BumperB_Pickoff_Time =
(837.81,837.93,837.89,837.98,838.05,838.12,838.18,838.25,838.30,838.33,838.36,838.51,838.55,838.53,838.60,838.66,838.72,83
8.78,838.84,838.78,838.88,838.93,839.02,839.04,839.05,839.07,839.08,839.06,839.10,839.14,839.18,839.22,839.25,839.29,839.3
4,839.38,839.43,839.47,839.51,839.56,839.76,839.77,839.78,839.59,839.63,839.67,839.70,839.74,839.78,839.81,839.82,839.89,8
39.95,840.02,840.08,840.15,840.22,840.28,840.35,840.30,840.40,840.50,840.60,840.69,840.79,840.89,840.76,840.82,840.88,840.
94,841.04,841.10,841.18,841.24,841.31,841.67,841.73,841.73,841.89,842.02,842.10,842.19,842.17,842.25,842.35,842.44,842.53,
842.62,842.67,842.77,842.87,842.89)
  SME2_BumperB_Offset_Time = 10110.00
  SME2_BumperB_Angle = 68607.0
END_GROUP = BUMPER_MODE_PARAMETERS

GROUP = SCAN_LINE_CORRECTOR
  Primary_Angular_Velocity = 0.00966
  Secondary_Angular_Velocity = 0.00960
  Primary_Corrector_Motion = (0.00000,0.00000,0.00000,0.00000,0.00000,0.00000)
  Secondary_Corrector_Motion = (0.00000,0.00000,0.00000,0.00000,0.00000,0.00000)
  Unpowered_Pointing_Bias = 0.0000000
END_GROUP = SCAN_LINE_CORRECTOR

GROUP = FOCAL_PLANE_PARAMETERS
  GROUP = BAND_OFFSETS
    Along_Scan_Band_Offsets = (+3628.958, +2566.458, +1503.958, +441.458, -2576.042, -4052.917, -1471.042)
    Across_Scan_Band_Offsets = (+1.700, +1.700, +0.425, +0.000, +4.675, +5.950, +5.100)
    Forward_Focal_Plane_Offsets = (+25.0,+50.0,+75.0,+100.0,+171.0,+208.0,+145.0)
    Reverse_Focal_Plane_Offsets = (-25.0,-50.0,-75.0,-100.0,-171.0,-211.0,-145.0)
  END_GROUP = BAND_OFFSETS
  GROUP = DETECTOR_OFFSETS
    Forward_Along_Scan_DO_B1 = (1.460, 1.470, 1.450, 1.465, 1.495, 1.425, 1.455, 1.415, 1.450, 1.425, 1.450, 1.455, 1.495,
1.430, 1.515, 1.455)
    Reverse_Along_Scan_DO_B1 = (1.075, 1.085, 1.030, 1.090, 1.055, 1.045, 1.005, 1.035, 1.015, 1.045, 1.015, 1.060, 1.065,
1.040, 1.070, 1.050)
    Forward_Along_Scan_DO_B2 = (1.175, 1.235, 1.175, 1.185, 1.215, 1.210, 1.185, 1.185, 1.175, 1.190, 1.155, 1.200, 1.195,
1.240, 1.195, 1.195)
    Reverse_Along_Scan_DO_B2 = (1.320, 1.390, 1.310, 1.350, 1.345, 1.365, 1.315, 1.335, 1.290, 1.345, 1.280, 1.345, 1.310,
1.380, 1.305, 1.330)
    Forward_Along_Scan_DO_B3 = (1.205, 1.210, 1.215, 1.210, 1.185, 1.175, 1.220, 1.160, 1.185, 1.220, 1.225, 1.220, 1.270,
1.205, 1.260, 1.205)
    Reverse_Along_Scan_DO_B3 = (1.320, 1.370, 1.330, 1.360, 1.290, 1.335, 1.335, 1.315, 1.290, 1.345, 1.325, 1.340, 1.375,
1.355, 1.355, 1.350)
    Forward_Along_Scan_DO_B4 = (1.260, 1.245, 1.260, 1.250, 1.250, 1.195, 1.210, 1.225, 1.225, 1.205, 1.225, 1.205, 1.250,
1.230, 1.285, 1.255)
    Reverse_Along_Scan_DO_B4 = (1.285, 1.290, 1.290, 1.295, 1.265, 1.235, 1.275, 1.260, 1.225, 1.275, 1.235, 1.235, 1.255,
1.255, 1.285, 1.280)
    Forward_Along_Scan_DO_B5 = (1.520, 1.330, 1.485, 1.350, 1.380, 1.420, 1.395, 1.415, 1.425, 1.435, 1.395, 1.350, 1.395,
1.370, 1.355, 1.360)
    Reverse_Along_Scan_DO_B5 = (1.235, 1.295, 1.255, 1.310, 1.170, 1.385, 1.195, 1.385, 1.235, 1.415, 1.230, 1.355, 1.275,
1.400, 1.175, 1.435)
    Forward_Along_Scan_DO_B6 = (4.900, 4.900, 4.900, 4.900)
    Reverse_Along_Scan_DO_B6 = (4.900, 4.900, 4.900, 4.900)
    Forward_Along_Scan_DO_B7 = (1.189, 1.055, 1.130, 1.115, 1.150, 1.100, 1.170, 1.125, 1.220, 1.100, 1.180, 1.035, 1.115,
1.040, 1.125, 1.075)
    Reverse_Along_Scan_DO_B7 = (1.395, 1.535, 1.390, 1.259, 1.440, 1.259, 1.250, 1.575, 1.545, 1.605, 1.475, 1.555, 1.425,
1.590, 1.450, 1.670)
    Forward_Across_Scan_DO_B1 = (0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000)
    Reverse_Across_Scan_DO_B1 = (0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000)
    Forward_Across_Scan_DO_B2 = (0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000)
    Reverse_Across_Scan_DO_B2 = (0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000)
    Forward_Across_Scan_DO_B3 = (0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000)
    Reverse_Across_Scan_DO_B3 = (0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000)
    Forward_Across_Scan_DO_B4 = (0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000)
    Reverse_Across_Scan_DO_B4 = (0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000)
    Forward_Across_Scan_DO_B5 = (0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000)
    Reverse_Across_Scan_DO_B5 = (0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000)
    Forward_Across_Scan_DO_B6 = (0.000, 0.000, 0.000, 0.000)
    Reverse_Across_Scan_DO_B6 = (0.000, 0.000, 0.000, 0.000)
    Forward_Across_Scan_DO_B7 = (0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000)
    Reverse_Across_Scan_DO_B7 = (0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000)
  END_GROUP = DETECTOR_OFFSETS
  GROUP = ODD_EVEN_OFFSETS
    Forward_Even_Detector_Shift = (51.0, 76.0, 101.0, 126.0, 197.0, 57.0, 171.0)
    Forward_Odd_Detector_Shift = (53.0, 78.0, 103.0, 128.0, 199.0, 59.0, 173.0)
    Reverse_Even_Detector_Shift = (43.0, 68.0, 93.0, 118.0, 189.0, 55.0, 163.0)
    Reverse_Odd_Detector_Shift = (46.0, 71.0, 96.0, 121.0, 192.0, 58.0, 166.0)
  END_GROUP = ODD_EVEN_OFFSETS
END_GROUP = FOCAL_PLANE_PARAMETERS

GROUP = ATTITUDE_PARAMETERS
  Gyro_To_Attitude_Matrix = (+1.0000E0, +8.9880E-5, +1.7320E-5, +5.8319E-4, +0.9999998E0, +7.7871E-4, -1.5694E-4, -
5.1692E-4, +0.9999998E0)
  ADSA_To_TM_Matrix = (+1.0000E0, +0.0000E0, +0.0000E0, +0.0000E0, +9.396926E-1, -3.420201E-1, +0.0000E0, +3.420201E-
1, +9.396926E-1)
```

```
  Attitude_To_TM_Matrix = (+9.99999851E-01, +5.45640973E-04, -9.00000000E-06, -5.45647569E-04, +9.99999570E-01,
-7.49999930E-04, +8.59076544E-06, +7.50004729E-04, +9.99999719E-01)
  Spacecraft_Roll_Bias = 0.0000000
  Spacecraft_Pitch_Bias = 0.0000000
  Spacecraft_Yaw_Bias = 0.0000000
END_GROUP = ATTITUDE_PARAMETERS

GROUP = TIME_PARAMETERS
  Scan_Time = 60743.0
  Forward_First_Half_Time = 30371.4
  Forward_Second_Half_Time = 30371.6
  Reverse_First_Half_Time = 30371.6
  Reverse_Second_Half_Time = 30371.4
END_GROUP = TIME_PARAMETERS

GROUP = TRANSFER_FUNCTION
  GROUP = IMU
    Fn = 2.2010
    Zeta = 0.7022
    Tau = 11.4468E-3
    P = -3.2590E-3
    Ak = 1.00518
  END_GROUP = IMU
  GROUP = ADS
    ADS_num = (0.0000E0, 0.0000E0, 0.0000E0, +4.3830E6, +5.4890E5, -3.5290E2, 0.0000E0, 0.0000E0, 0.0000E0, +5.1110E6,
+5.6490E5, -3.7400E2, 0.0000E0, 0.0000E0, 0.0000E0, +4.5030E6, +5.5060E5, -3.5960E2)
    ADS_den = (+2.8470E8, +6.2750E8, +1.6550E8, +1.4240E7, +5.9530E5, +9.2030E2, +3.2140E8, +7.1220E8, +1.7910E8,
+1.2780E7, +6.0710E5, +9.5650E2, +4.3520E8, +6.1010E8, +1.5350E8, +1.1730E7, +6.0310E5, +9.4910E2)
    ADS_num_temp = (0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0,
0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0)
    ADS_den_temp = (0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0,
0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0)
  END_GROUP = ADS
  GROUP = PREFILTER
    ADSPre_W = (0.0, 0.0, 0.0, 0.0, 0.0)
    ADSPre_H = (0.0, 0.0, 0.0, 0.0, 0.0)
    ADSPre_T = (0.0, 0.0, 0.0, 0.0, 0.0)
  END_GROUP = PREFILTER
END_GROUP = TRANSFER_FUNCTION

