Department of the Interior
U.S. Geological Survey

# LANDSAT 7 (L7) SYSTEM
# CALIBRATION PARAMETER FILE (CPF) DEFINITION

## Version 6.0

## May 2007

# LANDSAT 7 (L7) SYSTEM
# CALIBRATION PARAMETER FILE (CPF) DEFINITION

**May 2007**

Prepared By:

_____
E. Micijevic                    Date
Calibration Analyst
SAIC

Reviewed By:

_____
D. Strande                      Date
IAS Systems Engineer
SAIC

Reviewed By:

_____
R. Lamb                         Date
CalVal Systems Engineer
SAIC

Reviewed By:

_____
C. Engebretson                  Date
Software Project Lead
SAIC

Approved By:

_____
R. Hayes                        Date
CalVal Task Lead
SAIC

Approved By:

_____
C. Torbert                      Date
Landsat Ground Segment Manager
SAIC

Approved By:

_____
J. Lacasse                      Date
Mission Management Officer
USGS

EROS
Sioux Falls, South Dakota

IAS-207
                                      Version 6.0

# Executive Summary

This document describes the contents of the Calibration Parameter File (CPF) generated by the Enhanced Thematic Mapper Plus (ETM+) functionality of the Image Assessment System (IAS). The IAS periodically performs radiometric and geometric calibration and updates the CPF. This file is stamped with applicability dates and is sent to the Landsat Archive Manager (LAM) for storage and eventual bundling with outbound Level 0 Reformatted Products (L0Rp). The CPF is also sent to International Ground Stations (IGSs) via the Landsat 7 (L7) Mission Operations Center (MOC). The CPF supplies the radiometric and geometric correction parameters required during Level 1 (L1) processing to create superior products of uniform consistency across the L7 system.

# Document History

| Document Number | Document Version | Publication Date | Change Number |
|---|---|---|---|
| 430-15-01-002-0 | Version 1.0 | February 1998 | |
| 430-15-01-002-2 | Version 2.0 | July 1998 | IAS980070<br>IAS980071<br>IAS980078<br>IAS980080<br>IAS980098 |
| 430-15-01-002-3 | Version 3.0 | June 1999 | GS CCR 60<br>GS CCR 106<br>GS CCR 110 |
| IAS-207 | Version 4.0 | January 2000 | DHF CCR 1171 |
| IAS-207 | Version 5.0 | August 2005 | CCR 1819<br>CCR 3921 |
| IAS-207 | Version 6.0 | May 2007 | CCR 4788 |

# Contents

# List of Tables

# Section 1    Introduction

This document describes the contents of the Calibration Parameter File (CPF) generated by the Image Assessment System (IAS). The Landsat 7 (L7) functionality of the IAS is responsible for offline assessment of image quality to ensure compliance with the radiometric and geometric requirements of the L7 spacecraft and the Enhanced Thematic Mapper Plus (ETM+) sensor throughout the life of the mission.

In addition to its assessment functions, the IAS is responsible for the radiometric and geometric calibration of the L7 satellite and ETM+ sensor. The IAS periodically performs radiometric and geometric calibration and updates the CPF. This file is stamped with applicability dates and is archived at the U.S. Geological Survey (USGS) Center for Earth Resources Observation and Science (EROS) and eventually bundled with outbound Level Zero Reformatted Products (L0Rp). The CPF is also sent to the Landsat 7 Processing System (LPS), Level 1 (L1) production systems within EROS, and to International Ground Stations (IGSs) via the L7 Mission Operations Center (MOC). The CPF supplies the radiometric and geometric correction parameters required during L1 processing to create superior products of uniform consistency across the L7 system.

## 1.1    Document Organization

Section 1 introduces the CPF. It describes the CPF structure and language, the CPF updates, time stamps, and file-naming conventions, as well as the attributes used to characterize the calibration parameters.
Section 2 contains a table that lists and describes the CPF parameters. The actual prelaunch and postlaunch CPFs contain the most recent and accurate values available for these parameters.
Section 3 presents the syntax of the CPF Object Description Language (ODL) and provides a CPF example to illustrate the actual appearance of the file.

## 1.2    File Structure

All parameters are stored as American Standard Code for Information Interchange (ASCII) text using the ODL syntax developed by 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, in certain cases, provide for the handling of lexical elements (for example, building blocks) that are included in the Consultative Committee for Space Data Systems (CCSDS) specification of the Parameter Value Language (PVL). PVL 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 contains ODL details.

## 1.3    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, a CPF delivery also includes new versions of all CPFs for time periods affected by the most recent calibration update. Only the most recent available CPFs should be used in ETM+ 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 the time periods shorter than a calendar quarter.

Following the ETM+ switch to bumper operational mode (April 1, 2007), multiple version updates can be expected during any given quarter due to a hardly predictive nature of the scanning mirror bumper parameters. The irregular (mid-quarter) updates do not affect the three-month CPF release schedule.

### 1.3.1    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.

### 1.3.2    File-Naming Conventions

Through the course of the mission, a serial collection of CPFs is generated and sent to the Landsat Archive Manager (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 due to file error. The need for unique file version numbers becomes necessary as file contents change. The unique 00 version number is reserved for the original CPF, created before the satellite's launch. Version numbers for all quarterly CPFs released after the launch begin with 01.

The IAS uses the following file-naming procedure to name the CPF:

$$L7CPFy_1y_1y_1y_1m_1m_1d_1d_1\_y_2y_2y_2y_2m_2m_2d_2d_2.nn$$

where        L7   =   constant for Landsat 7

           CPF   =   3-letter CPF designator

       $y_1y_1y_1y_1$   =   4-digit effectivity starting year

$m_1m_1$ = 2-digit effectivity starting month

$d_1d_1$ = 2-digit effectivity starting day

_ = effectivity starting/ending date separator

$y_2y_2y_2y_2$ = 4-digit effectivity ending year

$m_2m_2$ = 2-digit effectivity ending month

$d_2d_2$ = 2-digit effectivity ending day

. = Ending day/version number separator

nn = version number for this file (starts with 01)

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    L7CPF20000101_20000331.01
          L7CPF20000101_20000331.02
          L7CPF20000101_20000331.03
File 2    L7CPF20000401_20000630.01
          L7CPF20000401_20000630.02
File 3    L7CPF20000701_20000930.01
          L7CPF20000701_20000930.02
File 4    L7CPF20001001_20001231.01
```

This example assumes that the effective date ranges do 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 remains unchanged. A post-problem CPF with a new file name would be created with the Effective_Date_Begin corresponding to the imaging date when the problem occurred and the Effective_Date_End corresponding to the original Effective_Date_End for the period under consideration. Both new CPFs, although they appeared for the first time for given effective dates, would a have version number for one higher than the CPF for the quarter they originated from. New versions of all other CPFs affected by the updated parameters also would be created.

Suppose, for example, that it was discovered that a detector stopped responding on July 25, 2000. Two new CPFs need to be created that supersede the period represented by file number three, version 2, and a new version of file number four. The new file names and version numbers become:

```
File 3    L7CPF20000701_20000930.01
          L7CPF20000701_20000930.02
```

L7CPF20000701_20000725.03
L7CPF20000726_20000930.03
File 4    L7CPF20001001_20001231.01
L7CPF20001001_20001231.02

## 1.4    File Content Description

Table 2-1**Error! Reference source not found.** lists all CPF parameters. Within this table, each parameter entry is characterized by five attributes:

1. Parameter Group—identifies a related set of parameters.
2. Parameter Name—uniquely identifies and describes the content of each parameter.
3. Value Type—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—referred to using Hierarchical Data Format (HDF) number type nomenclature, type#, where type is either 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. Table 1-1 shows 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 1-1. Data Types in CPF*

5. Description—briefly describes the parameter, its format, and its nominal, expected, or sample 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.

IAS-207
Version 6.0

# Section 2    CPF Parameters

Table 2-1 lists the L7 CPF parameters.

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| FILE_ATTRIBUTES | Spacecraft_Name<br><br>(available in all CPFs with effective dates of January 1, 2007 and thereafter) | Static | char8 | Descriptor used to identify the spacecraft for which the calibration parameters are applicable. Valid format: Landsat_7 |
| FILE_ATTRIBUTES | Sensor_Name<br><br>(available in all CPFs with effective dates of January 1, 2007 and thereafter) | Static | char8 | Descriptor used to identify the sensor for which the calibration parameters are applicable. Valid format: Enhanced_Thematic_Mapper_Plus |
| FILE_ATTRIBUTES | Effective_Date_Begin | Dynamic | char8 | Effective start date for this file Valid format: yyyy-mm-dd, where yyyy = 1998-2050, 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 = 1998-2050, mm = 01-12, and dd = 01-31 |
| FILE_ATTRIBUTES | CPF_File_Name | Dynamic | char8 | Original file name assigned by IAS Valid format: L7CPFyyyymmdd-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.NNN, where NNNNNNN.NNN = 6378137.000 |
| 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 Earth's deviation from an exact sphere (WGS84 standard) Valid format: N.NNNNNNNNNNNNNNN, where N.NNNNNNNNNNNNNNN = 0.00335281066474 = (1/298.257223563) |
| EARTH_CONSTANTS | Eccentricity | Static | float64 | Number describing the Earth ellipsoid eccentricity squared (WGS84 standard) Valid format: N.NNNNNNNNNNNNNNN, 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 times the mass of the Earth. This parameter is given in units of meters cubed per second squared (m3/s2). Valid format: N.NNNNNNNENN, 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 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| ORBIT_PARAMETERS | WRS_Cycle_Days | Static | uint8 | Time period, in days, required for the satellite to view Earth once<br>Valid format: NN, where NN = 16 |
| ORBIT_PARAMETERS | WRS_Cycle_Orbits | Static | uint8 | Number of orbits or paths in a complete World Reference System (WRS) cycle<br>Valid format: NNN, where NNN = 233 |
| ORBIT_PARAMETERS | Scenes_Per_Orbit | Static | uint8 | Number of scenes or row locations per orbit<br>Valid format: NNN, where NNN = 248 |
| ORBIT_PARAMETERS | Orbital_Period | Static | float64 | Time required, in seconds, to complete one orbit<br>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<br>Valid format: NN.NNNNNNNEN, where NN.NNNNNNNEN = 53.136250E9 |
| ORBIT_PARAMETERS | Orbit_Radius | Static | float64 | Nominal distance in km from the Earth's center to the spacecraft track<br>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<br>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<br>Valid format: NNNN.NNNN, where NNNN.NNNN = 7083.4408 |
| ORBIT_PARAMETERS | Orbit_Eccentricity | Static | float64 | Nominal eccentricity of the satellite's orbit<br>Valid format: N.NNNNNNNN, where N.NNNNNNNN = 0.00117604 |
| ORBIT_PARAMETERS | Inclination_Angle | Static | float64 | Angle in degrees formed by Earth's equatorial and satellite plane<br>Valid format: NN.NNNN, where NN.NNNN = 98.2096 |
| ORBIT_PARAMETERS | Argument_Of_Perigee | Static | float32 | Nominal angle in degrees of point nearest Earth in orbit as measured from ascending node in the direction of satellite motion<br>Valid format: NN.N, where NN.N = 90.0 |
| ORBIT_PARAMETERS | Descending_Node_ Row | Static | uint8 | Row corresponding to the Earth's equator<br>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)<br>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 a.m. hours and minutes<br>Valid format: HH:MM, where HH:MM = 09:45 |
| ORBIT_PARAMETERS | Descending_Node_ Time_Max | Static | char8 | Maximum local solar time of descending node in a.m. hours and minutes<br>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<br>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<br>Valid format: NN, where NN = 16 |
| SCANNER_ PARAMETERS | Lines_Per_Scan_60 | Static | uint8 | Detectors per scan for Band 6<br>Valid format: N, where N = 8 |
| SCANNER_ PARAMETERS | Lines_Per_Scan_15 | Static | uint8 | Detectors per scan for Band 8<br>Valid format: NN, where NN = 32 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| SCANNER_ PARAMETERS | Scans_Per_Scene | Static | int16 | Scans per nominal WRS scene<br>Valid format: NNN, where NNN = 375 |
| SCANNER_ PARAMETERS | Swath_Angle | Dynamic | float32 | Object space angle in radians of scan mirror travel during active scan time<br>Valid format: N.NNNNN, where N.NNNNN = 0.26868 (after measurement of as-built ETM+) |
| SCANNER_ PARAMETERS | Scan_Rate | Static | float32 | Angular scan velocity in radians per second of scan mirror<br>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<br>Valid format: N.NNNNNNN, where N.NNNNNNN = 9.6110206 |
| SCANNER_ PARAMETERS | Dwell_Time_60 | Static | float64 | Detector sample time in microseconds for Band 6<br>Valid format: N.NNNNNN, where N.NNNNNN = 19.222041 |
| SCANNER_ PARAMETERS | Dwell_Time_15 | Static | float64 | Detector sample time in microseconds for Band 8<br>Valid format: N.NNNNNNN, where N.NNNNNNN = 4.8055103 |
| SCANNER_ PARAMETERS | IC_Line_Length_30 | Static | int16 | Nominal number of detector samples for the Internal Calibrator (IC) for Bands 1-5 and 7<br>Valid format: NNNN, where NNNN = 1150 |
| SCANNER_ PARAMETERS | IC_Line_Length_60 | Static | int16 | Nominal number of detector samples for the internal calibrator for Band 6<br>Valid format: NNN, where NNN = 575 |
| SCANNER_ PARAMETERS | IC_Line_Length_15 | Static | int16 | Nominal number of detector samples for the internal calibrator for Band 8<br>Valid format: NNNN, where NNNN = 2300 |
| SCANNER_ PARAMETERS | Scan_Line_Length_30 | Static | int16 | Nominal number of detector samples during active scan time for Bands 1-5 and 7<br>Valid format: NNNN, where NNNN = 6320 |
| SCANNER_ PARAMETERS | Scan_Line_Length_60 | Static | int16 | Nominal number of detector samples during active scan time for Band 6<br>Valid format: NNNN, where NNNN = 3160 |
| SCANNER_ PARAMETERS | Scan_Line_Length_15 | Static | int16 | Nominal number of detector samples during active scan time for Band 8<br>Valid format: NNNNN, where NNNNN = 12640 |
| SCANNER_ PARAMETERS | Filter_Frequency_30 | Static | float32 | Bandwidth in kHz of detector presample filter (defined by 3-dB roll-off point) for Bands 1-5 and 7<br>Valid format: NN.NN, where NN.NN = 52.02 |
| SCANNER_ PARAMETERS | Filter_Frequency_60 | Static | float32 | Bandwidth in kHz of detector presample filter (defined by 3-dB roll-off point) for Band 6<br>Valid format: NN.NN, where NN.NN = 26.01 |
| SCANNER_ PARAMETERS | Filter_Frequency_15 | Static | float32 | Bandwidth in kHz of detector presample filter (defined by 3-dB roll-off point) for Band 8<br>Valid format: NNN.NN, where NNN.NN = 115.00 |
| 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<br>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<br>Valid format: NN.N, where NN.N = 39.4 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| 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<br>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 scanning motion is stopped<br>Valid format: NN.N, where NN.N = 85.0 |
| SCANNER_ PARAMETERS | IFOV_B8_along_scan | Static | float32 | Along-scan angle in µrad subtended by a Band 8 detector when the scanning motion is stopped<br>Valid format: NN.N, where NN.N = 18.5 |
| SCANNER_ PARAMETERS | IFOV_B8_across_ scan | Static | float32 | Across-scan angle in µrad subtended by a Band 8 detector when scanning motion is stopped<br>Valid format: NN.NN, where NN.NN = 21.25 |
| SCANNER_ PARAMETERS | Scan_Period | Static | float64 | Time in milliseconds of a complete scan cycle, including forward and reverse scans<br>Valid format: NNN.NN, where NNN.NN = 143.58 |
| SCANNER_ PARAMETERS | Scan_Frequency | Static | float32 | Number of scans in 1 second (Hz)<br>Valid format: N.NNNNN, where N.NNNNN = 6.96476 |
| SCANNER_ PARAMETERS | Active_Scan_Time | Static | float32 | Time in µs required for the scan mirror to travel from its scan-line-start to End-Of-Line (EOL)<br>Valid format: NNNNN.NNN, where NNNNN.NNN = 60743.346 |
| SCANNER_ PARAMETERS | Turn_Around_Time | Static | float32 | Time in milliseconds from EOL to next scan-line-start, during which scan mirror motion reverses direction<br>Valid format: NN.NNN, where:<br>NN.NNN = 11.055 |
| SPACECRAFT_ PARAMETERS | ADS_Interval | Static | float32 | Time in milliseconds between Attitude Displacement Sensors (ADS) samples<br>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 roll axis measurement<br>Valid format: N.NNN, where N.NNN = 0.375 |
| SPACECRAFT_ PARAMETERS | ADS_Yaw_Offset | Static | float32 | Amount of time in milliseconds from the start of a PCD cycle to the yaw axis measurement<br>Valid format: N.NNN, where N.NNN = 0.875 |
| SPACECRAFT_ PARAMETERS | ADS_Pitch_Offset | Static | float32 | Amount of time in milliseconds from the start of a PCD cycle to the pitch axis measurement<br>Valid format: N.NNN, where N.NNN = 1.375 |
| SPACECRAFT_ PARAMETERS | Data_Rate | Static | float32 | ETM+ output bit rate in Mbps<br>Valid format: NN.NNN, where NN.NNN = 74.914 |
| GROUP: MIRROR_PARAMETERS | Error_Conversion_ Factor | Static | float32 | First half and second half scan mirror error measurement units in microseconds<br>Valid format: N.NNNNNNNN, where N.NNNNNNNN = 0.18845139 (5.306437 MHz) |
| GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_SAM | Forward_Along_ SME1_SAM | Static | 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<br>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 | Static | 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<br>Valid format for each term: SN.NNNNNNESNN, where S = "+" or "-", N = 0 to 9, and E = "E" |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_SAM | Forward_Angle1_ SME1_SAM | Static | float32 | Angle in µrad from the start of the scan to the mid-scan point in forward direction; SAM mode with SME number 1<br>Valid format: NNNNN.N, where NNNNN.N = 67166.9 |
| GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_SAM | Forward_Angle2_ SME1_SAM | Static | float32 | Angle in µrad from the mid-scan point to the end of the scan in forward direction; SAM mode with SME number 1<br>Valid format: NNNNN.N, where NNNNN.N = 67145.9 |
| GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_SAM | Reverse_Along_ SME1_SAM | Static | 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<br>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 | Static | 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<br>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 | Static | float32 | Angle in µrad from the start of the scan to the mid-scan point in reverse direction; SAM mode with SME number 1<br>Valid format: NNNNN.N, where NNNNN.N = 67142.8 |
| GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_SAM | Reverse_Angle2_ SME1_SAM | Static | float32 | Angle in µrad from the mid-scan point to the end of the scan in reverse direction; SAM mode with SME number 1<br>Valid format: NNNNN.N, where NNNNN.N = 67169.9 |
| GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_SAM | Forward_Along_ SME2_SAM | Static | 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<br>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_ SME2_SAM | Static | 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<br>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_ SME2_SAM | Static | float32 | Angle in µrad from the start of the scan to mid-scan point in forward direction; SAM mode with SME number 2<br>Valid format: NNNNN.N, where NNNNN.N = 67162.7 |
| GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_SAM | Forward_Angle2_ SME2_SAM | Static | float32 | Angle in µrad from the mid-scan point to the end of the scan in forward direction; SAM mode with SME number 2<br>Valid format: NNNNN.N, where NNNNN.N = 67162.8 |
| GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_SAM | Reverse_Along_ SME2_SAM | Static | 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 2<br>Valid format for each term: SN.NNNNNNESNN, where S = "+" or "-", N = 0 to 9, and E = "E" |

IAS-207
Version 6.0

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_SAM | Reverse_Cross_ SME2_SAM | Static | 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 2<br>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 | Static | float32 | Angle in µrad from the start of the scan to the mid-scan point in reverse direction; SAM mode with SME number 2<br>Valid format: NNNNN.N, where NNNNN.N = 67162.8 |
| GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_SAM | Reverse_Angle2_ SME2_SAM | Static | float32 | Angle in µrad from the mid-scan point to the end of the scan in reverse direction; SAM mode with SME number 2<br>Valid format: NNNNN.N, where NNNNN.N = 67162.7 |
| GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_BUMP | Forward_Along_ SME1_Bump | Static | 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<br>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 | Static | 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<br>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_Angle1_ SME1_Bump | **For CPFs with effective dates prior to April 1, 2007** | | |
| | | Static | float32 | Angle in µrad from the start of the scan to the mid-scan point in forward direction; bumper mode with SME number 1.<br>Valid format: NNNNN.N, where NNNNN.N = 67156.3 |
| | | **For CPFs with effective dates of April 1, 2007 and thereafter** | | |
| | | Dynamic | float32 array of flexible length | Angle in µrad from the start of the scan to the mid-scan point in forward direction; bumper mode with SME number 1. The array contains daily values over one CPF interval.<br>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 April 1, 2007** | | |
| | | Static | float32 | Angle in µrad from the mid-scan point to the end of the scan in forward direction; bumper mode with SME number 1.<br>Valid format: NNNNN.N, where NNNNN.N = 67156.7 |
| | | **For CPFs with effective dates of April 1, 2007 and thereafter** | | |
| | | Dynamic | float32 array of flexible length | Angle in µrad from the mid-scan point to the end of the scan in forward direction; bumper mode with SME number 1. The array contains daily values over one CPF interval.<br>Valid format for each term: NNNNN.N, where N = 0 to 9. |
| GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_BUMP | Forward_FHSERR_SME1 _Bump<br><br>(available in all CPFs with effective dates of April 1, 2007 and thereafter) | Dynamic | int16 array of flexible length | First-half error of the forward scan angle; bumper mode with SME number 1. The array contains daily values over one CPF interval.<br>Valid format for each term: SNNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_BUMP | Forward_SHSERR_SME1 _Bump<br><br>(available in all CPFs with effective dates of April 1, 2007 and thereafter) | Dynamic | int16 array of flexible length | Second-half error of the forward scan angle; bumper mode with SME number 1. The 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 | Static | float64 array (6 values) | Fifth-order polynomial coefficients that describe the deviation of reverse along the scan mirror motion from linear; bumper mode with SME number 1<br>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 | Static | 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<br>Valid format: SN.NNNNNNESNN, where S = "+" or "-", N = 0 to 9, and E = "E" |
| GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_BUMP | Reverse_Angle1_ SME1_Bump | **For CPFs with effective dates prior to April 1, 2007** | | |
| | | Static | float32 | Angle in µrad from the start of the scan to the mid-scan point in reverse direction; Bumper mode with SME number 1.<br>Valid format: NNNNN.N where NNNNN.N = 67156.7 |
| | | **For CPFs with effective dates of April 1, 2007 and thereafter** | | |
| | | Dynamic | float32 array of flexible length | Angle in µrad from the start of the scan to the mid-scan point in reverse direction; Bumper mode with SME number 1. Array contains daily values over one CPF interval.<br>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 April 1, 2007** | | |
| | | Static | float32 | Angle in µrad from the mid-scan point to the end of the scan in reverse direction; Bumper mode with SME number 1.<br>Valid format: NNNNN.N where NNNNN.N = 67156.3 |
| | | **For CPFs with effective dates of April 1, 2007 and thereafter** | | |
| | | Dynamic | float32 array of flexible length | Angle in µrad from the mid-scan point to the end of the scan in reverse direction; Bumper mode with SME number 1. The array contains daily values over one CPF interval.<br>Valid format for each term: NNNNN.N, where N = 0 to 9. |
| GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME1_BUMP | Reverse_FHSERR_SME1 _Bump<br><br>(available in all CPFs with effective dates of April 1, 2007 and thereafter) | Dynamic | int16 array of flexible length | First-half error of the reverse scan angle; Bumper mode with SME number 1. The 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<br><br>(available in all CPFs with effective dates of April 1, 2007 and thereafter) | Dynamic | int16 array of flexible length | Second-half error of the reverse scan angle; bumper mode with SME number 1. The 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 | Static | float64 array (6 values) | Fifth-order polynomial coefficients that describe 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" |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_BUMP | Forward_Cross_ SME2_Bump | Static | float64 array (6 values) | Fifth-order polynomial coefficients that describe the deviation of the 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" |
| GROUP: MIRROR_PARAMETERS GROUP: ANGLES_SME2_BUMP | Forward_Angle1_ SME2_Bump | **For CPFs with effective dates prior to April 1, 2007** | | |
| | | Static | float32 | Angle in µrad from the start of the scan to the mid-scan point in forward direction; bumper mode with SME number 2. Valid format: NNNNN.N where NNNNN.N = 67162.7 |
| | | **For CPFs with effective dates of April 1, 2007 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 April 1, 2007** | | |
| | | Static | float32 | Angle in µrad from the mid-scan point to the end of the scan in forward direction; bumper mode with SME number 2. Valid format: NNNNN.N where NNNNN.N = 67162.8 |
| | | **For CPFs with effective dates of April 1, 2007 and thereafter** | | |
| | | Dynamic | float32 array of flexible length | Angle in µrad from the mid-scan point to the end of the scan in 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 April 1, 2007 and thereafter) | Dynamic | int16 array of flexible length | First-half error of the forward scan angle; bumper mode with SME number 2. The 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 April 1, 2007 and thereafter) | Dynamic | int16 array of flexible length | Second-half error of the forward scan angle; bumper mode with SME number 2. The 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 | Static | 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 | Static | 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 Group | 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 April 1, 2007** | | |
| | | 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 is NNNNN.N where NNNNN.N = 67162.8 |
| | | **For CPFs with effective dates of April 1, 2007 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. 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_Angle2_ SME2_Bump | **For CPFs with effective dates prior to April 1, 2007** | | |
| | | 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 is NNNNN.N where NNNNN.N = 67162.7 |
| | | **For CPFs with effective dates of April 1, 2007 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 (available in all CPFs with effective dates of April 1, 2007 and thereafter) | Reverse_FHSERR_SME2 _Bump | Dynamic | int16 array of flexible length | First-half error of the reverse scan angle; Bumper mode with SME number 2. The 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 (available in all CPFs with effective dates of April 1, 2007 and thereafter) | Reverse_SHSERR_SME2 _Bump | Dynamic | int16 array of flexible length | Second-half error of the reverse scan angle; bumper mode with SME number 2. The 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 (available in all CPFs with effective dates of April 1, 2007 and thereafter) | SME1_BumperA_Dwell_ Time | 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. The array contains daily values over one CPF interval. Valid format for each term: NNNNN.NN, where N = 0 to 9 |
| GROUP: BUMPER_MODE_ PARAMETERS (available in all CPFs with effective dates of April 1, 2007 and thereafter) | SME1_BumperA_Pickoff_ Time | 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. The array contains daily values over one CPF interval. Valid format for each term: NNNNN.NN, where N = 0 to 9 |
| GROUP: BUMPER_MODE_ PARAMETERS (available in all CPFs with effective dates of April 1, 2007 and thereafter) | SME1_BumperA_Offset_ Time | Static | float32 | "Physical" bumper mode mirror model parameter - time from bumper A pickoff signal to the start of the reverse active scan in microseconds. Valid format: NNNNN.NN, where NNNNN.NN = 10110.00 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: BUMPER_MODE_ PARAMETERS | SME1_BumperA_Angle<br><br>(available in all CPFs with effective dates of April 1, 2007 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.<br>Valid format: SNNNNN.N, where SNNNNN.N = -68665.0 |