GROUP = UT1_TIME_PARAMETERS
  UT1_Year =
(2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2
005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,20
05,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,200
5,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,
2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2
005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,20
05,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,200
5,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005,2005)
  UT1_Month =
("May","May","May","May","May","May","May","May","May","May","May","May","May","May","May","Jun","Jun","Jun","Jun","Jun","Jun
","Jun","Jun","Jun","Jun","Jun","Jun","Jun","Jun","Jun","Jun","Jun","Jun","Jun","Jun","Jun","Jun","Jun","Jun","Jun","Jun","Jun","Jun","
Jun","Jun","Jul","Jul","Jul","Jul","Jul","Jul","Jul","Jul","Jul","Jul","Jul","Jul","Jul","Jul","Jul","Jul","Jul","Jul","Jul","Jul","Jul","Jul","Jul","Ju
l","Jul","Jul","Jul","Jul","Jul","Jul","Jul","Aug","Aug","Aug","Aug","Aug","Aug","Aug","Aug","Aug","Aug","Aug","Aug","Aug","Aug","Aug",
"Aug","Aug","Aug","Aug","Aug","Aug","Aug","Aug","Aug","Aug","Aug","Aug","Aug","Aug","Aug","Aug","Sep","Sep","Sep","Sep","Sep","
Sep","Sep","Sep","Sep","Sep","Sep","Sep","Sep","Sep","Sep","Sep","Sep","Sep","Sep","Sep","Sep","Sep","Sep","Sep","Sep","Sep","
Sep","Sep","Sep","Sep","Oct","Oct","Oct","Oct","Oct","Oct","Oct","Oct","Oct","Oct","Oct","Oct","Oct","Oct","Oct","Oct","Oct","Oct","Oct
","Oct","Oct","Oct","Oct","Oct","Oct","Oct","Oct","Oct","Oct","Oct","Oct","Nov","Nov","Nov","Nov","Nov","Nov","Nov","Nov","Nov","Nov
","Nov","Nov")
  UT1_Day =
(17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,1
,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,1
9,20,21,22,23,24,25,26,27,28,29,30,31,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,1,2,3,4,5,
6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,1,2,3,4,5,6,7,8,9,10,11,12)
  UT1_Modified_Julian =
(53507,53508,53509,53510,53511,53512,53513,53514,53515,53516,53517,53518,53519,53520,53521,53522,53523,53524,53525,
53526,53527,53528,53529,53530,53531,53532,53533,53534,53535,53536,53537,53538,53539,53540,53541,53542,53543,53544,
53545,53546,53547,53548,53549,53550,53551,53552,53553,53554,53555,53556,53557,53558,53559,53560,53561,53562,53563,
53564,53565,53566,53567,53568,53569,53570,53571,53572,53573,53574,53575,53576,53577,53578,53579,53580,53581,53582,
53583,53584,53585,53586,53587,53588,53589,53590,53591,53592,53593,53594,53595,53596,53597,53598,53599,53600,53601,
53602,53603,53604,53605,53606,53607,53608,53609,53610,53611,53612,53613,53614,53615,53616,53617,53618,53619,53620,
53621,53622,53623,53624,53625,53626,53627,53628,53629,53630,53631,53632,53633,53634,53635,53636,53637,53638,53639,
53640,53641,53642,53643,53644,53645,53646,53647,53648,53649,53650,53651,53652,53653,53654,53655,53656,53657,53658,
53659,53660,53661,53662,53663,53664,53665,53666,53667,53668,53669,53670,53671,53672,53673,53674,53675,53676,53677,
53678,53679,53680,53681,53682,53683,53684,53685,53686)
  UT1_X = (-0.06747,-0.06832,-0.06926,-0.07003,-0.07063,-0.07101,-0.07126,-0.07141,-0.07133,-0.07042,-0.06902,-0.06806,-
0.06716,-0.06602,-0.06457,-0.06294,-0.06159,-0.06045,-0.05998,-0.05958,-0.05914,-0.05856,-0.05821,-0.05763,-0.05670,-
0.05554,-0.05441,-0.05333,-0.05253,-0.05206,-0.05144,-0.05056,-0.04954,-0.04834,-0.04718,-0.04625,-0.04576,-0.04513,-
0.04448,-0.04379,-0.04284,-0.04160,-0.04042,-0.03993,-0.03991,-0.04003,-0.03941,-0.03852,-0.03751,-0.03620,-0.03444,-
0.03279,-0.03132,-0.02991,-0.02862,-0.02764,-0.02662,-0.02563,-0.02440,-0.02306,-0.02171,-0.02051,-0.01973,-0.01915,-
0.01837,-0.01697,-0.01545,-0.01402,-0.01298,-0.01212,-0.01128,-0.01033,-0.00934,-0.00818,-0.00644,-0.00455,-0.00289,-
0.00122,0.00080,0.00296,0.00484,0.00664,0.00870,0.01110,0.01382,0.01626,0.01830,0.02003,0.02159,0.02282,0.02387,0.02489,
0.02604,0.02721,0.02845,0.02986,0.03132,0.03274,0.03426,0.03562,0.03700,0.03826,0.03945,0.04059,0.04146,0.04193,0.04192,
0.04201,0.04211,0.04233,0.04281,0.04361,0.04469,0.04601,0.04724,0.04850,0.04942,0.05010,0.05047,0.05040,0.05016,0.05018,
0.05092,0.05191,0.05255,0.05266,0.05266,0.05311,0.05364,0.05439,0.05535,0.05625,0.05679,0.05712,0.05784,0.05856,0.05877,
0.05872,0.05864,0.05899,0.05991,0.06090,0.06215,0.06326,0.06399,0.06452,0.06498,0.06514,0.06531,0.06573,0.06628,0.06698,
0.06796,0.06922,0.07052,0.07090,0.07100,0.07101,0.07101,0.07094,0.07090,0.07089,0.07057,0.07012,0.06983,0.06952,0.06946,
0.07014,0.07115,0.07229,0.07290,0.07307,0.07307,0.07291,0.07268,0.07256,0.07245,0.07222,0.07153,0.07082)
  UT1_Y =
(0.32492,0.32630,0.32766,0.32909,0.33054,0.33214,0.33382,0.33547,0.33721,0.33917,0.34142,0.34357,0.34569,0.34780,0.34958
,0.35094,0.35251,0.35438,0.35633,0.35811,0.35966,0.36102,0.36218,0.36326,0.36470,0.36671,0.36881,0.37094,0.37302,0.37478
,0.37628,0.37779,0.37934,0.38096,0.38261,0.38422,0.38583,0.38712,0.38819,0.38944,0.39060,0.39180,0.39329,0.39470,0.39594
,0.39679,0.39780,0.39878,0.39980,0.40113,0.40269,0.40445,0.40600,0.40725,0.40803,0.40865,0.40929,0.41012,0.41124,0.41253
,0.41358,0.41428,0.41478,0.41511,0.41525,0.41552,0.41618,0.41699,0.41783,0.41856,0.41908,0.41955,0.41992,0.42013,0.42036
,0.42074,0.42114,0.42140,0.42161,0.42186,0.42203,0.42245,0.42287,0.42337,0.42423,0.42520,0.42625,0.42712,0.42775,0.42833
,0.42885,0.42921,0.42933,0.42923,0.42898,0.42887,0.42891,0.42884,0.42850,0.42841,0.42829,0.42788,0.42692,0.42601,0.42551
,0.42522,0.42494,0.42504,0.42535,0.42523,0.42468,0.42405,0.42355,0.42322,0.42283,0.42282,0.42298,0.42311,0.42300,0.42237
,0.42138,0.42098,0.42108,0.42137,0.42175,0.42161,0.42094,0.42011,0.41929,0.41877,0.41818,0.41760,0.41724,0.41686,0.41669
,0.41686,0.41695,0.41702,0.41681,0.41626,0.41551,0.41521,0.41524,0.41559,0.41558,0.41505,0.41428,0.41337,0.41274,0.41202
,0.41102,0.40977,0.40863,0.40802,0.40776,0.40756,0.40723,0.40687,0.40628,0.40561,0.40499,0.40441,0.40367,0.40281,0.40205
,0.40141,0.40048,0.39954,0.39876,0.39822,0.39757,0.39704,0.39641,0.39590,0.39540,0.39519,0.39482,0.39449,0.39375,0.39329
)
  UT1_UTC = (-0.60960,-0.61043,-0.61138,-0.61241,-0.61337,-0.61423,-0.61485,-0.61515,-0.61513,-0.61487,-0.61457,-0.61442,-
0.61450,-0.61484,-0.61540,-0.61611,-0.61683,-0.61739,-0.61769,-0.61778,-0.61765,-0.61732,-0.61685,-0.61627,-0.61575,-
0.61536,-0.61507,-0.61491,-0.61492,-0.61508,-0.61539,-0.61579,-0.61614,-0.61631,-0.61625,-0.61596,-0.61548,-0.61497,-
0.61459,-0.61443,-0.61455,-0.61482,-0.61514,-0.61541,-0.61556,-0.61547,-0.61512,-0.61446,-0.61355,-0.61252,-0.61147,-
0.61048,-0.60963,-0.60891,-0.60835,-0.60801,-0.60785,-0.60778,-0.60777,-0.60769,-0.60749,-0.60712,-0.60651,-0.60571,-
0.60485,-0.60405,-0.60342,-0.60316,-0.60327,-0.60366,-0.60415,-0.60459,-0.60485,-0.60486,-0.60467,-0.60422,-0.60366,-
0.60304,-0.60239,-0.60178,-0.60127,-0.60097,-0.60090,-0.60107,-0.60140,-0.60185,-0.60227,-0.60257,-0.60268,-0.60259,-
0.60224,-0.60177,-0.60136,-0.60101,-0.60090,-0.60114,-0.60172,-0.60246,-0.60310,-0.60351,-0.60367,-0.60354,-0.60308,-
0.60242,-0.60158,-0.60066,-0.59981,-0.59911,-0.59866,-0.59851,-0.59862,-0.59886,-0.59923,-0.59968,-0.60015,-0.60053,-
0.60074,-0.60075,-0.60058,-0.60041,-0.60039,-0.60061,-0.60116,-0.60208,-0.60329,-0.60458,-0.60580,-0.60676,-0.60741,-
0.60773,-0.60779,-0.60767,-0.60748,-0.60733,-0.60724,-0.60732,-0.60762,-0.60807,-0.60874,-0.60949,-0.61023,-0.61088,-
0.61137,-0.61171,-0.61195,-0.61204,-0.61204,-0.61207,-0.61224,-0.61266,-0.61337,-0.61440,-0.61557,-0.61672,-0.61776,-
0.61859,-0.61917,-0.61953,-0.61974,-0.61988,-0.62002,-0.62025,-0.62055,-0.62099,-0.62166,-0.62248,-0.62350,-0.62459,-
0.62566,-0.62658,-0.62732,-0.62785,-0.62810,-0.62827,-0.62842,-0.62874,-0.62933,-0.63031,-0.63156,-0.63301)
END_GROUP = UT1_TIME_PARAMETERS

GROUP = TIME_SINCE_LAUNCH
  Decimal_Years =
(2005.4986,2005.5014,2005.5041,2005.5068,2005.5096,2005.5123,2005.5151,2005.5178,2005.5205,2005.5233,2005.5260,2005.5
288,2005.5315,2005.5342,2005.5370,2005.5397,2005.5425,2005.5452,2005.5479,2005.5507,2005.5534,2005.5562,2005.5589,20
05.5616,2005.5644,2005.5671,2005.5699,2005.5726,2005.5753,2005.5781,2005.5808,2005.5836,2005.5863,2005.5890,2005.591
8,2005.5945,2005.5973,2005.6000,2005.6027,2005.6055,2005.6082,2005.6110,2005.6137,2005.6164,2005.6192,2005.6219,2005.
6247,2005.6274,2005.6301,2005.6329,2005.6356,2005.6384,2005.6411,2005.6438,2005.6466,2005.6493,2005.6521,2005.6548,2
005.6575,2005.6603,2005.6630,2005.6658,2005.6685,2005.6712,2005.6740,2005.6767,2005.6795,2005.6822,2005.6849,2005.68
77,2005.6904,2005.6932,2005.6959,2005.6986,2005.7014,2005.7041,2005.7068,2005.7096,2005.7123,2005.7151,2005.7178,200
5.7205,2005.7233,2005.7260,2005.7288,2005.7315,2005.7342,2005.7370,2005.7397,2005.7425,2005.7452,2005.7479)
  Days_Since_Launch =
(7793,7794,7795,7796,7797,7798,7799,7800,7801,7802,7803,7804,7805,7806,7807,7808,7809,7810,7811,7812,7813,7814,7815,7
816,7817,7818,7819,7820,7821,7822,7823,7824,7825,7826,7827,7828,7829,7830,7831,7832,7833,7834,7835,7836,7837,7838,78
39,7840,7841,7842,7843,7844,7845,7846,7847,7848,7849,7850,7851,7852,7853,7854,7855,7856,7857,7858,7859,7860,7861,786
2,7863,7864,7865,7866,7867,7868,7869,7870,7871,7872,7873,7874,7875,7876,7877,7878,7879,7880,7881,7882,7883,7884)
  Day_Of_Year =
(182,183,184,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211
,212,213,214,215,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,240,241
,242,243,244,245,246,247,248,249,250,251,252,253,254,255,256,257,258,259,260,261,262,263,264,265,266,267,268,269,270,271
,272,273)
END_GROUP = TIME_SINCE_LAUNCH
```

```
GROUP = DETECTOR_STATUS
  Status_Band1 = ("00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000",
"00000", "00000", "00000", "00000")
  Status_Band2 = ("00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000",
"00000", "00000", "00000", "00000")
  Status_Band3 = ("00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000",
"00000", "00000", "00000", "00000")
  Status_Band4 = ("00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000",
"00000", "00000", "00000", "00000")
  Status_Band5 = ("00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000",
"00000", "00000", "00000", "00000")
  Status_Band6 = ("00000", "00000", "00000", "00000")
  Status_Band7 = ("00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000", "00000",
"00000", "00000", "00000", "00000")
END_GROUP = DETECTOR_STATUS