| GROUP: BUMPER_MODE_ PARAMETERS | SME1_BumperB_Dwell_ Time<br><br>(available in all CPFs with effective dates of April 1, 2007 and thereafter) | Dynamic | float32 array of flexible length | "Physical" bumper mode mirror model parameter - time from bumper B pickoff signal to the start of the forward scan linear motion in microseconds. The array contains daily values over one CPF interval.<br>Valid format for each term: NNNNN.NN, where N = 0 to 9 |
| GROUP: BUMPER_MODE_ PARAMETERS | SME1_BumperB_Pickoff_ Time<br><br>(available in all CPFs with effective dates of April 1, 2007 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. The array contains daily values over one CPF interval.<br>Valid format for each term: NNNNN.NN, where N = 0 to 9 |
| GROUP: BUMPER_MODE_ PARAMETERS | SME1_BumperB_Offset_ Time<br><br>(available in all CPFs with effective dates of April 1, 2007 and thereafter) | Static | float32 | "Physical" bumper mode mirror model parameter - time from bumper B pickoff signal to the start of the forward active scan in microseconds.<br>Valid format: NNNNN.NN, where NNNNN.NN = 10110.00 |
| GROUP: BUMPER_MODE_ PARAMETERS | SME1_BumperB_Angle<br><br>(available in all CPFs with effective dates of April 1, 2007 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.<br>Valid format: SNNNNN.N, where SNNNNN.N = 68607.0 |
| GROUP: BUMPER_MODE_ PARAMETERS | SME2_BumperA_Dwell_ Time<br><br>(available in all CPFs with effective dates of April 1, 2007 and thereafter) | Dynamic | float32 array of flexible length | "Physical" bumper mode mirror model parameter - time from bumper A pickoff signal to the start of the reverse scan linear motion in microseconds. The array contains daily values over one CPF interval.<br>Valid format for each term: NNNNN.NN, where N = 0 to 9 |
| GROUP: BUMPER_MODE_ PARAMETERS | SME2_BumperA_Pickoff_ Time<br><br>(available in all CPFs with effective dates of April 1, 2007 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. The array contains daily values over one CPF interval.<br>Valid format for each term: NNNNN.NN, where N = 0 to 9 |
| GROUP: BUMPER_MODE_ PARAMETERS | SME2_BumperA_Offset_ Time<br><br>(available in all CPFs with effective dates of April 1, 2007 and thereafter) | Static | float32 | "Physical" bumper mode mirror model parameter - time from bumper A pickoff signal to the start of the reverse active scan in microseconds.<br>Valid format: NNNNN.NN, where NNNNN.NN = 10110.00 |
| GROUP: BUMPER_MODE_ PARAMETERS | SME2_BumperA_Angle<br><br>(available in all CPFs with effective dates of April 1, 2007 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.<br>Valid format: SNNNNN.N, where SNNNNN.N = -68665.0 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: BUMPER_MODE_ PARAMETERS | SME2_BumperB_Dwell_ Time<br><br>(available in all CPFs with effective dates of April 1, 2007 and thereafter) | Dynamic | float32 array of flexible length | "Physical" bumper mode mirror model parameter - time from bumper B pickoff signal to the start of the forward scan linear motion in microseconds. The array contains daily values over one CPF interval.<br>Valid format for each term: NNNNN.NN, where N = 0 to 9 |
| GROUP: BUMPER_MODE_ PARAMETERS | SME2_BumperB_Pickoff_ Time<br><br>(available in all CPFs with effective dates of April 1, 2007 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 bumper B pickoff signal in microseconds. The array contains daily values over one CPF interval.<br>Valid format for each term: NNNNN.NN, where N = 0 to 9 |
| GROUP: BUMPER_MODE_ PARAMETERS | SME2_BumperB_Offset_ Time<br><br>(available in all CPFs with effective dates of April 1, 2007 and thereafter) | Static | float32 | "Physical" bumper mode mirror model parameter - time from bumper B pickoff signal to the start of the forward active scan in microseconds.<br>Valid format: NNNNN.NN, where NNNNN.NN = 10110.00 |
| GROUP: BUMPER_MODE_ PARAMETERS | SME2_BumperB_Angle<br><br>(available in all CPFs with effective dates of April 1, 2007 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.<br>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<br>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<br>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<br>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<br>Valid format for each term: N.NNNNN, where N = 0 to 9 |
| GROUP: SCAN_LINE_CORRECTOR | Unpowered_Pointing_Bias<br><br>(available in all CPFs with effective dates of July 14, 2003 and thereafter) | Dynamic | Float32 | The best estimate of the pointing angle of the scan line corrector in its unpowered, "at-rest" pointing position<br>Valid format: N.NNNNNNN, where N.NNNNNNN = 0.0000427 |
| GROUP: FOCAL_PLANE_ PARAMETERS<br>GROUP: BAND_OFFSETS | Along_Scan_Band_ Offsets | Static | float32 array (8 values) | Nominal displacement in µrad from the center of the focal plane to each band's optical axis<br>Valid format: SNNNN.NNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS<br>GROUP: BAND_OFFSETS | Across_Scan_Band_ Offsets | Static | float32 array (8 values) | Nominal displacement in µrad from the center of the focal plane to each band's scan motion axis<br>Valid format: SNNNN.NNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS<br>GROUP: BAND_OFFSETS | Forward_Focal_ Plane_Offsets | Static | float32 array (8 values) | Offset in Instrument Fields of View (IFOVs) for focal plane forward scans<br>Valid format: SNNN.N, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: FOCAL_PLANE_ PARAMETERS<br>GROUP: BAND_OFFSETS | Reverse_Focal_ Plane_Offsets | Static | float32 array (8 values) | Offset in IFOVs for focal plane reverse scans<br>Valid format: SNNN.N, where S = "+" or "-" and N = 0 to 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS<br>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<br>Valid format: N.NNN, where N = 0 TO 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS<br>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<br>Valid format: N.NNN, where N = 0 TO 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS<br>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<br>Valid format: N.NNN, where N = 0 TO 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS<br>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<br>Valid format: N.NNN, where N = 0 TO 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS<br>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<br>Valid format: N.NNN, where N = 0 TO 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS<br>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<br>Valid format: N.NNN, where N = 0 TO 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS<br>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<br>Valid format: N.NNN, where N = 0 TO 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS<br>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<br>Valid format: N.NNN, where N = 0 TO 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS<br>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<br>Valid format: N.NNN, where N = 0 TO 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS<br>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<br>Valid format: N.NNN, where N = 0 TO 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS<br>GROUP: DETECTOR_OFFSETS | Forward_Along_ Scan_DO_B6 | Static | float32 array (8 values) | Forward along scan detector offsets in IFOV for each detector in Band 6<br>Valid format: N.NNN, where N = 0 TO 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS<br>GROUP: DETECTOR_OFFSETS | Reverse_Along_ Scan_DO_B6 | Static | float32 array (8 values) | Reverse along scan detector offsets in IFOV for each detector in Band 6<br>Valid format: N.NNN, where N = 0 TO 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS<br>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<br>Valid format: N.NNN, where N = 0 TO 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS<br>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<br>Valid format: N.NNN, where N = 0 TO 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS | Forward_Along_ Scan_DO_B8 | Static | float32 array (32 values) | Forward along scan detector offsets in IFOV for each detector in Band 8 Valid format: N.NNN, where N = 0 TO 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS | Reverse_Along_ Scan_DO_B8 | Static | float32 array (32 values) | Reverse along scan detector offsets in IFOV for each detector in Band 8 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 |
| 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_ DO_B6 | Static | float32 array (8 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 (8 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 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| 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: DETECTOR_OFFSETS | Forward_Across_ Scan_DO_B8 | Static | float32 array (32 values) | Forward across scan detector offsets in IFOV for each detector in Band 8 Valid format: N.NNN, where N = 0 TO 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS GROUP: DETECTOR_OFFSETS | Reverse_Across_ Scan_DO_B8 | Static | float32 array (32 values) | Reverse across scan detector offsets in IFOV for each detector in Band 8 Valid format: N.NNN, where N = 0 TO 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS GROUP: ODD_EVEN_OFFSETS | Forward_Even_ Detector_Shift | Static | float32 array (8 values) | Adjustments in IFOVs to compensate for forward band offsets, even detector layout geometry and multiplexer sampling for Bands 1 through 8 Valid format: NNN.N, where N = 0 TO 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS GROUP: ODD_EVEN_OFFSETS | Forward_Odd_ Detector_Shift | Static | float32 array (8 values) | Adjustments in IFOVs to compensate for forward band offsets, odd detector layout geometry and multiplexer sampling for Bands 1 through 8 Valid format: NNN.N, where N = 0 TO 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS GROUP: ODD_EVEN_OFFSETS | Reverse_Even_ Detector_Shift | Static | float32 array (8 values) | Adjustments in IFOVs to compensate for reverse band offsets, even detector layout geometry and multiplexer sampling for Bands 1 through 8 Valid format: NNN.N, where N = 0 TO 9 |
| GROUP: FOCAL_PLANE_ PARAMETERS GROUP: ODD_EVEN_OFFSETS | Reverse_Odd_ Detector_Shift | Static | float32 array (8 values) | Adjustments in IFOVs to compensate for reverse band offsets, odd detector layout geometry and multiplexer sampling for Bands 1 through 8 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.NNNNNNNNNESNN, where S = "+" or "-", N = 0 to 9, and E = "E" |
| GROUP: ATTITUDE_PARAMETERS | ADSA_To_ETM_ Matrix | Static | float32 array (9 values) | Matrix describing the relationship of the Attitude Displacement Sensor Assembly (ADSA) to the ETM+ optical axis Valid format: SN.NNNNNNNNNESNN, where S = "+" or "-", N = 0 to 9, and E = "E" |
| GROUP: ATTITUDE_PARAMETERS | Attitude_To_ETM_ Matrix | Static | float32 array (9 values) | Matrix describing the relationship of the attitude control reference axis to the ETM+ optical axis Valid format: SN.NNNNNNNNNESNN, 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.NNNNNNNNNESNN, where N.NNNNNNNNNESNN = 0.00000000E+00 |
| GROUP: ATTITUDE_PARAMETERS | Spacecraft_Pitch_ Bias | Static | float32 | Spacecraft pitch bias in radians Valid format: N.NNNNNNNNNESNN, where N.NNNNNNNNNESNN = 0.00000000E+00 |
| GROUP: ATTITUDE_PARAMETERS | Spacecraft_Yaw_Bias | Static | float32 | Spacecraft yaw bias in radians Valid format: N.NNNNNNNNNESNN, where N.NNNNNNNNNESNN = 0.00000000E+00 |
| GROUP: ATTITUDE_PARAMETERS | IMU_Drift_Bias_XA | Static | float32 | Inertial Measurement Unit (IMU) XA axis drift bias in radians per second. Valid format: SN.NNNNNNNNNESNN, where SN.NNNNNNNNNESNN = -2.23500000E-06 |
| GROUP: ATTITUDE_PARAMETERS | IMU_Drift_Bias_YA | Static | float32 | IMU YA axis drift bias in radians per second. Valid format: SN.NNNNNNNNNESNN, where SN.NNNNNNNNNESNN = -2.23500000E-06 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: ATTITUDE_PARAMETERS | IMU_Drift_Bias_ZA | Static | float32 | IMU ZA axis drift bias in radians per second. Valid format: N.NNNNNNNNESNN, where N.NNNNNNNNESNN = 1.68230000E-06 |
| GROUP: ATTITUDE_PARAMETERS | IMU_Drift_Bias_XB | Static | float32 | IMU XB axis drift bias in radians per second. Valid format: N.NNNNNNNNESNN, where N.NNNNNNNNESNN = 1..86665000E-06 |
| GROUP: ATTITUDE_PARAMETERS | IMU_Drift_Bias_YB | Static | float32 | IMU YB axis drift bias in radians per second. Valid format: SN.NNNNNNNNESNN, where SN.NNNNNNNNESNN = -6.35100000E-07 |
| GROUP: ATTITUDE_PARAMETERS | IMU_Drift_Bias_ZB | Static | float32 | IMU ZB axis drift bias in radians per second. Valid format: N.NNNNNNNNESNN, where N.NNNNNNNNESNN = 4.84810000E-08 |
| 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.NNNNNNN, where N.NNNNNNN = 3.3113091 |
| GROUP: TRANSFER_FUNCTION GROUP: IMU | Zeta | Static | float64 | Inertial measurement unit transfer function damping coefficient Valid format: N.NNNNNNNN, where N.NNNNNNNN = 0.66882924 |
| GROUP: TRANSFER_FUNCTION GROUP: IMU | Tau | Static | float64 | Inertial measurement unit transfer function denominator time constant (seconds) Valid format: SN.NNNNNNNESN, where SN.NNNNNNNESN = -1.6086176E-2 |
| GROUP: TRANSFER_FUNCTION GROUP: IMU | P | Static | float64 | Inertial measurement unit transfer function numerator time constant (seconds) Valid format: SN.NNNNNNNESN, where SN.NNNNNNNESN = -4.1138195E-3 |
| GROUP: TRANSFER_FUNCTION GROUP: IMU | Ak | Static | float64 | Inertial measurement unit transfer function DC gain Valid format: N.NNNNNNN, where N.NNNNNNN = 1.0103061 |
| 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 the three ADS units; determined at 15 degrees C Valid format: SN.NNNNNNNEN, where S = "+" or "-", N = 0 to 9, and E = "E" |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| 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<br>Valid format: SN.NNNNNNNEN, where S = "+" or "-", 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<br>Valid format: SN.NNNNNNNESN, where S = "+" or "-", 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<br>Valid format: SN.NNNNNNNESN, where S = "+" or "-", 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.)<br>Valid format: N.NNNNNNNNN, 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<br>Valid format: SN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: TRANSFER_FUNCTION GROUP: PREFILTER | ADSPre_T | Static | float64 array (5 values) | ADS prefilter transfer function linear term time constants<br>Valid format: N.NNNNNNN, 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<br>Valid format: YYYY, where YYYY = 1998-2020 |
| GROUP: UT1_TIME_PARAMETERS | UT1_Month | Dynamic | char8 array (180 values) | Month of UT1 time correction prediction; values span 180 days<br>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<br>Valid format: NN, where NN = 1-31 |
| 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.<br>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<br>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<br>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<br>Valid format: N.NNNNN, where N.NNNNN = e.g., 0.44321 |

IAS-207
Version 6.0

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: DETECTOR_STATUS | Status_Band1 | Dynamic | char8 array (16 values) | Health status of Band 1's 16 detectors<br>Valid format: ABCDE, where<br>A = 0 (live), 1 (dead), 2 (intermittent)<br>B = 0 (noise in spec, low gain), 1 (noisy low signal), 2 (noisy high signal), 3 (noisy both signals), 4 (inoperable)<br>C = 0 (noise in spec, high gain), 1 (noisy low signal), 2 (noisy high signal), 3 (noisy both signals), 4 (inoperable)<br>D = 0 (dynamic range in spec, low gain) 1 (fail, high end), 2 (fail, low end), 3 (fail, both ends), 4 (inoperable)<br>E = 0 (dynamic range in spec, high gain), 1 (fail, low end), 2 (fail, low end), 3 (fail, both ends), 4 (inoperable) |
| GROUP: DETECTOR_STATUS | Status_Band2 | Dynamic | char8 array (16 values) | Health status of Band 2's 16 detectors<br>Valid format: as above |
| GROUP: DETECTOR_STATUS | Status_Band3 | Dynamic | char8 array (16 values) | Health status of Band 3's 16 detectors<br>Valid format: as above. |
| GROUP: DETECTOR_STATUS | Status_Band4 | Dynamic | char8 array (16 values) | Health status of Band 4's 16 detectors<br>Valid format: as above |
| GROUP: DETECTOR_STATUS | Status_Band5 | Dynamic | char8 array (16 values) | Health status of Band 5's 16 detectors<br>Valid format: as above |
| GROUP: DETECTOR_STATUS | Status_Band6 | Dynamic | char8 array (8 values) | Health status of Band 6's 8 detectors<br>Valid format: as above |
| GROUP: DETECTOR_STATUS | Status_Band7 | Dynamic | char8 array (16 values) | Health status of Band 7's 16 detectors<br>Valid format: as above |
| GROUP: DETECTOR_STATUS | Status_Band8 | Dynamic | char8 array (32 values) | Health status of Band 8's 32 detectors<br>Valid format: as above |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B1L_Prelaunch | Static | float32 array (16 values) | Band 1 prelaunch low gain in counts/W/m^2-ster-μm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B1L_Postlaunch | Static | float32 array (16 values) | Band 1 postlaunch low gain in counts/W/m^2-ster-μm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B1L_Current | Dynamic | float32 array (16 values) | Band 1 current low gain in counts/W/m^2-ster-μm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B2L_Prelaunch | Static | float32 array (16 values) | Band 2 prelaunch low gain in counts/W/m^2-ster-μm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B2L_Postlaunch | Static | float32 array (16 values) | Band 2 postlaunch low gain in counts/W/m^2-ster-μm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B2L_Current | Dynamic | float32 array (16 values) | Band 2 current low gain in counts/W/m^2-ster-μm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B3L_Prelaunch | Static | float32 array (16 values) | Band 3 prelaunch low gain in counts/W/m^2-ster-μm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B3L_Postlaunch | Static | float32 array (16 values) | Band 3 postlaunch low gain in counts/W/m^2-ster-μm<br>Valid format: NN.NNNNN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B3L_Current | Dynamic | float32 array (16 values) | Band 3 current low gain in counts/W/m^2-ster-µm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B4L_Prelaunch | Static | float32 array (16 values) | Band 4 prelaunch low gain in counts/W/m^2-ster-µm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B4L_Postlaunch | Static | float32 array (16 values) | Band 4 postlaunch low gain in counts/W/m^2-ster-µm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B4L_Current | Dynamic | float32 array (16 values) | Band 4 current low gain in counts/W/m^2-ster-µm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B5L_Prelaunch | Static | float32 array (16 values) | Band 5 prelaunch low gain in counts/W/m^2-ster-µm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B5L_Postlaunch | Static | float32 array (16 values) | Band 5 postlaunch low gain in counts/W/m^2-ster-µm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B5L_Current | Dynamic | float32 array (16 values) | Band 5 current low gain in counts/W/m^2-ster-µm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B6L_Prelaunch | Static | float32 array (8 values) | Band 6 prelaunch low gain in counts/W/m^2-ster-µm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B6L_Postlaunch | Static | float32 array (8 values) | Band 6 postlaunch low gain in counts/W/m^2-ster-µm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B6L_Current | Dynamic | float32 array (8 values) | Band 6 current low gain in counts/W/m^2-ster-µm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B7L_Prelaunch | Static | float32 array (16 values) | Band 7 prelaunch low gain in counts/W/m^2-ster-µm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B7L_Postlaunch | Static | float32 array (16 values) | Band 7 postlaunch low gain in counts/W/m^2-ster-µm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B7L_Current | Dynamic | float32 array (16 values) | Band 7 current low gain in counts/W/m^2-ster-µm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B8L_Prelaunch | Static | float32 array (32 values) | Band 8 prelaunch low gain in counts/W/m^2-ster-µm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B8L_Postlaunch | Static | float32 array (32 values) | Band 8 postlaunch low gain in counts/W/m^2-ster-µm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_LOW | B8L_Current | Dynamic | float32 array (32 values) | Band 8 current low gain in counts/W/m^2-ster-µm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B1H_Prelaunch | Static | float32 array (16 values) | Band 1 prelaunch high gain in counts/W/m^2-ster-µm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B1H_Postlaunch | Static | float32 array (16 values) | Band 1 postlaunch high gain in counts/W/m^2-ster-µm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B1H_Current | Dynamic | float32 array (16 values) | Band 1 current high gain in counts/W/m^2-ster-µm<br>Valid format: NN.NNNNN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B2H_Prelaunch | Static | float32 array (16 values) | Band 2 prelaunch high gain in counts/W/m^2-ster-μm Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B2H_Postlaunch | Static | float32 array (16 values) | Band 2 postlaunch high gain in counts/W/m^2-ster-μm Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B2H_Current | Dynamic | float32 array (16 values) | Band 2 current high gain in counts/W/m^2-ster-μm Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B3H_Prelaunch | Static | float32 array (16 values) | Band 3 prelaunch high gain in counts/W/m^2-ster-μm Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B3H_Postlaunch | Static | float32 array (16 values) | Band 3 postlaunch high gain in counts/W/m^2-ster-μm Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B3H_Current | Dynamic | float32 array (16 values) | Band 3 current high gain in counts/W/m^2-ster-μm Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B4H_Prelaunch | Static | float32 array (16 values) | Band 4 prelaunch high gain in counts/W/m^2-ster-μm Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B4H_Postlaunch | Static | float32 array (16 