GROUP = DETECTOR_GAINS
  GROUP = GAIN_MODEL_PARAMETERS
    Band_1_Normalized_IC_Model_Coefficients = (0.103, 0.955, 0.881, 0.00423766)
    Band_2_Normalized_IC_Model_Coefficients = (0.081, 0.836, 0.902, 0.00947568)
    Band_3_Normalized_IC_Model_Coefficients = (0.108, 1.002, 0.876, 0.00925087)
    Band_4_Normalized_IC_Model_Coefficients = (0.030, 1.277, 0.904, 0.00943492)
    Band_5_Normalized_IC_Model_Coefficients = (0.031, 1.093, 0.959, 0.00765694)
    Band_6_Normalized_IC_Model_Coefficients = (0.000, 0.000, 0.000, 0.00000000)
    Band_7_Normalized_IC_Model_Coefficients = (0.033, 0.979, 0.972, 0.00803238)
    Time_Zero = 1984.207
    Band_1_LT_Model_Coefficients = (0.290100, 0.139900, 1.209000, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
    Band_2_LT_Model_Coefficients = (0.124600, 0.104500, 0.630500, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
    Band_3_LT_Model_Coefficients = (0.083900, 0.238600, 0.902800, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
    Band_4_LT_Model_Coefficients = (0.000000, 0.000000, 1.082000, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
    Band_5_LT_Model_Coefficients = (0.000000, 0.000000, 8.209000, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
    Band_6_LT_Model_Coefficients = (0.000000, 0.000000, 0.000000, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
    Band_7_LT_Model_Coefficients = (0.000000, 0.000000, 14.695000, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
  END_GROUP = GAIN_MODEL_PARAMETERS
  GROUP = OUTGASSING_CORRECTION
    Outgassing_Events = (1, 135, 177, 262, 351, 437, 499, 569, 674, 736, 801, 862, 920, 996, 1065, 1142, 1240, 1338, 1436, 1534,
1632, 1723, 1856, 1961, 2108, 2212, 2332, 2468, 2619, 2773, 2983, 3088, 3584, 3640, 4020, 4368, 4762, 5126, 5494, 5867, 6861,
7330, 8052, 8353, 9439, 9999, 9999, 9999, 9999, 9999)
    Band_5_Film_Refractive_Index_Part_1 = (1.2878, 1.2878, 1.2878, 1.2878, 1.2878, 1.2878, 1.2878, 1.2878, 1.2878, 1.2878,
1.2878, 1.2878, 1.2878, 1.2878, 1.2878, 1.2878)
    Band_5_Film_Absorption_Index_Part_1 = (7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4,
7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4)
    Band_5_ARC_Refractive_Index_Part_1 = (1.6739, 1.6739, 1.6739, 1.6739, 1.6739, 1.6739, 1.6739, 1.6739, 1.6739, 1.6739,
1.6739, 1.6739, 1.6739, 1.6739, 1.6739, 1.6739)
    Band_5_ARC_Thickness_Part_1 = (269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0,
269.0, 269.0, 269.0)
    Band_5_Oscillating_Period_Part_1 = (0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)
    Band_7_Film_Refractive_Index_Part_1 = (1.2606, 1.2606, 1.2606, 1.2606, 1.2606, 1.2606, 1.2606, 1.2606, 1.2606, 1.2606,
1.2606, 1.2606, 1.2606, 1.2606, 1.2606, 1.2606)
    Band_7_Film_Absorption_Index_Part_1 = (2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3,
2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3)
    Band_7_ARC_Refractive_Index_Part_1 = (1.6677, 1.6677, 1.6677, 1.6677, 1.6677, 1.6677, 1.6677, 1.6677, 1.6677, 1.6677,
1.6677, 1.6677, 1.6677, 1.6677, 1.6677, 1.6677)
    Band_7_ARC_Thickness_Part_1 = (326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9,
326.9, 326.9, 326.9)
    Band_7_Oscillating_Period_Part_1 = (0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)
    Band_5_Film_Refractive_Index_Part_2 = (1.2878, 1.2878, 1.2878, 1.2878, 1.2878, 1.2878, 1.2878, 1.2878, 1.2878, 1.2878,
1.2878, 1.2878, 1.2878, 1.2878, 1.2878, 1.2878)
    Band_5_Film_Absorption_Index_Part_2 = (7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4,
7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4)
    Band_5_ARC_Refractive_Index_Part_2 = (1.6739, 1.6739, 1.6739, 1.6739, 1.6739, 1.6739, 1.6739, 1.6739, 1.6739, 1.6739,
1.6739, 1.6739, 1.6739, 1.6739, 1.6739, 1.6739)
    Band_5_ARC_Thickness_Part_2 = (269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0,
269.0, 269.0, 269.0)
    Band_5_Oscillating_Period_Part_2 = (0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)
    Band_7_Film_Refractive_Index_Part_2 = (1.2606, 1.2606, 1.2606, 1.2606, 1.2606, 1.2606, 1.2606, 1.2606, 1.2606, 1.2606,
1.2606, 1.2606, 1.2606, 1.2606, 1.2606, 1.2606)
    Band_7_Film_Absorption_Index_Part_2 = (2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3,
2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3)
```

```
    Band_7_ARC_Refractive_Index_Part_2 = (1.6677, 1.6677, 1.6677, 1.6677, 1.6677, 1.6677, 1.6677, 1.6677, 1.6677, 1.6677,
1.6677, 1.6677, 1.6677, 1.6677, 1.6677, 1.6677)
    Band_7_ARC_Thickness_Part_2 = (326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9,
326.9, 326.9, 326.9)
    Band_7_Oscillating_Period_Part_2 = (0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)
    Band_5_Film_Refractive_Index_Part_3 = (1.2878, 1.2878, 1.2878, 1.2878, 1.2878, 1.2878, 1.2878, 1.2878, 1.2878, 1.2878,
1.2878, 1.2878, 1.2878, 1.2878, 1.2878, 1.2878)
    Band_5_Film_Absorption_Index_Part_3 = (7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4,
7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4, 7.258E-4)
    Band_5_ARC_Refractive_Index_Part_3 = (1.6739, 1.6739, 1.6739, 1.6739, 1.6739, 1.6739, 1.6739, 1.6739, 1.6739, 1.6739,
1.6739, 1.6739, 1.6739, 1.6739, 1.6739, 1.6739)
    Band_5_ARC_Thickness_Part_3 = (269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0, 269.0,
269.0, 269.0, 269.0)
    Band_5_Oscillating_Period_Part_3 = (0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)
    Band_7_Film_Refractive_Index_Part_3 = (1.2606, 1.2606, 1.2606, 1.2606, 1.2606, 1.2606, 1.2606, 1.2606, 1.2606, 1.2606,
1.2606, 1.2606, 1.2606, 1.2606, 1.2606, 1.2606)
    Band_7_Film_Absorption_Index_Part_3 = (2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3,
2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3, 2.472E-3)
    Band_7_ARC_Refractive_Index_Part_3 = (1.6677, 1.6677, 1.6677, 1.6677, 1.6677, 1.6677, 1.6677, 1.6677, 1.6677, 1.6677,
1.6677, 1.6677, 1.6677, 1.6677, 1.6677, 1.6677)
    Band_7_ARC_Thickness_Part_3 = (326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9, 326.9,
326.9, 326.9, 326.9)
    Band_7_Oscillating_Period_Part_3 = (0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)
    Band_5_Period_LT_Model_Exp_Scaling = (0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000)
    Band_5_Period_LT_Model_Attenuation = (0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0,
0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0)
    Band_5_Period_LT_Model_Slope = (0.03876, 0.03876, 0.03876, 0.03876, 0.03876, 0.03876, 0.03876, 0.03876, 0.03876,
0.03876, 0.03876, 0.03876, 0.03876, 0.03876, 0.03876, 0.03876)
    Band_5_Period_LT_Model_Offset = (-0.94, -0.94, -0.94, -0.94, -0.94, -0.94, -0.94, -0.94, -0.94, -0.94, -0.94, -0.94, -0.94, -0.94, -
0.94, -0.94)
    Band_7_Period_LT_Model_Exp_Scaling = (0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000)
    Band_7_Period_LT_Model_Attenuation = (0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0,
0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0, 0.0000E0)
    Band_7_Period_LT_Model_Slope = (0.06224, 0.06224, 0.06224, 0.06224, 0.06224, 0.06224, 0.06224, 0.06224, 0.06224,
0.06224, 0.06224, 0.06224, 0.06224, 0.06224, 0.06224, 0.06224)
    Band_7_Period_LT_Model_Offset = (-18.59, -18.59, -18.59, -18.59, -18.59, -18.59, -18.59, -18.59, -18.59, -18.59, -18.59, -18.59,
-18.59, -18.59, -18.59, -18.59)
  END_GROUP = OUTGASSING_CORRECTION
  GROUP = BAND_AVERAGE_GAINS
    Band_1_Average_Gain =
(1.2238,1.2238,1.2237,1.2237,1.2237,1.2237,1.2237,1.2237,1.2237,1.2237,1.2237,1.2237,1.2237,1.2237,1.2237,1.2237,1.2237,1.2
237,1.2237,1.2236,1.2236,1.2236,1.2236,1.2236,1.2236,1.2236,1.2236,1.2236,1.2236,1.2236,1.2236,1.2236,1.2236,1.2236,1.2236
,1.2236,1.2236,1.2235,1.2235,1.2235,1.2235,1.2235,1.2235,1.2235,1.2235,1.2235,1.2235,1.2235,1.2235,1.2235,1.2235,1.2235,1.2
235,1.2235,1.2235,1.2234,1.2234,1.2234,1.2234,1.2234,1.2234,1.2234,1.2234,1.2234,1.2234,1.2234,1.2234,1.2234,1.2234,1.2234
,1.2234,1.2234,1.2234,1.2233,1.2233,1.2233,1.2233,1.2233,1.2233,1.2233,1.2233,1.2233,1.2233,1.2233,1.2233,1.2233,1.2233,1.2
233,1.2233,1.2233,1.2233,1.2233)
    Band_2_Average_Gain =
(0.6440,0.6440,0.6440,0.6440,0.6440,0.6439,0.6439,0.6439,0.6439,0.6439,0.6439,0.6439,0.6439,0.6439,0.6439,0.6439,0.6439,0.6
439,0.6439,0.6439,0.6439,0.6439,0.6439,0.6439,0.6439,0.6439,0.6439,0.6439,0.6439,0.6438,0.6438,0.6438,0.6438
,0.6438,0.6438,0.6438,0.6438,0.6438,0.6438,0.6438,0.6438,0.6438,0.6438,0.6438,0.6438,0.6438,0.6438,0.6438,0.6438,0.6438,0.6
438,0.6438,0.6438,0.6438,0.6438,0.6437,0.6437,0.6437,0.6437,0.6437,0.6437,0.6437,0.6437,0.6437,0.6437,0.6437,0.6437,0.6437
,0.6437,0.6437,0.6437,0.6437,0.6437,0.6437,0.6437,0.6437,0.6437,0.6437,0.6437,0.6437,0.6437,0.6437,0.6436,0.6436,0.6436,0.6
436,0.6436,0.6436,0.6436,0.6436)
    Band_3_Average_Gain =
(0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9
033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033
,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9
033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033
,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9033,0.9
033,0.9033,0.9033,0.9033,0.9033)
    Band_4_Average_Gain =
(1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0
820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820
,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0
820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820
,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0820,1.0
820,1.0820,1.0820,1.0820,1.0820)
```