values) | Band 4 postlaunch high gain in counts/W/m^2-ster-μm Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B4H_Current | Dynamic | float32 array (16 values) | Band 4 current high gain in counts/W/m^2-ster-μm Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B5H_Prelaunch | Static | float32 array (16 values) | Band 5 prelaunch high gain in counts/W/m^2-ster-μm Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B5H_Postlaunch | Static | float32 array (16 values) | Band 5 postlaunch high gain in counts/W/m^2-ster-μm Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B5H_Current | Dynamic | float32 array (16 values) | Band 5 current high gain in counts/W/m^2-ster-μm Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B6H_Prelaunch | Static | float32 array (8 values) | Band 6 prelaunch high gain in counts/W/m^2-ster-μm Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B6H_Postlaunch | Static | float32 array (8 values) | Band 6 postlaunch high gain in counts/W/m^2-ster-μm Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B6H_Current | Dynamic | float32 array (8 values) | Band 6 current high gain in counts/W/m^2-ster-μm Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B7H_Prelaunch | Static | float32 array (16 values) | Band 7 prelaunch high gain in counts/W/m^2-ster-μm Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B7H_Postlaunch | Static | float32 array (16 values) | Band 7 postlaunch high gain in counts/W/m^2-ster-μm Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B7H_Current | Dynamic | float32 array (16 values) | Band 7 current high gain in counts/W/m^2-ster-μm Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B8H_Prelaunch | Static | float32 array (32 values) | Band 8 prelaunch high gain in counts/W/m^2-ster-μm Valid format: NN.NNNNN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B8H_Postlaunch | Static | float32 array (32 values) | Band 8 postlaunch high gain in counts/W/m^2-ster-μm<br>Valid format: NN.NNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_GAINS GROUP: DETECTOR_GAINS_HIGH | B8H_Current | Dynamic | float32 array (32 values) | Band 8 current high gain in counts/W/m^2-ster-μm<br>Valid format: NN.NNNNN, 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<br>Valid format: NNN, where NNN = 143 |
| 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<br>Valid format: NNN, where NNN = 500 |
| GROUP: BIAS_LOCATIONS | Forward_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<br>Valid format: NNN, where NNN = 814 |
| 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<br>Valid format: NNN, where NNN = 810 |
| GROUP: BIAS_LOCATIONS | Reverse_Bias_Length_30 | Dynamic | int16 | Number of pixels to use per line, in calculating bias for Bands 1-5 and 7<br>Valid format: NNN, where NNN = 500 |
| 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<br>Valid format: NNN, where NNN = 810 |
| GROUP: BIAS_LOCATIONS | Forward_Bias_Location_60 | 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<br>Valid format: NNN, where NNN = 85 |
| GROUP: BIAS_LOCATIONS | Forward_Bias_Length_60 | Dynamic | int16 | Number of pixels to use, per line, in calculating bias for Band 6<br>Valid format: NNN, where NNN = 275 |
| GROUP: BIAS_LOCATIONS | Forward_IC_Region_60 | 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<br>Valid format: NNN, where NNN = 380 |
| GROUP: BIAS_LOCATIONS | Reverse_Bias_Location_60 | 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<br>Valid format: NNN, where NNN = 400 |
| GROUP: BIAS_LOCATIONS | Reverse_Bias_Length_60 | Dynamic | int16 | Number of pixels to use, per line, in calculating bias for Band 6<br>Valid format: NNN, where NNN = 275 |
| GROUP: BIAS_LOCATIONS | Reverse_IC_Region_60 | 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<br>Valid format: NNN, where NNN = 410 |
| GROUP: BIAS_LOCATIONS | Forward_Bias_Location_15 | 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 8<br>Valid format: NNN, where NNN = 286 |
| GROUP: BIAS_LOCATIONS | Forward_Bias_Length_15 | Dynamic | int16 | Number of pixels to use, per line, in calculating bias for Band 8<br>Valid format: NNNN, where NNNN = 1000 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: BIAS_LOCATIONS | Forward_IC_ Region_15 | 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 8<br>Valid format: NNNN, where NNNN = 1635 |
| GROUP: BIAS_LOCATIONS | Reverse_Bias_ Location_15 | 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 8<br>Valid format: NNNN, where NNNN = 1610 |
| GROUP: BIAS_LOCATIONS | Reverse_Bias_ Length_15 | Dynamic | int16 | Number of pixels to use, per line, in calculating bias for Band 8<br>Valid format: NNNN, where NNNN = 1000 |
| GROUP: BIAS_LOCATIONS | Reverse_IC_ Region_15 | 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 8<br>Valid format: NNNN, where NNNN = 1646 |
| GROUP: DETECTOR_BIASES_B6 GROUP: DETECTOR_ BIASES_B6_LOW | B6L_Bias_Prelaunch | Static | float32 array (8 values) | Band 6 prelaunch low gain bias in digital counts<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: DETECTOR_BIASES_B6 GROUP: DETECTOR_ BIASES_B6_LOW | B6L_Bias_Postlaunch | Static | float32 array (8 values) | Band 6 postlaunch low gain bias in digital counts<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: DETECTOR_BIASES_B6 GROUP: DETECTOR_ BIASES_B6_LOW | B6L_Bias_Current | Dynamic | float32 array (8 values) | Band 6 current low gain bias in digital counts<br>Valid format: NN.NNN, where N = 0 to 9 |
| GROUP: DETECTOR_BIASES_B6 GROUP: DETECTOR_ BIASES_B6_HIGH | B6H_Bias_Prelaunch | Static | float32 array (8 values) | Band 6 prelaunch high gain bias in digital counts<br>Valid format: SNN.NN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: DETECTOR_BIASES_B6 GROUP: DETECTOR_ BIASES_B6_HIGH | B6H_Bias_Postlaunch | Static | float32 array (8 values) | Band 6 postlaunch high gain bias in digital counts<br>Valid format: SNN.NN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: DETECTOR_BIASES_B6 GROUP: DETECTOR_ BIASES_B6_HIGH | B6H_Bias_Current | Dynamic | float32 array (8 values) | Band 6 current high gain bias in digital counts<br>Valid format: SNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: ACCA_BIASES GROUP: ACCA_BIASES_LOW | B1L_ACCA_Bias | Dynamic | float32 array (16 values) | Band 1 low-gain Automated Cloud Cover Assessment (ACCA) bias in digital counts for detectors 1-16<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: ACCA_BIASES GROUP: ACCA_BIASES_LOW | B2L_ACCA_Bias | Dynamic | float32 array (16 values) | Band 2 low-gain ACCA bias in digital counts for detectors 1-16<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: ACCA_BIASES GROUP: ACCA_BIASES_LOW | B3L_ACCA_Bias | Dynamic | float32 array (16 values) | Band 3 low-gain ACCA bias in digital counts for detectors 1-16<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: ACCA_BIASES GROUP: ACCA_BIASES_LOW | B4L_ACCA_Bias | Dynamic | float32 array (16 values) | Band 4 low-gain ACCA bias in digital counts for detectors 1-16<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: ACCA_BIASES GROUP: ACCA_BIASES_LOW | B5L_ACCA_Bias | Dynamic | float32 array (16 values) | Band 5 low-gain ACCA bias in digital counts for detectors 1-16<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: ACCA_BIASES GROUP: ACCA_BIASES_LOW | B6L_ACCA_Bias | Dynamic | float32 array (8 values) | Band 6 low-gain ACCA bias in digital counts for detectors 1-8<br>Valid format: NN.NNN, where N = 0 to 9 |

IAS-207
Version 6.0

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: ACCA_BIASES GROUP: ACCA_BIASES_LOW | B7L_ACCA_Bias | Dynamic | float32 array (16 values) | Band 7 low-gain ACCA bias in digital counts for detectors 1-16 Valid format: NN.NN, where N = 0 to 9 |
| GROUP: ACCA_BIASES GROUP: ACCA_BIASES_LOW | B8L_ACCA_Bias | Dynamic | float32 array (32 values) | Band 8 low-gain ACCA bias in digital counts for detectors 1-32 Valid format: NN.NN, where N = 0 to 9 |
| GROUP: ACCA_BIASES GROUP: ACCA_BIASES_HIGH | B1H_ACCA_Bias | Dynamic | float32 array (16 values) | Band 1 high-gain ACCA bias in digital counts for detectors 1-16 Valid format: NN.NN, where N = 0 to 9 |
| GROUP: ACCA_BIASES GROUP: ACCA_BIASES_HIGH | B2H_ACCA_Bias | Dynamic | float32 array (16 values) | Band 2 high-gain ACCA bias in digital counts for detectors 1-16 Valid format: NN.NN, where N = 0 to 9 |
| GROUP: ACCA_BIASES GROUP: ACCA_BIASES_HIGH | B3H_ACCA_Bias | Dynamic | float32 array (16 values) | Band 3 high-gain ACCA bias in digital counts for detectors 1-16 Valid format: NN.NN, where N = 0 to 9 |
| GROUP: ACCA_BIASES GROUP: ACCA_BIASES_HIGH | B4H_ACCA_Bias | Dynamic | float32 array (16 values) | Band 4 high-gain ACCA bias in digital counts for detectors 1-16 Valid format: NN.NN, where N = 0 to 9 |
| GROUP: ACCA_BIASES GROUP: ACCA_BIASES_HIGH | B5H_ACCA_Bias | Dynamic | float32 array (16 values) | Band 5 high-gain ACCA bias in digital counts for detectors 1-16 Valid format: NN.NN, where N = 0 to 9 |
| GROUP: ACCA_BIASES GROUP: ACCA_BIASES_HIGH | B6H_ACCA_Bias | Dynamic | float32 array (8 values) | Band 6 high-gain ACCA bias in digital counts for detectors 1- 8 Valid format: SNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: ACCA_BIASES GROUP: ACCA_BIASES_HIGH | B7H_ACCA_Bias | Dynamic | float32 array (16 values) | Band 7 high-gain ACCA bias in digital counts for detectors 1-16 Valid format: NN.NN, where N = 0 to 9 |
| GROUP: ACCA_BIASES GROUP: ACCA_BIASES_HIGH | B8H_ACCA_Bias | Dynamic | float32 array (32 values) | Band 8 high-gain ACCA bias in digital counts for detectors 1-32 Valid format: NN.NN, where N = 0 to 9 |
| GROUP: ACCA_THRESHOLDS | Thresh_B3 | Dynamic | float32 | Band 3 ACCA threshold Valid format: N.NNNN, where N.NNNN = 0.0800 |
| GROUP: ACCA_THRESHOLDS | Thresh_B3_Lower | Dynamic | float32 | Band 3 land reflectance threshold Valid format: NN.NN, where NN.NN = 0.07 |
| GROUP: ACCA_THRESHOLDS | Thresh_B56_High | Dynamic | float32 | Band 5-6 high-composite threshold Valid format: NNN.NNN, where NNN.NNN = 225.000 |
| GROUP: ACCA_THRESHOLDS | Thresh_B56_Low | Dynamic | float32 | Band 5-6 low-composite threshold Valid format: NNN.NNN, where NNN.NNN = 210.000 |
| GROUP: ACCA_THRESHOLDS | Thresh_B6 | Dynamic | float32 | Band 6 threshold - maximum cloud temperature Valid format: NNN.NNN, where NNN.NNN = 300.000 |
| GROUP: ACCA_THRESHOLDS | Thresh_B45_Ratio | Dynamic | float32 | Band 4-5 ratio threshold Valid format: N.NNNN, where N.NNNN = 1.0000 |
| GROUP: ACCA_THRESHOLDS | Thresh_B42_Ratio | Dynamic | float32 | Band 4-2 ratio threshold Valid format: N.NNNNN, where N.NNNNN = 2.16248 |
| GROUP: ACCA_THRESHOLDS | Thresh_B43_Ratio | Dynamic | float32 | Band 4-3 ratio threshold Valid format: N.NNNN, where N.NNNN = 2.3500 |
| GROUP: ACCA_THRESHOLDS | Thresh_NDSI_Max | Dynamic | float32 | Normalized Snow Difference Index (NDSI) ceiling Valid format: N.NNNN, where N.NNNN = 0.7000 |
| GROUP: ACCA_THRESHOLDS | Thresh_NDSI_Min | Dynamic | float32 | Normalized snow difference index floor Valid format: SN.NNNN, where SN.NNNN = -0.2500 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: ACCA_THRESHOLDS | Thresh_NDSI_Snow | Dynamic | float32 | NDSI threshold used to identify snow<br>Valid format: NN.NNNN, where<br>NN.NNNN = 0.8000 |
| GROUP: ACCA_THRESHOLDS | Cloud_Percent_Min | Dynamic | float32 | Minimum cloud cover percentage required for pass two<br>Valid format: N.NNNN, where N.NNNN = 0.4000 |
| GROUP: ACCA_THRESHOLDS | Desert_Index | Dynamic | float32 | Desert index (Thresh_45_Ratio/ Thresh_42_Ratio)<br>Valid format: N.NNN, where N.NNN = 0.500 |
| GROUP: ACCA_THRESHOLDS | Thresh_Snow_Percent | Dynamic | float32 | Maximum snow cover percentage allowed to use looser cloud properties for pass two<br>Valid format: N.NNNN, where N.NNNN = 1.0000 |
| GROUP: ACCA_THRESHOLDS | Thermal_Effect_High | Dynamic | float32 | Maximum allowable pass two percentage cloud cover increase allowed using looser cloud properties<br>Valid format: NN.NNNN, where<br>NN.NNNN = 35.0000 |
| GROUP: ACCA_THRESHOLDS | Thermal_Effect_Low | Dynamic | float32 | Maximum allowable pass two percentage cloud cover increase allowed using narrower cloud properties<br>Valid format: NN.NNN, where NN.NNN = 25.000 |
| GROUP: ACCA_THRESHOLDS | B6Max_Maxthresh_Diff | Dynamic | float32 | Minimum difference allowed between maximum cloud temperature and maximum thermal threshold<br>Valid format: NN.NNN, where NN.NNN = 2.000 |
| GROUP: SOLAR_SPECTRAL_ IRRADIANCES | B1_Solar_Irradiance | Static | float32 | Mean solar exoatmospheric irradiance for Band 1 in W/m^2-ster-µm<br>Valid format: NNNN.NNN, where<br>NNNN.NNN = 1969.000 |
| GROUP: SOLAR_SPECTRAL_ IRRADIANCES | B2_Solar_Irradiance | Static | float32 | Mean solar exoatmospheric irradiance for Band 2 in W/m^2-ster-µm<br>Valid format: NNNN.NNN, where<br>NNNN.NNN = 1840.000 |
| GROUP: SOLAR_SPECTRAL_ IRRADIANCES | B3_Solar_Irradiance | Static | float32 | Mean solar exoatmospheric irradiance for Band 3 in W/m^2-ster-µm<br>Valid format: NNNN.NNN, where<br>NNNN.NNN = 1551.000 |
| GROUP: SOLAR_SPECTRAL_ IRRADIANCES | B4_Solar_Irradiance | Static | float32 | Mean solar exoatmospheric irradiance for Band 4 in W/m^2-ster-µm<br>Valid format: NNNN.NNN, where<br>NNNN.NNN = 1044.000 |
| GROUP: SOLAR_SPECTRAL_ IRRADIANCES | B5_Solar_Irradiance | Static | float32 | Mean solar exoatmospheric irradiance for Band 5 in W/m^2-ster-µm<br>Valid format: NNNN.NNN, where<br>NNNN.NNN = 225.700 |
| GROUP: SOLAR_SPECTRAL_ IRRADIANCES | B7_Solar_Irradiance | Static | float32 | Mean solar exoatmospheric irradiance for Band 7 in W/m^2-ster-µm<br>Valid format: NNNN.NNN, where<br>NNNN.NNN = 82.070 |
| GROUP: SOLAR_SPECTRAL_ IRRADIANCES | B8_Solar_Irradiance | Static | float32 | Mean solar exoatmospheric irradiance for Band 8 in W/m^2-ster-µm<br>Valid format: NNNN.NNN, where<br>NNNN.NNN = 1368.000 |
| GROUP: THERMAL_CONSTANTS | K1_Constant | Static | float32 | Thermal calibration constant 1 in W/m^2-ster-µm<br>Valid format: NNN.NN, where NNN.NN = 666.09 |
| GROUP: THERMAL_CONSTANTS | K2_Constant | Static | float32 | Thermal calibration constant 2 kelvin<br>Valid format: NNNN.NN, where<br>NNNNN.NN = 1282.71 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: SCALING_PARAMETERS GROUP: SCALING_ PARAMETERS_LOW | B1L_Lmin_Lmax | Static | float32 array (2 values) | Postcalibration 8-bit dynamic range scaling factors for Band 1, low gain, W/m^2-ster-µm Valid format: SNNN.NN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCALING_PARAMETERS GROUP: SCALING_ PARAMETERS_LOW | B2L_Lmin_Lmax | Static | float32 array (2 values) | Postcalibration 8-bit dynamic range scaling factors for Band 2, low gain, W/m^2-ster-µm Valid format: SNNN.NN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCALING_PARAMETERS GROUP: SCALING_ PARAMETERS_LOW | B3L_Lmin_Lmax | Static | float32 array (2 values) | Postcalibration 8-bit dynamic range scaling factors for Band 3, low gain, W/m^2-ster-µm Valid format: SNNN.NN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCALING_PARAMETERS GROUP: SCALING_ PARAMETERS_LOW | B4L_Lmin_Lmax | Static | float32 array (2 values) | Postcalibration 8-bit dynamic range scaling factors for Band 4, low gain, W/m^2-ster-µm Valid format: SNNN.NN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCALING_PARAMETERS GROUP: SCALING_ PARAMETERS_LOW | B5L_Lmin_Lmax | Static | float32 array (2 values) | Postcalibration 8-bit dynamic range scaling factors for Band 5, low gain, W/m^2-ster-µm Valid format: SNNN.NN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCALING_PARAMETERS GROUP: SCALING_ PARAMETERS_LOW | B6L_Lmin_Lmax | Static | float32 array (2 values) | Postcalibration 8-bit dynamic range scaling factors for Band 6, low gain, W/m^2-ster-µm Valid format: SNNN.NN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCALING_PARAMETERS GROUP: SCALING_ PARAMETERS_LOW | B7L_Lmin_Lmax | Static | float32 array (2 values) | Postcalibration 8-bit dynamic range scaling factors for Band 7, low gain, W/m^2-ster-µm Valid format: SNNN.NN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCALING_PARAMETERS GROUP: SCALING_ PARAMETERS_LOW | B8L_Lmin_Lmax | Static | float32 array (2 values) | Postcalibration 8-bit dynamic range scaling factors for Band 8, low gain, W/m^2-ster-µm Valid format: SNNN.NN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCALING_PARAMETERS GROUP: SCALING_ PARAMETERS_LOW | B1H_Lmin_Lmax | Static | float32 array (2 values) | Postcalibration 8-bit dynamic range scaling factors for Band 1, high gain, W/m^2-ster-µm Valid format: SNNN.NN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCALING_PARAMETERS GROUP: SCALING_ PARAMETERS_HIGH | B2H_Lmin_Lmax | Static | float32 array (2 values) | Postcalibration 8-bit dynamic range scaling factors for Band 2, high gain, W/m^2-ster-µm Valid format: SNNN.NN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCALING_PARAMETERS GROUP: SCALING_ PARAMETERS_HIGH | B3H_Lmin_Lmax | Static | float32 array (2 values) | Postcalibration 8-bit dynamic range scaling factors for Band 3, high gain, W/m^2-ster-µm Valid format: SNNN.NN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCALING_PARAMETERS GROUP: SCALING_ PARAMETERS_HIGH | B4H_Lmin_Lmax | Static | float32 array (2 values) | Postcalibration 8-bit dynamic range scaling factors for Band 4, high gain, W/m^2-ster-µm Valid format: SNNN.NN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCALING_PARAMETERS GROUP: SCALING_ PARAMETERS_HIGH | B5H_Lmin_Lmax | Static | float32 array (2 values) | Postcalibration 8-bit dynamic range scaling factors for Band 5, high gain, W/m^2-ster-µm Valid format: SNNN.NN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCALING_PARAMETERS GROUP: SCALING_ PARAMETERS_HIGH | B6H_Lmin_Lmax | Static | float32 array (2 values) | Postcalibration 8-bit dynamic range scaling factors for Band 6, high gain, W/m^2-ster-µm Valid format: SNNN.NN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCALING_PARAMETERS GROUP: SCALING_ PARAMETERS_HIGH | B7H_Lmin_Lmax | Static | float32 array (2 values) | Postcalibration 8-bit dynamic range scaling factors for Band 7, high gain, W/m^2-ster-µm Valid format: SNNN.NN, where S = "+" or "-" and N = 0 to 9 |

IAS-207
Version 6.0

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: SCALING_PARAMETERS GROUP: SCALING_ PARAMETERS_HIGH | B8H_Lmin_Lmax | Static | float32 array (2 values) | Postcalibration 8-bit dynamic range scaling factors for Band 8, high gain, W/m^2-ster-µm Valid format: SNNN.NN, 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: SN.NNNNNNNN, 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: SN.NNNNNNNN, 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: SN.NNNNNNNN, 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: SN.NNNNNNNN, 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: SN.NNNNNNNN, 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: SN.NNNNNNNN, 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: SN.NNNNNNNN, 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: SN.NNNNNNNN, 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: SN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| 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: SN.NNNNNNNN, 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: SN.NNNNNNNN, 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: SN.NNNNNNNN, 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: SN.NNNNNNNN, 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: SN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: MTF_COMPENSATION | B8_weights_along | Dynamic | float64 array (5 values) | Weighting function coefficients used to compute along-scan MTFC for Band 8 Valid format: SN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: MTF_COMPENSATION | B8_weights_across | Dynamic | float64 array (5 values) | Weighting function coefficients used to compute across-scan MTFC for Band 8 Valid format: SN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: MEMORY_EFFECT GROUP: ME_MAGNITUDES | B1_ME_Magnitude | Dynamic | float32 array (16 values) | Band 1 memory effect magnitude measured in Digital Numbers (DNs) Valid format: NNN.NNNNNNNN, where N = 0 to 9 |
| GROUP: MEMORY_EFFECT GROUP: ME_MAGNITUDES | B2_ME_Magnitude | Dynamic | float32 array (16 values) | Band 2 memory effect magnitude measured in DNs Valid format: NNN.NNNNNNNN, where N = 0 to 9 |
| GROUP: MEMORY_EFFECT GROUP: ME_MAGNITUDES | B3_ME_Magnitude | Dynamic | float32 array (16 values) | Band 3 memory effect magnitude measured in DNs Valid format: NNN.NNNNNNNN, where N = 0 to 9 |
| GROUP: MEMORY_EFFECT GROUP: ME_MAGNITUDES | B4_ME_Magnitude | Dynamic | float32 array (16 values) | Band 3 memory effect magnitude measured in DNs Valid format: NNN.NNNNNNNN, where N = 0 to 9 |
| GROUP: MEMORY_EFFECT GROUP: ME_MAGNITUDES | B5_ME_Magnitude | Dynamic | float32 array (16 values) | Band 3 memory effect magnitude measured in DNs Valid format: NNN.NNNNNNNN, where N = 0 to 9 |
| GROUP: MEMORY_EFFECT GROUP: ME_MAGNITUDES | B6_ME_Magnitude | Dynamic | float32 array (8 values) | Band 3 memory effect magnitude measured in DNs Valid format: NNN.NNNNNNNN, where N = 0 to 9 |
| GROUP: MEMORY_EFFECT GROUP: ME_MAGNITUDES | B7_ME_Magnitude | Dynamic | float32 array (16 values) | Band 3 memory effect magnitude measured in DNs Valid format: NNN.NNNNNNNN, where N = 0 to 9 |
| GROUP: MEMORY_EFFECT GROUP: ME_MAGNITUDES | B8_ME_Magnitude | Dynamic | float32 array (32 values) | Band 3 memory effect magnitude measured in DNs Valid format: NNN.NNNNNNNN, 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 Valid format: NNNN.NNNNNNNN, where N = 0 to 9 |
| GROUP: MEMORY_EFFECT GROUP: ME_TIME_CONSTANTS | B2_ME_Time_Constant | Dynamic | float32 array (16 values) | Band 2 time constant measured in minor frames Valid format: NNNN.NNNNNNNN, 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 Valid format: NNNN.NNNNNNNN, 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 Valid format: NNNN.NNNNNNNN, 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 Valid format: NNNN.NNNNNNNN, where N = 0 to 9 |
| GROUP: MEMORY_EFFECT GROUP: ME_TIME_CONSTANTS | B6_ME_Time_Constant | Dynamic | float32 array (8 values) | Band 6 time constant measured in minor frames Valid format: NNNN.NNNNNNNN, 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 Valid format: NNNN.NNNNNNNN, where N = 0 to 9 |