```
    Band_5_Average_Gain =
(8.0310,8.0340,8.0370,8.0400,8.0431,8.0461,8.0492,8.0522,8.0553,8.0583,8.0614,8.0645,8.0675,8.0706,8.0737,8.0767,8.0798,8.0
828,8.0858,8.0888,8.0919,8.0948,8.0978,8.1008,8.1037,8.1067,8.1096,8.1124,8.1153,8.1181,8.1210,8.1237,8.1265,8.1292,8.1319
,8.1346,8.1372,8.1398,8.1424,8.1449,8.1474,8.1498,8.1522,8.1546,8.1569,8.1592,8.1614,8.1636,8.1658,8.1679,8.1699,8.1719,8.1
738,8.1757,8.1776,8.1794,8.1811,8.1828,8.1844,8.1860,8.1875,8.1889,8.1903,8.1916,8.1929,8.1941,8.1953,8.1963,8.1974,8.1983
,8.1992,8.2001,8.2008,8.2015,8.2022,8.2027,8.2032,8.2037,8.2041,8.2044,8.2046,8.2048,8.2049,8.2049,8.2049,8.2048,8.2047,8.2
044,8.2041,8.2038,8.2033,8.2028)
    Band_6_Average_Gain =
(0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0
000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000
,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0
000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000
,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0
000,0.0000,0.0000,0.0000,0.0000)
    Band_7_Average_Gain =
(14.5103,14.5098,14.5094,14.5089,14.5084,14.5078,14.5073,14.5067,14.5060,14.5053,14.5046,14.5039,14.5032,14.5024,14.5015
,14.5007,14.4998,14.4989,14.4980,14.4970,14.4960,14.4950,14.4939,14.4928,14.4917,14.4906,14.4894,14.4882,14.4870,14.4857
,14.4845,14.4831,14.4818,14.4804,14.4791,14.4776,14.4762,14.4747,14.4732,14.4717,14.4702,14.4686,14.4670,14.4654,14.4637
,14.4620,14.4604,14.4586,14.4569,14.4551,14.4533,14.4515,14.4497,14.4478,14.4460,14.4441,14.4421,14.4402,14.4382,14.4362
,14.4342,14.4322,14.4302,14.4281,14.4260,14.4239,14.4218,14.4197,14.4175,14.4154,14.4132,14.4110,14.4087,14.4065,14.4043
,14.4020,14.3997,14.3974,14.3951,14.3928,14.3905,14.3881,14.3857,14.3834,14.3810,14.3786,14.3762,14.3738,14.3713,14.3689
,14.3665,14.3640)
    Band_5_Average_Gain_No_OG_Cor =
(8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2
090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090
,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2
090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090
,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2090,8.2
090,8.2090,8.2090,8.2090,8.2090)
    Band_7_Average_Gain_No_OG_Cor =
(14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950
,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950
,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950
,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950
,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950
,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950,14.6950
,14.6950,14.6950)
    Prelaunch_Average_Gains = (1.5553,0.7860,1.0203,1.0821,7.8751,0.000,14.7719)
  END_GROUP = BAND_AVERAGE_GAINS
  GROUP = PRELAUNCH_GAINS
    Band_1_Prelaunch_Gain =
(1.5597,1.5484,1.5662,1.5474,1.5713,1.5441,1.5614,1.5480,1.5508,1.5569,1.5614,1.5587,1.5582,1.5422,1.5537,1.5558)
    Band_2_Prelaunch_Gain =
(0.7878,0.7848,0.7840,0.7843,0.7854,0.7819,0.7837,0.7920,0.7813,0.7875,0.7893,0.7922,0.7875,0.7843,0.7912,0.7779)
    Band_3_Prelaunch_Gain =
(1.0208,1.0285,1.0185,1.0275,1.0189,1.0161,1.0044,1.0246,1.0096,1.0228,1.0105,1.0280,1.0157,1.0271,1.0217,1.0302)
    Band_4_Prelaunch_Gain =
(1.0895,1.0842,1.0776,1.0732,1.0820,1.0809,1.0796,1.0892,1.0725,1.0935,1.0802,1.0810,1.0799,1.0859,1.0778,1.0859)
    Band_5_Prelaunch_Gain =
(7.8980,7.8046,7.8503,7.8244,7.8647,7.8687,7.8170,7.9484,7.8722,7.8828,7.9342,7.9206,7.8783,7.8331,7.8926,7.9112)
    Band_6_Prelaunch_Gain =
(0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000)
    Band_7_Prelaunch_Gain =
(14.7592,14.6212,14.7372,14.6549,14.8868,14.6334,14.8428,14.7975,14.7647,14.7310,14.8674,14.8455,14.8476,14.8371,14.7428
,14.7810)
    Bandwidth = (0.066,0.082,0.067,0.128,0.217,1.970,0.252)
  END_GROUP = PRELAUNCH_GAINS
  GROUP = DETECTOR_RELATIVE_GAIN_PARAMETERS
    Band_1_Relative_Gain_Slope = (-2.840677E-7, 0.637819E-7, -0.453487E-7, 0.294604E-7, -1.043359E-7, -0.097234E-7,
1.300783E-7, -4.889805E-7, 3.257239E-7, 2.935155E-7, 0.370741E-7, 3.635368E-7, -0.487606E-7, -2.936268E-7, -0.537611E-7,
0.854337E-7)
    Band_2_Relative_Gain_Slope = (-2.875951E-7, -2.122416E-7, -4.207739E-7, -10.359747E-7, -2.064478E-7, -7.997079E-7, -
0.702890E-7, 1.779881E-7, 3.388579E-7, 0.191252E-7, 5.477989E-7, 2.931633E-7, 4.886511E-7, 2.906302E-7, 4.316067E-7,
4.452086E-7)
    Band_3_Relative_Gain_Slope = (-1.848917E-7, -2.473247E-7, -2.973620E-7, -2.314306E-7, 0.790002E-7, 3.272370E-7,
0.421390E-7, 5.108955E-7, 0.389347E-7, 2.747092E-7, 1.279057E-7, -0.780519E-7, -1.812731E-7, 0.725644E-7, -0.923700E-7, -
1.606817E-7)
    Band_4_Relative_Gain_Slope = (-0.094401E-7, -1.487869E-7, -1.784592E-7, -2.502456E-7, -0.592337E-7, 1.746026E-7,
1.243442E-7, 4.173347E-7, 5.390481E-7, -1.756787E-7, -2.359960E-7, 1.815873E-7, -2.672411E-7, -0.082633E-7, -1.765906E-7,
0.730183E-7)
```

```
    Band_5_Relative_Gain_Slope = (4.407965E-7, -3.669412E-7, -0.950583E-7, 1.530095E-7, 0.408164E-7, 2.517849E-7,
7.297025E-7, 0.984112E-7, 2.887332E-7, -12.465128E-7, 1.743392E-7, 3.965112E-7, 1.405165E-7, 0.273550E-7, -5.716163E-7, -
4.618474E-7)
    Band_6_Relative_Gain_Slope = (0.000E0,0.000E0,0.000E0,0.000E0)
    Band_7_Relative_Gain_Slope = (0.774561E-7, -0.381910E-7, 1.204894E-7, 0.399337E-7, 0.741647E-7, -2.799350E-7, -
1.266330E-7, -0.659051E-7, 0.479818E-7, 1.354188E-7, -0.670541E-7, 2.707472E-7, 2.004347E-7, -0.412806E-7, -1.773801E-7, -
1.702473E-7)
    Band_1_Relative_Gain_Intercept = (1.005217, 0.997814 , 1.004207, 1.003727, 1.001864, 0.989295, 1.001400, 0.989632,
0.997919, 0.999905, 1.007375, 1.001667, 1.000372, 0.991086,   1.003981, 1.004539)
    Band_2_Relative_Gain_Intercept = (0.997476, 1.000167, 0.993595, 1.008824, 0.992895, 1.014107, 0.990820, 1.018941,
0.988571, 1.008689, 1.001347, 1.017453, 0.993478, 0.991787, 0.999934, 0.981916)
    Band_3_Relative_Gain_Intercept = (1.004214, 1.018876, 0.995623, 1.005946, 0.996870, 0.990159, 0.978429, 1.006966,
0.983905, 1.005666, 0.985251, 1.008277, 0.994572, 1.004898, 1.004844, 1.015504)
    Band_4_Relative_Gain_Intercept = (1.006075, 1.001024, 0.996634, 0.987920, 1.007140, 1.002460, 1.020738, 1.005942,
0.986959, 1.008343, 0.999707, 0.993704, 0.994367, 0.993957, 0.995985, 0.999045)
    Band_5_Relative_Gain_Intercept = (0.997357, 0.992960, 0.993249, 0.983525, 1.007485, 0.991294, 0.982687, 1.022127,
1.001778, 1.005992, 0.999070, 1.002080, 1.004032, 1.001503, 1.006445, 1.008418)
    Band_6_Relative_Gain_Intercept = (0.0000,0.0000,0.0000,0.0000)
    Band_7_Relative_Gain_Intercept = (0.996700, 0.992548, 1.005695, 1.004850, 1.013378, 0.985534, 0.998672, 0.995592,
1.006528, 0.990521, 1.011307, 0.995497, 0.997351, 0.996631, 1.003821, 1.005378)
    Band_1_RG_ExpPar1 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_2_RG_ExpPar1 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_3_RG_ExpPar1 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_4_RG_ExpPar1 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_5_RG_ExpPar1 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_6_RG_ExpPar1 = (0.00,0.00,0.00,0.00)
    Band_7_RG_ExpPar1 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_1_RG_ExpPar2 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_2_RG_ExpPar2 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_3_RG_ExpPar2 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_4_RG_ExpPar2 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_5_RG_ExpPar2 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_6_RG_ExpPar2 = (0.00,0.00,0.00,0.00)
    Band_7_RG_ExpPar2 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_1_RG_AddPar1 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_2_RG_AddPar1 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_3_RG_AddPar1 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_4_RG_AddPar1 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_5_RG_AddPar1 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_6_RG_AddPar1 = (0.00,0.00,0.00,0.00)
    Band_7_RG_AddPar1 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_1_RG_AddPar2 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_2_RG_AddPar2 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_3_RG_AddPar2 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_4_RG_AddPar2 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_5_RG_AddPar2 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    Band_6_RG_AddPar2 = (0.00,0.00,0.00,0.00)
    Band_7_RG_AddPar2 = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
  END_GROUP = DETECTOR_RELATIVE_GAIN_PARAMETERS
END_GROUP = DETECTOR_GAINS

GROUP = BIAS_LOCATIONS
  Forward_Bias_Location_30 = 160
  Forward_Bias_Length_30 = 500
  Forward_IC_Region_30 = 830
  Reverse_Bias_Location_30 = 800
  Reverse_Bias_Length_30 = 500
  Reverse_IC_Region_30 = 660
  Forward_Bias_Location_120 = 40
  Forward_Bias_Length_120 = 130
  Forward_IC_Region_120 = 169
  Reverse_Bias_Location_120 = 200
  Reverse_Bias_Length_120 = 130
  Reverse_IC_Region_120 = 165
END_GROUP = BIAS_LOCATIONS

GROUP = DETECTOR_BIASES
  Band_1_Detector_Bias = (4.1, 3.7, 3.5, 2.9, 3.5, 3.4, 3.2, 3.5, 3.0, 3.1, 2.9, 3.1, 3.3, 3.4, 3.2, 3.3)
  Band_2_Detector_Bias = (3.4, 2.4, 2.8, 2.1, 2.5, 2.2, 2.5, 2.2, 2.5, 2.1, 2.5, 2.0, 2.8, 2.5, 2.7, 3.1)
  Band_3_Detector_Bias = (3.7, 2.9, 3.4, 3.0, 3.3, 3.2, 3.3, 2.6, 3.4, 2.9, 3.3, 3.0, 2.9, 2.9, 3.0, 3.0)
  Band_4_Detector_Bias = (3.7, 2.8, 2.9, 2.9, 2.3, 2.6, 2.2, 2.8, 2.9, 2.5, 3.0, 2.8, 3.0, 3.3, 2.7, 3.0)
  Band_5_Detector_Bias = (2.6, 2.3, 2.5, 2.3, 2.2, 2.3, 2.4, 2.5, 2.4, 2.5, 2.4, 2.3, 2.3, 2.4, 2.2, 2.3)
  Band_6_Detector_Bias = (0.0, 0.0, 0.0, 0.0)
  Band_7_Detector_Bias = (2.8, 2.3, 2.3, 2.3, 2.2, 2.3, 2.3, 2.4, 2.3, 2.2, 2.3, 2.2, 2.2, 2.4, 2.2, 2.3)
  GROUP = BIAS_LIMITS
    Band_1_Lower_Limit = (0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)
    Band_2_Lower_Limit = (0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)
    Band_3_Lower_Limit = (0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)
    Band_4_Lower_Limit = (0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)
    Band_5_Lower_Limit = (0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)
    Band_7_Lower_Limit = (0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)
    Band_1_Upper_Limit = (6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0)
    Band_2_Upper_Limit = (6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0)
    Band_3_Upper_Limit = (6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0)
    Band_4_Upper_Limit = (6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0)
    Band_5_Upper_Limit = (3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5)
    Band_7_Upper_Limit = (3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5)
  END_GROUP = BIAS_LIMITS
  GROUP = PRELAUNCH_BIASES
    Band_1_Prelaunch_Bias = (2.2965, 1.9313, 1.8734, 1.8895, 1.7628, 1.9744, 1.7435, 2.1147, 1.6412, 1.8049, 1.5761, 1.7649,
1.6324, 1.8487, 1.6416, 1.8337)
    Band_2_Prelaunch_Bias = (2.2691, 1.5379, 1.8693, 1.5357, 1.5069, 1.8161, 1.5605, 1.7427, 1.7117, 1.5873, 1.8789, 1.6117,
1.5945, 1.5986, 1.6357, 1.5766)
    Band_3_Prelaunch_Bias = (2.4569, 1.9920, 1.9709, 1.7282, 1.8442, 1.7437, 1.8571, 1.8852, 2.0105, 1.8130, 1.8650, 1.7688,
1.7462, 1.8063, 1.7881, 1.8836)
    Band_4_Prelaunch_Bias = (2.6652, 2.1440, 2.4287, 1.9523, 2.2046, 2.3107, 2.6408, 2.2091, 2.1524, 2.0022, 2.4152, 2.0702,
2.1371, 2.1669, 2.1682, 2.1291)
    Band_5_Prelaunch_Bias = (3.5727, 3.2601, 3.2736, 3.2701, 3.1314, 3.2506, 3.1252, 3.5024, 3.2843, 3.3784, 3.2440, 3.2882,
3.1520, 3.3691, 3.1806, 3.3460)
    Band_6_Prelaunch_Bias = (0.00, 0.00, 0.00, 0.00)
    Band_7_Prelaunch_Bias = (3.8241, 3.2194, 3.3549, 3.2758, 3.1052, 3.2558, 3.0121, 3.2427, 3.1006, 3.1922, 3.1158, 3.0190,
3.0402, 3.2440, 3.1068, 3.2790)
  END_GROUP = PRELAUNCH_BIASES
END_GROUP = DETECTOR_BIASES

GROUP = ACCA_BIASES
  B1_ACCA_Bias = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
  B2_ACCA_Bias = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
  B3_ACCA_Bias = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
  B4_ACCA_Bias = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
  B5_ACCA_Bias = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
  B6_ACCA_Bias = (0.00,0.00,0.00,0.00)
  B7_ACCA_Bias = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
END_GROUP = ACCA_BIASES

GROUP = ACCA_THRESHOLDS
  Thresh_B3 = 0.0000
  Thresh_B3_Lower = 0.00
  Thresh_B56 = 0.000
  Thresh_B6 = 0.000
  Thresh_B45_Ratio = 0.0000
  Thresh_B42_Ratio = 0.0000
  Thresh_B43_Ratio = 0.0000
  Thresh_NDSI_Max = 0.0000
  Thresh_NDSI_Min = 0.0000
  Thresh_NDSI_Snow = 0.0000
  Cloud_Percent_Min = 0.0000
  Desert_Index = 0.0000
  Thresh_Snow_Percent = 0.0000
  Thermal_Effect_High = 0.0000
  Thermal_Effect_Low = 0.0000
  B6Max_Maxthresh_Diff = 0.000
END_GROUP = ACCA_THRESHOLDS

GROUP = SOLAR_SPECTRAL_IRRADIANCES
  B1_Solar_Irradiance = 1957.000
  B2_Solar_Irradiance = 1826.000
  B3_Solar_Irradiance = 1554.000
  B4_Solar_Irradiance = 1036.000
  B5_Solar_Irradiance =  215.000
  B7_Solar_Irradiance =   80.670
END_GROUP = SOLAR_SPECTRAL_IRRADIANCES

GROUP = BAND_6_CALIBRATION_COEFFICIENTS
  Temp_To_Rad = (5.1292E-4, 1.7651E-1, 1.6023E1)
  a = (0.69, 0.65, 0.69, 0.64)
  b = (0.841, 0.841, 0.831, 0.829)
  c = (1.639, 1.990, 1.583, 1.971)
END_GROUP = BAND_6_CALIBRATION_COEFFICIENTS

GROUP = THERMAL_CONSTANTS
  K1_Constant =   607.76
  K2_Constant =  1260.56
END_GROUP = THERMAL_CONSTANTS

GROUP = SCALING_PARAMETERS
  B1_Lmin_Lmax = (-1.5200, 193.0000)
  B2_Lmin_Lmax = (-2.8400, 365.0000)
  B3_Lmin_Lmax = (-1.1700, 264.0000)
  B4_Lmin_Lmax = (-1.5100, 221.0000)
  B5_Lmin_Lmax = (-0.3700,  30.2000)
  B6_Lmin_Lmax = ( 1.2378,  15.3030)
  B7_Lmin_Lmax = (-0.1500,  16.5000)
  B1_Lmin_Lmax_LUT03 = (-1.5200,193.0000)
  B2_Lmin_Lmax_LUT03 = (-2.8400,365.0000)
  B3_Lmin_Lmax_LUT03 = (-1.1700,264.0000)
  B4_Lmin_Lmax_LUT03 = (-1.5100,221.0000)
  B5_Lmin_Lmax_LUT03 = (-0.3700,30.2000)
  B6_Lmin_Lmax_LUT03 = (1.2378,15.3030)
  B7_Lmin_Lmax_LUT03 = (-0.1500,16.5000)
  B1_Lmin_Lmax_IC = (-1.5200, 152.1000)
  B2_Lmin_Lmax_IC = (-2.8400, 296.8100)
  B3_Lmin_Lmax_IC = (-1.1700, 204.3000)
  B4_Lmin_Lmax_IC = (-1.5100, 206.2000)
  B5_Lmin_Lmax_IC = (-0.3700,  27.1900)
  B6_Lmin_Lmax_IC = ( 1.2378,  15.3030)
  B7_Lmin_Lmax_IC = (-0.1500,  14.3800)
END_GROUP = SCALING_PARAMETERS

GROUP = MTF_COMPENSATION
  B1_weights_along = (+0.5000,+0.0000,-0.5000,+0.0000,+0.0000)
  B1_weights_across = (+0.5000,+0.0000,-0.5000,+0.0000,+0.0000)
  B2_weights_along = (+0.5000,+0.0000,-0.5000,+0.0000,+0.0000)
  B2_weights_across = (+0.5000,+0.0000,-0.5000,+0.0000,+0.0000)
  B3_weights_along = (+0.5000,+0.0000,-0.5000,+0.0000,+0.0000)
  B3_weights_across = (+0.5000,+0.0000,-0.5000,+0.0000,+0.0000)
  B4_weights_along = (+0.5000,+0.0000,-0.5000,+0.0000,+0.0000)
  B4_weights_across = (+0.5000,+0.0000,-0.5000,+0.0000,+0.0000)
  B5_weights_along = (+0.5000,+0.0000,-0.5000,+0.0000,+0.0000)
  B5_weights_across = (+0.5000,+0.0000,-0.5000,+0.0000,+0.0000)
  B6_weights_along = (+0.5000,+0.0000,-0.5000,+0.0000,+0.0000)
  B6_weights_across = (+0.5000,+0.0000,-0.5000,+0.0000,+0.0000)
  B7_weights_along = (+0.5000,+0.0000,-0.5000,+0.0000,+0.0000)
  B7_weights_across = (+0.5000,+0.0000,-0.5000,+0.0000,+0.0000)
END_GROUP = MTF_COMPENSATION

GROUP = MEMORY_EFFECT
  GROUP = ME_MAGNITUDES
    B1_kME_Magnitude = (-1.76447e-05, -1.86280e-05, -1.37093e-05, -1.52647e-05, -1.35158e-05, -1.54136e-05, -1.44766e-05, -
1.56425e-05, -1.16601e-05, -1.22041e-05, -1.27097e-05, -1.22420e-05, -1.63133e-05, -1.51161e-05, -1.54951e-05, -1.60915e-05)
    B2_kME_Magnitude = (-1.54035e-05, -1.29685e-05, -1.31833e-05, -4.67519e-06, -1.25527e-05, -5.09582e-06, -1.32666e-05, -
5.02027e-06, -1.03880e-05, -3.60229e-06, -7.42092e-06, -3.30370e-06, -1.90365e-05, -9.09655e-06, -1.53207e-05, -2.02743e-05)
    B3_kME_Magnitude = (-1.76613e-05, -1.23867e-05, -1.66545e-05, -1.35345e-05, -2.19871e-05, -1.48708e-05, -2.06274e-05, -
7.82505e-06, -1.64582e-05, -1.34487e-05, -1.62413e-05, -1.23451e-05, -1.21724e-05, -1.02491e-05, -1.33464e-05, -1.27219e-05)
    B4_kME_Magnitude = (-2.61555e-05, -1.05822e-05, -1.31148e-05, -1.29141e-05, -3.95619e-06, -8.34379e-06, -2.80861e-06, -
1.17550e-05, -1.25015e-05, -6.92007e-06, -1.52136e-05, -1.10395e-05, -1.36381e-05, -1.90905e-05, -9.62389e-06, -1.41052e-05)
    B5_kME_Magnitude = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B6_kME_Magnitude = (0.0, 0.0, 0.0, 0.0)
    B7_kME_Magnitude = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  END_GROUP = ME_MAGNITUDES
  GROUP = ME_SCALING
    B1_ME_Scal_Factor = (1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0)
    B2_ME_Scal_Factor = (1.7, 1.9, 1.8, 5.6, 1.8, 4.2, 2.0, 5.2, 2.6, 6.8, 2.6, 8.9, 1.4, 2.5, 1.5, 1.2)
    B3_ME_Scal_Factor = (0.8, 1.3, 1.1, 2.5, 1.3, 1.7, 1.8, 1.4, 1.7, 1.6, 1.9, 1.4, 2.7, 1.8, 1.8, 1.5)
    B4_ME_Scal_Factor = (0.6, 2.0, 1.8, 1.8, 5.3, 2.6, 9.7, 2.2, 2.0, 3.3, 1.6, 2.3, 1.6, 1.2, 2.4, 2.0)
    B5_ME_Scal_Factor = (1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0)
    B6_ME_Scal_Factor = (1.0, 1.0, 1.0, 1.0)
    B7_ME_Scal_Factor = (1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0)
  END_GROUP = ME_SCALING
  GROUP = ME_TIME_CONSTANTS
    B1_ME_Time_Constant = (1258.45, 1187.62, 1359.82, 658.490, 1557.41, 1096.47, 1071.37, 1121.04, 1311.20, 1488.32,
1205.67, 1391.68, 1148.51, 1317.53, 1193.48, 1130.78)
    B2_ME_Time_Constant = (1169.14, 1072.15, 1089.48, 1075.71, 1179.10, 1091.09, 1057.39, 1195.10, 1158.54, 1189.14,
1389.76, 625.268, 1107.47, 1391.17, 1093.13, 1236.27)
    B3_ME_Time_Constant = (1209.25, 1284.96, 1268.57, 1225.72, 1062.90, 1280.65, 1171.59, 1211.24, 1330.47, 1091.47,
1262.61, 1295.73, 1291.65, 1583.28, 1196.07, 1232.87)
    B4_ME_Time_Constant = (1059.56, 1211.49, 1216.66, 1195.98, 1296.84, 1264.98, 799.308, 1292.08, 1276.50, 1397.12,
1220.40, 1382.13, 1309.25, 1360.02, 1382.83, 1212.61)
    B5_ME_Time_Constant = (1100, 1100, 1100, 1100, 1100, 1100, 1100, 1100, 1100, 1100, 1100, 1100, 1100, 1100, 1100, 1100)
    B6_ME_Time_Constant = (1100, 1100, 1100, 1100)
    B7_ME_Time_Constant = (1100, 1100, 1100, 1100, 1100, 1100, 1100, 1100, 1100, 1100, 1100, 1100, 1100, 1100, 1100, 1100)
  END_GROUP = ME_TIME_CONSTANTS
  GROUP = ME_FILTER_PARAMETERS
    ME_Filter_Widths = (1, 19, 25, 25, 50, 50, 50, 75, 75, 75, 75, 100, 100, 100, 100, 100, 130, 130, 130, 130, 130, 130, 150, 150,
150, 150, 150, 150, 150, 150)
  END_GROUP = ME_FILTER_PARAMETERS
END_GROUP = MEMORY_EFFECT

GROUP = GHOST_PULSE
  Ghost_Pulse_Endpoints = (0.00, 0.00)
END_GROUP = GHOST_PULSE

GROUP = SCAN_CORRELATED_SHIFT
  SCS_Reference_Detector_1 = (7, 7, 1)
  SCS_Reference_Detector_2 = (0, 0, 0)
  SCS_Reference_Detector_3 = (0, 0, 0)
  SCS_State_Mask_Parameters = (0.000007113387, 601, 2.15, 0.05, 0.05)
  B1_SCS_Additive_Correction_Factors = (1.1569804e-02, -1.5625911e-01, 3.2996424e-02, -1.1304116e-01, 1.4710412e-02, -
1.6058411e-01, 7.2760472e-02, -1.2827364e-01, 1.8843410e-02, -3.3344272e-01, 1.8333317e-02, -1.9159730e-01, -3.9324667e-
02, -3.4885825e-01, -3.9938065e-02, -2.5303062e-01)
  B2_SCS_Additive_Correction_Factors = (7.2098294e-01, 3.7425930e-02, 3.6839192e-01, -5.9152038e-02, 2.5968120e-01,
8.3694987e-02, 2.8259830e-01, 9.5723345e-02, 1.6736335e-01, 1.2281209e-02, 2.2465824e-01, 3.5409573e-02, 3.6006368e-01,
1.6311572e-01, 3.0721359e-01, 6.7545385e-01)
  B3_SCS_Additive_Correction_Factors = (5.2192020e-01, 1.3320758e-01, 5.4508526e-01, 4.0118462e-01, 4.0680130e-01,
5.6667455e-01, 3.1066546e-01, 1.7622443e-01, 4.7273162e-01, 3.3055915e-01, 4.6439925e-01, 3.9249319e-01, 3.6471880e-01,
2.7372833e-01, 3.8050369e-01, 2.4974505e-01)
  B4_SCS_Additive_Correction_Factors = (4.4828032e-01, 8.9812094e-02, 7.5109591e-02, 2.4592970e-01, 3.4749578e-02,
8.6333297e-03, 1.4461188e-02, -3.4450797e-02, 6.4423982e-02, -1.6255673e-04, 7.4066195e-02, -1.4154289e-02, 1.5053092e-
01, -1.4437600e-02, 3.8741075e-02, -9.7693635e-02)
  B5_SCS_Additive_Correction_Factors = (1.7000721e-01, -1.6055916e-02, 2.4704872e-01, -5.2022618e-02, -5.3244174e-02, -
1.4390805e-01, 8.3640852e-02, -1.3654846e-01, 1.0964904e-01, 7.5864344e-02, 1.2969957e-01, -1.5621545e-01, 6.0418422e-
02, -9.2451386e-02, -7.0901081e-02, -1.5514363e-01)
  B6_SCS_Additive_Correction_Factors = (0.0, 0.0, 0.0, 0.0)
  B7_SCS_Additive_Correction_Factors = (7.1424251e-02, -6.9256432e-02, -2.3865368e-02, -1.7026510e-01, 5.1228012e-02, -
1.5863398e-01, 2.6906569e-01, -2.4568812e-01, 2.1776210e-01, -2.4221505e-01, 1.6413804e-01, -1.1843650e-01, 8.0542090e-
02, -8.8635934e-02, 7.5825711e-02, -1.5623075e-01)
END_GROUP = SCAN_CORRELATED_SHIFT

GROUP = STRIPING
  Correction_Reference_B1 = 0
  Correction_Reference_B2 = 0
  Correction_Reference_B3 = 0
  Correction_Reference_B4 = 0
  Correction_Reference_B5 = 0
  Correction_Reference_B6 = 0
  Correction_Reference_B7 = 0
END_GROUP = STRIPING

GROUP = HISTOGRAM
  GROUP = DETECTOR_NOISE
```

```
    Detector_Noise_Level_B1 = (1.01420, 1.10810, 0.97312, 1.09620, 1.02470, 1.11080, 1.02750, 1.14780, 0.85664, 0.99731,
0.87608, 0.99549, 0.92797, 1.11260, 0.91055, 0.99585)
    Detector_Noise_Level_B2 = (0.43186, 0.78657, 0.30505, 0.29766, 0.40456, 0.25636, 0.35438, 0.21768, 0.21608, 0.22246,
0.22161, 0.21231, 0.36519, 0.28803, 0.30388, 0.46041)
    Detector_Noise_Level_B3 = (0.57846, 0.41966, 0.48105, 0.44310, 0.48783, 0.32639, 0.43177, 0.32664, 0.48924, 0.34578,
0.41729, 0.61269, 0.41055, 0.37578, 0.38505, 0.46610)
    Detector_Noise_Level_B4 = (0.50116, 0.25937, 0.27456, 0.09725, 0.06361, 0.29896, 0.10155, 0.34510, 0.20048, 0.28941,
0.32645, 0.27481, 0.12329, 0.39061, 0.15644, 0.53212)
    Detector_Noise_Level_B5 = (0.84463, 0.85743, 1.00180, 0.95999, 0.92987, 0.86803, 1.72470, 0.90295, 1.05760, 1.31020,
0.89113, 1.00900, 0.98761, 0.89530, 0.88833, 0.89932)
    Detector_Noise_Level_B6 = (0.50000, 0.50000, 0.50000, 0.50000)
    Detector_Noise_Level_B7 = (0.83853, 0.89672, 0.87652, 0.89100, 0.80657, 0.94658, 0.85140, 1.06780, 0.81896, 1.07000,
0.86032, 1.00920, 0.79012, 0.97129, 0.79065, 0.93681)
  END_GROUP = DETECTOR_NOISE
  GROUP = DET_SHUTTER_NOISE
    Det_Shutter_Noise_Level_B1 = (0.974674, 1.085858, 0.948344, 1.074075, 1.005915, 1.084420, 1.015990, 1.115435, 0.836026,
0.939500, 0.864744, 0.960542, 0.878079, 1.063528, 0.926818, 0.943876)
    Det_Shutter_Noise_Level_B2 = (0.462904, 0.780979, 0.380299, 0.276695, 0.364058, 0.282149, 0.330045, 0.234384, 0.260725,
0.188100, 0.269713, 0.153767, 0.366932, 0.310009, 0.338156, 0.483779)
    Det_Shutter_Noise_Level_B3 = (0.582067, 0.430551, 0.554728, 0.491207, 0.543677, 0.434223, 0.483965, 0.389486, 0.575046,
0.424408, 0.499904, 0.652198, 0.485007, 0.430786, 0.473244, 0.503794)
    Det_Shutter_Noise_Level_B4 = (0.321735, 0.303078, 0.306971, 0.260420, 0.157535, 0.295219, 0.169388, 0.294909, 0.253392,
0.266226, 0.352027, 0.244123, 0.236554, 0.357290, 0.193870, 0.509188)
    Det_Shutter_Noise_Level_B5 = (0.841330, 0.861242, 1.015888, 0.960300, 0.928367, 0.863805, 1.734060, 0.900582, 1.062418,
1.316920, 0.902176, 1.007545, 0.987794, 0.891277, 0.883873, 0.885317)
    Det_Shutter_Noise_Level_B6 = (0.50, 0.50, 0.50, 0.50)
    Det_Shutter_Noise_Level_B7 = (0.834749, 0.893375, 0.879477, 0.877903, 0.817128, 0.938552, 0.876364, 1.061585, 0.843950,
1.057590, 0.882281, 1.002640, 0.799837, 0.952818, 0.800899, 0.920797)
  END_GROUP = DET_SHUTTER_NOISE
  GROUP = REFERENCE_DETECTORS
    Reference_Detector_B1 = 01
    Reference_Detector_B2 = 01
    Reference_Detector_B3 = 01
    Reference_Detector_B4 = 01
    Reference_Detector_B5 = 01
    Reference_Detector_B6 = 01
    Reference_Detector_B7 = 01
  END_GROUP = REFERENCE_DETECTORS
  GROUP = SATURATION_THRESHOLDS
    Saturation_Bin_Threshold_B1 = 10000
    Saturation_Bin_Threshold_B2 = 10000
    Saturation_Bin_Threshold_B3 = 10000
    Saturation_Bin_Threshold_B4 = 10000
    Saturation_Bin_Threshold_B5 = 10000
    Saturation_Bin_Threshold_B6 = 10000
    Saturation_Bin_Threshold_B7 = 10000
  END_GROUP = SATURATION_THRESHOLDS
  GROUP = ADJACENT_BINS
    GROUP = BIN_NUMBER
      Adjacent_Bin_Number_B1 = 2
      Adjacent_Bin_Number_B2 = 2
      Adjacent_Bin_Number_B3 = 2
      Adjacent_Bin_Number_B4 = 2
      Adjacent_Bin_Number_B5 = 2
      Adjacent_Bin_Number_B6 = 3
      Adjacent_Bin_Number_B7 = 2
    END_GROUP = BIN_NUMBER
    GROUP = BIN_THRESHOLD
      Adjacent_Bin_Threshold_B1 = 1000
      Adjacent_Bin_Threshold_B2 = 1000
      Adjacent_Bin_Threshold_B3 = 1000
      Adjacent_Bin_Threshold_B4 = 1000
      Adjacent_Bin_Threshold_B5 = 1000
      Adjacent_Bin_Threshold_B6 = 1000
      Adjacent_Bin_Threshold_B7 = 1000
    END_GROUP = BIN_THRESHOLD
  END_GROUP = ADJACENT_BINS
  GROUP = STARTING_PIXEL
    Start_pixel_B1 = 1
    Start_pixel_B2 = 1
    Start_pixel_B3 = 1
    Start_pixel_B4 = 1
    Start_pixel_B5 = 1
    Start_pixel_B6 = 1
    Start_pixel_B7 = 1
  END_GROUP = STARTING_PIXEL
  GROUP = WINDOW_WIDTH
    Window_Samples_B1 = 6300
    Window_Samples_B2 = 6300
    Window_Samples_B3 = 6300
    Window_Samples_B4 = 6300
    Window_Samples_B5 = 6300
    Window_Samples_B6 = 1800
    Window_Samples_B7 = 6300
  END_GROUP = WINDOW_WIDTH
 GROUP = WINDOW_LENGTH
    Window_Scans_B1 = 374
    Window_Scans_B2 = 374
    Window_Scans_B3 = 374
    Window_Scans_B4 = 374
    Window_Scans_B5 = 374
    Window_Scans_B6 = 374
    Window_Scans_B7 = 374
  END_GROUP = WINDOW_LENGTH
  GROUP = OVERLAPPING_SCANS
    Overlap_Scans_B1 = 0
    Overlap_Scans_B2 = 0
    Overlap_Scans_B3 = 0
    Overlap_Scans_B4 = 0
    Overlap_Scans_B5 = 0
    Overlap_Scans_B6 = 0
    Overlap_Scans_B7 = 0
  END_GROUP = OVERLAPPING_SCANS
END_GROUP = HISTOGRAM

GROUP = IMPULSE_NOISE
  Median_Filter_Width = 3
  GROUP = IN_THRESHOLD
    B1_Threshold = (11.126630, 7.047142, 11.258280, 7.086417, 7.313617, 7.051933, 7.280033, 6.948550, 11.819869, 11.302503,
11.676280, 11.197291, 11.609605, 7.121575, 11.365911, 11.280620)
    B2_Threshold = (27.370958, 12.095104, 28.197010, 29.233050, 28.359418, 29.178515, 28.699555, 29.656160, 29.392755,
30.119000, 29.302868, 30.462333, 28.330680, 28.899908, 28.618438, 27.162210)
    B3_Threshold = (13.089668, 27.694488, 13.226363, 27.087933, 13.281614, 27.657768, 27.160348, 28.105145, 13.124771,
27.755923, 27.000958, 12.739013, 27.149933, 27.692145, 27.267560, 13.481029)
    B4_Threshold = (28.782648, 28.969225, 28.930288, 29.395801, 30.424654, 29.047815, 30.306125, 29.050908, 29.466080,
29.337743, 28.479733, 29.558770, 29.634457, 28.427103, 30.061300, 13.454063)
    B5_Threshold = (11.793350, 11.693791, 7.280375, 11.198499, 11.358166, 11.680976, 3.664850, 11.497091, 7.125275,
6.276933, 11.489123, 7.308183, 11.061033, 11.543616, 11.580634, 11.573416)
    B6_Threshold = (14.00, 14.00, 14.00, 14.00)
    B7_Threshold = (11.826255, 11.533128, 11.602615, 11.610484, 11.914359, 11.307243, 11.618183, 7.128050, 11.780253,
7.141367, 11.588596, 7.324533, 12.000815, 11.235910, 11.995505, 11.396015)
  END_GROUP = IN_THRESHOLD

  GROUP = IN_SIGMA_THRESHOLD
    B1_Sigma_Threshold = (11.470646, 9.752749, 11.897233, 9.884621, 10.960780, 9.744098, 10.851037, 9.288015, 14.249458,
11.721827, 13.674893, 11.531426, 12.974815, 9.872976,     12.428756, 11.700274)
    B2_Sigma_Threshold = (36.577715, 15.616701, 49.727019, 57.486982, 34.521843, 66.925042, 39.557548, 88.785050,
78.132323, 88.525930, 78.823698, 90.807645, 38.055956, 53.019960, 47.984342, 29.657406)
    B3_Sigma_Threshold = (23.551293, 34.343703, 27.891597, 31.003132, 27.542815, 42.639758, 32.333824, 45.687685,
27.210565, 40.902707, 32.512785, 20.987118, 32.703827, 38.679962, 34.988554, 29.934559)
    B4_Sigma_Threshold = (29.662313, 64.157919, 59.652783, 194.972189, 379.640959, 54.284023, 189.930603, 47.623189,
91.193503, 55.560118, 47.555149, 61.843089, 162.685097, 40.240850, 129.948732, 25.682897)
    B5_Sigma_Threshold = (14.147629, 13.568071, 10.888876, 11.608848, 12.162483, 13.397505, 4.239762, 12.671498,
10.084078, 7.186834, 12.898634, 10.850300, 11.171587, 12.902970, 13.054485, 12.903827)
    B6_Sigma_Threshold = (14.00, 14.00, 14.00, 14.00)
    B7_Sigma_Threshold = (14.247058, 12.840438, 13.210084, 13.027571, 14.779241, 11.935567, 13.642113, 10.025293,
14.340698, 9.998927, 13.453274, 10.904363, 15.173158, 11.638835, 15.229216, 12.227392)
  END_GROUP = IN_SIGMA_THRESHOLD
END_GROUP = IMPULSE_NOISE
```

```
GROUP = COHERENT_NOISE
  Frequency_Components = 10
  GROUP = CN_FREQUENCY_PARAMETERS
    GROUP = FREQUENCY_MEANS
      B1_Frequency_Mean = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B2_Frequency_Mean = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B3_Frequency_Mean = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B4_Frequency_Mean = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B5_Frequency_Mean = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B6_Frequency_Mean = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B7_Frequency_Mean = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    END_GROUP = FREQUENCY_MEANS
    GROUP = FREQUENCY_SIGMAS
      B1_Frequency_Sigma = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B2_Frequency_Sigma = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B3_Frequency_Sigma = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B4_Frequency_Sigma = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B5_Frequency_Sigma = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B6_Frequency_Sigma = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B7_Frequency_Sigma = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    END_GROUP = FREQUENCY_SIGMAS
    GROUP = FREQUENCY_MINIMUMS
      B1_Frequency_Min = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B2_Frequency_Min = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B3_Frequency_Min = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B4_Frequency_Min = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B5_Frequency_Min = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B6_Frequency_Min = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B7_Frequency_Min = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    END_GROUP = FREQUENCY_MINIMUMS
    GROUP = FREQUENCY_MAXIMUMS
      B1_Frequency_Max = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B2_Frequency_Max = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B3_Frequency_Max = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B4_Frequency_Max = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B5_Frequency_Max = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B6_Frequency_Max = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B7_Frequency_Max = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    END_GROUP = FREQUENCY_MAXIMUMS
  END_GROUP = CN_FREQUENCY_PARAMETERS
  GROUP = CN_PHASE_MEANS
    B1_Phase_Mean      = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    B2_Phase_Mean      = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    B3_Phase_Mean      = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    B4_Phase_Mean      = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    B5_Phase_Mean      = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    B6_Phase_Mean      = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    B7_Phase_Mean      = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
  END_GROUP = CN_PHASE_MEANS
  GROUP = CN_MAGNITUDE_PARAMETERS
    GROUP = MAGNITUDE_MEANS
      B1_Magnitude_Mean      = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B2_Magnitude_Mean      = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B3_Magnitude_Mean      = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B4_Magnitude_Mean      = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B5_Magnitude_Mean      = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B6_Magnitude_Mean      = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B7_Magnitude_Mean      = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    END_GROUP = MAGNITUDE_MEANS
    GROUP = MAGNITUDE_SIGMAS
      B1_Magnitude_Sigma     = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B2_Magnitude_Sigma     = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B3_Magnitude_Sigma     = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B4_Magnitude_Sigma     = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B5_Magnitude_Sigma     = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B6_Magnitude_Sigma     = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B7_Magnitude_Sigma     = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    END_GROUP = MAGNITUDE_SIGMAS
    GROUP = MAGNITUDE_MINIMUMS
      B1_Magnitude_Min       = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B2_Magnitude_Min       = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B3_Magnitude_Min       = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B4_Magnitude_Min       = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B5_Magnitude_Min       = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B6_Magnitude_Min       = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B7_Magnitude_Min       = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    END_GROUP = MAGNITUDE_MINIMUMS
    GROUP = MAGNITUDE_MAXIMUMS
      B1_Magnitude_Max       = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B2_Magnitude_Max       = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B3_Magnitude_Max       = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B4_Magnitude_Max       = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B5_Magnitude_Max       = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B6_Magnitude_Max       = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B7_Magnitude_Max       = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    END_GROUP = MAGNITUDE_MAXIMUMS
  END_GROUP = CN_MAGNITUDE_PARAMETERS
END_GROUP = COHERENT_NOISE

GROUP = CHANNEL_SATURATION
  High_Level_B1 = (255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255)
  High_Level_B2 = (255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255)
  High_Level_B3 = (255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255)
  High_Level_B4 = (255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255)
  High_Level_B5 = (255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255)
  High_Level_B6 = (255, 255, 255, 255)
  High_Level_B7 = (255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255)
  Low_Level_B1  = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
  Low_Level_B2  = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
  Low_Level_B3  = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
  Low_Level_B4  = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
  Low_Level_B5  = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
  Low_Level_B6  = (0, 0, 0, 0)
  Low_Level_B7  = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
END_GROUP = CHANNEL_SATURATION

GROUP = REFERENCE_TEMPERATURE
  B1_RTemp    = 23.000
  B2_RTemp    = 23.000
  B3_RTemp    = 23.000
  B4_RTemp    = 23.000
  B5_RTemp    = -181.00
  B6_RTemp    = -181.00
  B7_RTemp    = -181.00
END_GROUP = REFERENCE_TEMPERATURE

GROUP = SENSITIVITY_TEMPERATURES
  B1_SCoeff       = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B2_SCoeff       = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B3_SCoeff       = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B4_SCoeff       = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B5_SCoeff       = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B6_SCoeff       = (0.0, 0.0, 0.0, 0.0)
  B6_SCoeff_Off   = (0.0, 0.0, 0.0, 0.0)
  B7_SCoeff       = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
END_GROUP = SENSITIVITY_TEMPERATURES

GROUP = LAMP_RADIANCE
  GROUP = TRENDING_COEFFS
    Lamp1_Coeffs = (+0.0, +0.0)
    Lamp2_Coeffs = (+0.0, +0.0)
    Lamp3_Coeffs = (+0.0, +0.0)
  END_GROUP = TRENDING_COEFFS
  GROUP = STATE_000_RADIANCE
    B1_Rad_State_000 = (0.24, 0.26, 0.32, 0.31, 0.27, 0.26, 0.28, 0.29, 0.28, 0.27, 0.27, 0.30, 0.25, 0.24, 0.28, 0.26)
    B2_Rad_State_000 = (0.31, 0.34, 0.25, 0.35, 0.45, 0.12, 0.41, 0.27, 0.26, 0.25, 0.12, 0.37, 0.35, 0.38, 0.33, 0.48)
    B3_Rad_State_000 = (0.29, 0.11, 0.31, 0.34, 0.22, 0.28, 0.22, 0.21, 0.19, 0.20, 0.19, 0.21, 0.22, 0.24, 0.20, 0.24)
    B4_Rad_State_000 = (0.13, 0.11, -0.23, 0.13, -0.01, -0.19, -0.57, -0.08, -0.16, 0.07, -0.22, 0.06, 0.04, -0.04, -0.06, 0.11)
    B5_Rad_State_000 = (-0.09, -0.10, -0.07, -0.10, -0.09, -0.10, -0.09, -0.10, -0.09, -0.10, -0.09, -0.09, -0.09, -0.10, -0.09, -0.10)
    B7_Rad_State_000 = (-0.07, -0.06, -0.06, -0.06, -0.06, -0.06, -0.06, -0.05, -0.05, -0.05, -0.06, -0.05, -0.05, -0.06, -0.06, -0.06)
  END_GROUP = STATE_000_RADIANCE
  GROUP = STATE_001_RADIANCE
    B1_Rad_State_001 = (37.33, 32.39, 37.17, 32.83, 37.02, 32.81, 36.78, 32.70, 36.74, 32.76, 37.20, 32.58, 37.08, 32.58, 37.69,
32.46)
    B2_Rad_State_001 = (68.25, 57.62, 68.28, 58.23, 68.18, 58.66, 68.41, 58.20, 68.39, 58.21, 68.37, 58.64, 68.66, 58.14, 68.42,
58.27)
    B3_Rad_State_001 = (57.12, 58.81, 57.33, 58.88, 57.73, 58.73, 57.75, 58.52, 57.63, 58.74, 57.68, 58.86, 57.88, 58.90, 57.69,
59.16)
    B4_Rad_State_001 = (39.50, 40.18, 39.63, 40.25, 39.94, 39.28, 39.91, 40.47, 39.98, 40.13, 39.87, 40.46, 40.14, 40.49, 40.02,
40.44)
    B5_Rad_State_001 = (2.05, 2.07, 2.00, 2.03, 1.98, 2.05, 1.97, 2.03, 1.97, 2.02, 1.97, 2.03, 1.98, 2.03, 2.03, 2.08)
    B7_Rad_State_001 = (1.70, 2.12, 1.67, 2.11, 1.66, 2.10, 1.65, 2.10, 1.64, 2.08, 1.62, 2.08, 1.63, 2.07, 1.64, 2.11)
  END_GROUP = STATE_001_RADIANCE
  GROUP = STATE_010_RADIANCE
    B1_Rad_State_010 = (52.16, 46.95, 51.84, 47.58, 51.75, 47.48, 51.34, 47.32, 51.35, 47.35, 51.98, 47.07, 51.72, 47.01, 52.52,
46.69)
    B2_Rad_State_010 = (92.21, 76.47, 92.33, 77.19, 92.14, 77.58, 92.39, 76.94, 92.31, 77.04, 92.20, 77.38, 92.53, 76.76, 92.18,
76.89)
    B3_Rad_State_010 = (67.09, 67.60, 67.37, 67.65, 67.76, 67.37, 67.79, 67.06, 67.59, 67.27, 67.55, 67.45, 67.68, 67.45, 67.43,
67.68)
    B4_Rad_State_010 = (64.69, 61.85, 64.85, 61.93, 65.15, 60.51, 65.35, 62.24, 65.23, 61.73, 64.95, 62.19, 65.17, 62.23, 64.98,
62.03)
    B5_Rad_State_010 = (3.71, 3.67, 3.61, 3.61, 3.59, 3.62, 3.58, 3.61, 3.57, 3.59, 3.56, 3.59, 3.57, 3.60, 3.64, 3.69)
    B7_Rad_State_010 = (2.72, 2.68, 2.68, 2.66, 2.67, 2.65, 2.66, 2.65, 2.64, 2.63, 2.61, 2.62, 2.61, 2.61, 2.63, 2.66)
  END_GROUP = STATE_010_RADIANCE
  GROUP = STATE_011_RADIANCE
    B1_Rad_State_011 = (89.25, 79.09, 88.60, 80.18, 88.55, 79.96, 87.90, 79.82, 87.87, 79.79, 88.94, 79.34, 88.61, 79.27, 89.98,
78.80)
    B2_Rad_State_011 = (159.84, 133.57, 159.98, 134.88, 159.79, 135.70, 160.21, 134.64, 160.13, 134.78, 1597.4, 135.31, 160.58,
134.27, 160.00, 134.43)
    B3_Rad_State_011 = (123.81, 125.98, 124.34, 126.05, 125.12, 125.67, 125.24, 125.17, 124.78, 125.64, 124.83, 125.89, 125.09,
125.91, 124.59, 126.32)
    B4_Rad_State_011 = (103.68, 101.65, 104.00, 101.81, 104.53, 99.69, 105.01, 102.29, 104.76, 101.38, 104.34, 102.08, 104.72,
102.13, 104.35, 101.98)
    B5_Rad_State_011 = (5.83, 5.82, 5.70, 5.73, 5.65, 5.76, 5.64, 5.73, 5.63, 5.71, 5.62, 5.71, 5.64, 5.73, 5.73, 5.84)
    B7_Rad_State_011 = (4.46, 4.85, 4.39, 4.81, 4.38, 4.79, 4.37, 4.80, 4.34, 4.76, 4.30, 4.75, 4.29, 4.73, 4.32, 4.79)
  END_GROUP = STATE_011_RADIANCE
  GROUP = STATE_100_RADIANCE
    B1_Rad_State_100 = (63.82, 64.28, 63.49, 65.11, 63.53, 64.99, 63.01, 64.83, 63.06, 64.90, 63.80, 64.53, 63.52, 64.40, 64.53,
63.96)
    B2_Rad_State_100 = (114.69, 110.24, 114.83, 111.33, 114.72, 112.08, 115.09, 111.06, 114.94, 111.15, 114.88, 111.69, 115.30,
110.79, 114.73, 110.97)
    B3_Rad_State_100 = (89.03, 81.93, 89.28, 82.01, 89.69, 81.81, 89.83, 81.39, 89.35, 81.58, 89.39, 81.75, 89.60, 81.75, 89.28,
82.06)
    B4_Rad_State_100 = (76.69, 62.54, 76.80, 62.70, 77.04, 61.21, 77.33, 62.93, 77.09, 62.38, 76.96, 62.83, 77.15, 62.92, 76.94,
62.74)
    B5_Rad_State_100 = (3.38, 3.78, 3.29, 3.69, 3.27, 3.71, 3.25, 3.70, 3.25, 3.66, 3.24, 3.67, 3.25, 3.69, 3.32, 3.75)
    B7_Rad_State_100 = (4.04, 4.03, 3.97, 3.99, 3.96, 3.97, 3.95, 3.98, 3.91, 3.95, 3.89, 3.94, 3.88, 3.93, 3.91, 3.99)
  END_GROUP = STATE_100_RADIANCE
  GROUP = STATE_101_RADIANCE
    B1_Rad_State_101 = (100.50, 96.11, 100.04, 97.47, 99.85, 97.27, 99.14, 96.98, 99.12, 97.11, 100.28, 96.56, 99.88, 96.41,
101.44, 95.88)
    B2_Rad_State_101 = (182.51, 167.28, 182.57, 168.84, 182.45, 169.92, 182.98, 168.56, 182.78, 168.77, 182.40, 169.29, 183.46,
168.34, 182.66, 168.74)
    B3_Rad_State_101 = (145.88, 140.37, 146.33, 140.62, 147.00, 140.22, 147.28, 139.45, 146.72, 140.02, 146.82, 140.26, 147.10,
140.27, 146.63, 140.75)
    B4_Rad_State_101 = (115.36, 101.47, 115.62, 101.80, 115.99, 99.34, 116.35, 101.95, 116.34, 101.07, 115.89, 101.99, 116.29,
102.21, 115.98, 101.86)
    B5_Rad_State_101 = (5.55, 5.94, 5.44, 5.84, 5.37, 5.84, 5.35, 5.82, 5.35, 5.82, 5.34, 5.80, 5.36, 5.81, 5.42, 5.91)
    B7_Rad_State_101 = (5.76, 6.18, 5.67, 6.14, 5.67, 6.12, 5.66, 6.12, 5.62, 6.07, 5.57, 6.06, 5.55, 6.05, 5.58, 6.12)
  END_GROUP = STATE_101_RADIANCE
  GROUP = STATE_110_RADIANCE
    B1_Rad_State_110 = (115.49, 110.86, 115.05, 112.30, 114.65, 112.07, 113.88, 111.82, 113.87, 111.86, 115.28, 111.22, 114.64,
110.98, 116.39, 110.20)
    B2_Rad_State_110 = (206.04, 185.96, 206.16, 187.51, 206.03, 188.52, 206.64, 187.06, 206.43, 187.30, 205.75, 187.88, 207.01,
186.69, 206.12, 186.92)
    B3_Rad_State_110 = (155.82, 149.00, 156.33, 149.21, 157.05, 148.77, 157.23, 148.01, 156.62, 148.56, 156.72, 148.84, 156.95,
148.69, 156.40, 149.13)
    B4_Rad_State_110 = (140.29, 123.27, 140.31, 123.60, 140.79, 120.68, 141.16, 124.02, 141.15, 122.83, 140.57, 123.88, 140.99,
123.97, 140.62, 123.64)
```

```
    B5_Rad_State_110 = (7.19, 7.50, 7.04, 7.39, 6.96, 7.40, 6.95, 7.37, 6.93, 7.36, 6.91, 7.34, 6.93, 7.35, 7.02, 7.48)
    B7_Rad_State_110 = (6.77, 6.74, 6.67, 6.69, 6.66, 6.67, 6.65, 6.67, 6.60, 6.63, 6.55, 6.61, 6.52, 6.58, 6.55, 6.65)
  END_GROUP = STATE_110_RADIANCE
  GROUP = STATE_111_RADIANCE
    B1_Rad_State_111 = (142.50, 137.03, 141.59, 138.24, 141.14, 138.09, 141.05, 137.57, 141.65, 137.24, 141.90, 136.63, 141.87,
136.98, 142.71, 135.92)
    B2_Rad_State_111 = (273.39, 242.99, 273.05, 245.14, 273.54, 246.21, 274.31, 244.38, 273.88, 244.95, 272.72, 245.58, 274.88,
244.06, 273.95, 244.60)
    B3_Rad_State_111 = (212.29, 207.07, 213.20, 207.67, 214.28, 207.16, 214.48, 206.12, 213.74, 206.77, 213.81, 207.15, 214.26,
207.04, 213.43, 207.80)
    B4_Rad_State_111 = (179.00, 162.02, 179.06, 162.57, 179.92, 158.58, 180.21, 163.09, 180.59, 161.66, 179.75, 162.82, 180.48,
162.86, 179.86, 162.56)
    B5_Rad_State_111 = (9.28, 9.63, 9.08, 9.50, 8.99, 9.52, 8.97, 9.49, 8.96, 9.47, 8.94, 9.44, 8.96, 9.46, 9.06, 9.61)
    B7_Rad_State_111 = (8.49, 8.88, 8.39, 8.85, 8.39, 8.82, 8.37, 8.82, 8.30, 8.76, 8.24, 8.74, 8.20, 8.70, 8.22, 8.78)
  END_GROUP = STATE_111_RADIANCE
END_GROUP = LAMP_RADIANCE

GROUP = LAMP_REFERENCE
  Lmp_Rtemp = (+23.0, +23.0, +23.00, +23.0, -181.0, +23.0, +23.0, +23.0, +23.0, +23.0)
END_GROUP = LAMP_REFERENCE

GROUP = REFLECTIVE_IC_COEFFS
  B1_Coefficients_Detector1 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B1_Coefficients_Detector2 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B1_Coefficients_Detector3 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B1_Coefficients_Detector4 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B1_Coefficients_Detector5 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B1_Coefficients_Detector6 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B1_Coefficients_Detector7 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B1_Coefficients_Detector8 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B1_Coefficients_Detector9 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B1_Coefficients_Detector10 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B1_Coefficients_Detector11 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B1_Coefficients_Detector12 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B1_Coefficients_Detector13 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B1_Coefficients_Detector14 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B1_Coefficients_Detector15 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B1_Coefficients_Detector16 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B2_Coefficients_Detector1 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B2_Coefficients_Detector2 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B2_Coefficients_Detector3 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B2_Coefficients_Detector4 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B2_Coefficients_Detector5 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B2_Coefficients_Detector6 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B2_Coefficients_Detector7 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B2_Coefficients_Detector8 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B2_Coefficients_Detector9 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B2_Coefficients_Detector10 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B2_Coefficients_Detector11 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B2_Coefficients_Detector12 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B2_Coefficients_Detector13 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B2_Coefficients_Detector14 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B2_Coefficients_Detector15 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B2_Coefficients_Detector16 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B3_Coefficients_Detector1 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B3_Coefficients_Detector2 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B3_Coefficients_Detector3 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B3_Coefficients_Detector4 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B3_Coefficients_Detector5 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B3_Coefficients_Detector6 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B3_Coefficients_Detector7 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B3_Coefficients_Detector8 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B3_Coefficients_Detector9 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B3_Coefficients_Detector10 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B3_Coefficients_Detector11 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B3_Coefficients_Detector12 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B3_Coefficients_Detector13 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B3_Coefficients_Detector14 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B3_Coefficients_Detector15 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B3_Coefficients_Detector16 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B4_Coefficients_Detector1 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B4_Coefficients_Detector2 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B4_Coefficients_Detector3 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B4_Coefficients_Detector4 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B4_Coefficients_Detector5 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B4_Coefficients_Detector6 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B4_Coefficients_Detector7 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B4_Coefficients_Detector8 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B4_Coefficients_Detector9 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B4_Coefficients_Detector10 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B4_Coefficients_Detector11 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B4_Coefficients_Detector12 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B4_Coefficients_Detector13 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B4_Coefficients_Detector14 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B4_Coefficients_Detector15 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B4_Coefficients_Detector16 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B5_Coefficients_Detector1 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B5_Coefficients_Detector2 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B5_Coefficients_Detector3 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B5_Coefficients_Detector4 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B5_Coefficients_Detector5 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B5_Coefficients_Detector6 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B5_Coefficients_Detector7 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B5_Coefficients_Detector8 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B5_Coefficients_Detector9 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B5_Coefficients_Detector10 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B5_Coefficients_Detector11 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B5_Coefficients_Detector12 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B5_Coefficients_Detector13 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B5_Coefficients_Detector14 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B5_Coefficients_Detector15 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B5_Coefficients_Detector16 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B7_Coefficients_Detector1 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B7_Coefficients_Detector2 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B7_Coefficients_Detector3 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B7_Coefficients_Detector4 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B7_Coefficients_Detector5 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B7_Coefficients_Detector6 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B7_Coefficients_Detector7 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B7_Coefficients_Detector8 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B7_Coefficients_Detector9 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B7_Coefficients_Detector10 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B7_Coefficients_Detector11 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B7_Coefficients_Detector12 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B7_Coefficients_Detector13 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B7_Coefficients_Detector14 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B7_Coefficients_Detector15 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
  B7_Coefficients_Detector16 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
END_GROUP = REFLECTIVE_IC_COEFFS

GROUP = THERMISTOR_COEFFS
  Black_Body_Temp           = (17.073, 0.10263, 2.2576E-4, 0.0, 0.0, 0.0)
  Silicon_FP_Array_Temp      = (10.049, 0.83456E-1, 0.14176E-3, 0.0, 0.0, 0.0)
  Cal_Shutter_Flag_Temp      = (36.898, -0.1598, 1.957E-6, 0.0, 0.0, 0.0)
  Baffle_Temp               = (-2.9072, 0.089583, 2.7115E-4, 0.0, 0.0, 0.0)
  Cold_Stage_FP_Array_Temp   = (-162.94, -0.1000, 0.0, 0.0, 0.0, 0.0)
  Scan_Line_Corrector_Temp   = (147.84, -1.8384, 0.016092, -9.2715E-5, 2.839E-7, -3.683E-10)
  Cal_Shutter_Hub_Temp       = (121.23, -1.9147, 0.019275, -0.11865E-3, 0.37343E-6, -0.47899E-9)
  Relay_Optics_Temp         = (121.23, -1.9147, 0.019275, -0.11865E-3, 0.37343E-6, -0.47899E-9)
  Primary_Mirror_Temp        = (121.23, -1.9147, 0.019275, -0.11865E-3, 0.37343E-6, -0.47899E-9)
  Secondary_Mirror_Temp      = (121.23, -1.9147, 0.019275, -0.11865E-3, 0.37343E-6, -0.47899E-9)
END_GROUP = THERMISTOR_COEFFS

GROUP = FILL_PATTERNS
  Band_Fill_Pattern = (0, 255)
END_GROUP = FILL_PATTERNS

END
```

References

Please see http://landsat.usgs.gov/resources/acronyms.php for a list of acronyms.

Jet Propulsion Laboratory (JPL). California Institute of Technology. Pasadena, California. JPL-D-7669. Part 2. Planetary Data System Standards Reference. Revision 3.6. August 1, 2003.