| GROUP: MEMORY_EFFECT GROUP: ME_TIME_CONSTANTS | B8_ME_Time_Constant | Dynamic | float32 array (32 values) | Band 8 time constant measured in minor frames Valid format: NNNN.NNNNNNNN, 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 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: SCAN_CORRELATED_SHIFT | SCS_Reference_ Detectors | Dynamic | uint8 array (7 values) | Scan correlated shift reference detector, one per band<br>Valid format: NN, where NN = 1-16 |
| GROUP: SCAN_CORRELATED_SHIFT GROUP: SCS_LOW | B1L_SCS_Magnitudes | Dynamic | float32 array (16 values) | Magnitude of Band 1 low-gain shift in digital numbers<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCAN_CORRELATED_SHIFT GROUP: SCS_LOW | B2L_SCS_Magnitudes | Dynamic | float32 array (16 values) | Magnitude of Band 2 low-gain shift in digital numbers<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCAN_CORRELATED_SHIFT GROUP: SCS_LOW | B3L_SCS_Magnitudes | Dynamic | float32 array (16 values) | Magnitude of Band 3 low-gain shift in digital numbers<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCAN_CORRELATED_SHIFT GROUP: SCS_LOW | B4L_SCS_Magnitudes | Dynamic | float32 array (16 values) | Magnitude of Band 4 low-gain shift in digital numbers<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCAN_CORRELATED_SHIFT GROUP: SCS_LOW | B5L_SCS_Magnitudes | Dynamic | float32 array (16 values) | Magnitude of Band 5 low-gain shift in digital numbers<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCAN_CORRELATED_SHIFT GROUP: SCS_LOW | B7L_SCS_Magnitudes | Dynamic | float32 array (16 values) | Magnitude of Band 7 low-gain shift in digital numbers<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCAN_CORRELATED_SHIFT GROUP: SCS_LOW | B8L_SCS_Magnitudes | Dynamic | float32 array (32 values) | Magnitude of Band 8 low-gain shift in digital numbers<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCAN_CORRELATED_SHIFT GROUP: SCS_HIGH | B1H_SCS_Magnitudes | Dynamic | float32 array (16 values) | Magnitude of Band 1 high-gain shift in digital numbers<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCAN_CORRELATED_SHIFT GROUP: SCS_HIGH | B2H_SCS_Magnitudes | Dynamic | float32 array (16 values) | Magnitude of Band 2 high-gain shift in digital numbers<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCAN_CORRELATED_SHIFT GROUP: SCS_HIGH | B3H_SCS_Magnitudes | Dynamic | float32 array (16 values) | Magnitude of Band 3 high-gain shift in digital numbers<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCAN_CORRELATED_SHIFT GROUP: SCS_HIGH | B4H_SCS_Magnitudes | Dynamic | float32 array (16 values) | Magnitude of Band 4 high-gain shift in digital numbers<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCAN_CORRELATED_SHIFT GROUP: SCS_HIGH | B5H_SCS_Magnitudes | Dynamic | float32 array (16 values) | Magnitude of Band 5 high-gain shift in digital numbers<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCAN_CORRELATED_SHIFT GROUP: SCS_HIGH | B7H_SCS_Magnitudes | Dynamic | float32 array (16 values) | Magnitude of Band 7 high-gain shift in digital numbers<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SCAN_CORRELATED_SHIFT GROUP: SCS_HIGH | B8H_SCS_Magnitudes | Dynamic | float32 array (32 values) | Magnitude of Band 8 high-gain shift in digital numbers<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: STRIPING GROUP: STRIPING_FLAG_LOW | Correction_ Reference_B1_Low | Static | uint8 | Striping correction methodology flag, relative to band average or reference detector, Band 1, low gain<br>Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction) |
| GROUP: STRIPING GROUP: STRIPING_FLAG_LOW | Correction_ Reference_B2_Low | Static | uint8 | Striping correction methodology flag, relative to band average or reference detector, Band 2, low gain<br>Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction) |
| GROUP: STRIPING GROUP: STRIPING_FLAG_LOW | Correction_ Reference_B3_Low | Static | uint8 | Striping correction methodology flag, relative to band average or reference detector, Band 3, low gain<br>Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction) |
| GROUP: STRIPING GROUP: STRIPING_FLAG_LOW | Correction_ Reference_B4_Low | Static | uint8 | Striping correction methodology flag, relative to band average or reference detector, Band 4, low gain<br>Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction) |
| GROUP: STRIPING GROUP: STRIPING_FLAG_LOW | Correction_ Reference_B5_Low | Static | uint8 | Striping correction methodology flag, relative to band average or reference detector, Band 5, low gain<br>Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction) |
| GROUP: STRIPING GROUP: STRIPING_FLAG_LOW | Correction_ Reference_B6_Low | Static | uint8 | Striping correction methodology flag, relative to band average or reference detector, Band 6, low gain<br>Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction) |
| GROUP: STRIPING GROUP: STRIPING_FLAG_LOW | Correction_ Reference_B7_Low | Static | uint8 | Striping correction methodology flag, relative to band average or reference detector, Band 7, low gain<br>Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction) |
| GROUP: STRIPING GROUP: STRIPING_FLAG_LOW | Correction_ Reference_B8_Low | Static | uint8 | Striping correction methodology flag, relative to band average or reference detector, Band 8, low gain<br>Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction) |
| GROUP: STRIPING GROUP: STRIPING_FLAG_HIGH | Correction_ Reference_B1_High | Static | uint8 | Striping correction methodology flag, relative to band average or reference detector, Band 1, high gain<br>Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction) |
| GROUP: STRIPING GROUP: STRIPING_FLAG_HIGH | Correction_ Reference_B2_High | Static | uint8 | Striping correction methodology flag, relative to band average or reference detector, Band 2, high gain<br>Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction) |
| GROUP: STRIPING GROUP: STRIPING_FLAG_HIGH | Correction_ Reference_B3_High | Static | uint8 | Striping correction methodology flag, relative to band average or reference detector, Band 3, high gain<br>Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction) |
| GROUP: STRIPING GROUP: STRIPING_FLAG_HIGH | Correction_ Reference_B4_High | Static | uint8 | Striping correction methodology flag, relative to band average or reference detector, Band 4, high gain<br>Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction) |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: STRIPING GROUP: STRIPING_FLAG_HIGH | Correction_ Reference_B5_High | Static | uint8 | Striping correction methodology flag, relative to band average or reference detector, Band 5, high gain<br>Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction) |
| GROUP: STRIPING GROUP: STRIPING_FLAG_HIGH | Correction_ Reference_B6_High | Static | uint8 | Striping correction methodology flag, relative to band average or reference detector, Band 6, high gain<br>Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction) |
| GROUP: STRIPING GROUP: STRIPING_FLAG_HIGH | Correction_ Reference_B7_High | Static | uint8 | Striping correction methodology flag, relative to band average or reference detector, Band 7, high gain<br>Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction) |
| GROUP: STRIPING GROUP: STRIPING_FLAG_HIGH | Correction_ Reference_B8_High | Static | uint8 | Striping correction methodology flag, relative to band average or reference detector, Band 8, high gain<br>Valid format: N, where N = 0 (band average), 1 (reference detector), or 2 (no correction) |
| GROUP: HISTOGRAM GROUP: DETECTOR_NOISE GROUP: DETECTOR_NOISE_LOW | Detector_Noise_ Level_B1_Low | Dynamic | float32 array (16 values) | Standard deviation of image region data for each detector of Band 1, low gain<br>Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DETECTOR_NOISE GROUP: DETECTOR_NOISE_LOW | Detector_Noise_ Level_B2_Low | Dynamic | float32 array (16 values) | Standard deviation of image region data for each detector of Band 2, low gain<br>Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DETECTOR_NOISE GROUP: DETECTOR_NOISE_LOW | Detector_Noise_ Level_B3_Low | Dynamic | float32 array (16 values) | Standard deviation of image region data for each detector of Band 3, low gain<br>Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DETECTOR_NOISE GROUP: DETECTOR_NOISE_LOW | Detector_Noise_ Level_B4_Low | Dynamic | float32 array (16 values) | Standard deviation of image region data for each detector of Band 4, low gain<br>Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DETECTOR_NOISE GROUP: DETECTOR_NOISE_LOW | Detector_Noise_ Level_B5_Low | Dynamic | float32 array (16 values) | Standard deviation of image region data for each detector of Band 5, low gain<br>Valid format: N.NNNNNN, where N = 0 to 9, where NN.NNNN = CPF |
| GROUP: HISTOGRAM GROUP: DETECTOR_NOISE GROUP: DETECTOR_NOISE_LOW | Detector_Noise_ Level_B6_Low | Dynamic | float32 array (8 values) | Standard deviation of image region data for each detector of Band 6, low gain<br>Valid format: N.NNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DETECTOR_NOISE GROUP: DETECTOR_NOISE_LOW | Detector_Noise_ Level_B7_Low | Dynamic | float32 array (16 values) | Standard deviation of image region data for each detector of Band 7, low gain<br>Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DETECTOR_NOISE GROUP: DETECTOR_NOISE_LOW | Detector_Noise_ Level_B8_Low | Dynamic | float32 array (32 values) | Standard deviation of image region data for each detector of Band 8, low gain<br>Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DETECTOR_NOISE GROUP: DETECTOR_NOISE_HIGH | Detector_Noise_ Level_B1_High | Dynamic | float32 array (16 values) | Standard deviation of image region data for each detector of Band 1, high gain<br>Valid format: N.NNNNNN, where N = 0 to 9 |

IAS-207
Version 6.0

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: HISTOGRAM GROUP: DETECTOR_NOISE GROUP: DETECTOR_NOISE_HIGH | Detector_Noise_ Level_B2_High | Dynamic | float32 array (16 values) | Standard deviation of image region data for each detector of Band 2, high gain Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DETECTOR_NOISE GROUP: DETECTOR_NOISE_HIGH | Detector_Noise_ Level_B3_High | Dynamic | float32 array (16 values) | Standard deviation of image region data for each detector of Band 3, high gain Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DETECTOR_NOISE GROUP: DETECTOR_NOISE_HIGH | Detector_Noise_ Level_B4_High | Dynamic | float32 array (16 values) | Standard deviation of image region data for each detector of Band 4, high gain Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DETECTOR_NOISE GROUP: DETECTOR_NOISE_HIGH | Detector_Noise_ Level_B5_High | Dynamic | float32 array (16 values) | Standard deviation of image region data for each detector of Band 5, high gain Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DETECTOR_NOISE GROUP: DETECTOR_NOISE_HIGH | Detector_Noise_ Level_B6_High | Dynamic | float32 array (8 values) | Standard deviation of image region data for each detector of Band 6, high gain Valid format: N.NNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DETECTOR_NOISE GROUP: DETECTOR_NOISE_HIGH | Detector_Noise_ Level_B7_High | Dynamic | float32 array (16 values) | Standard deviation of image region data for each detector of Band 7, high gain Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DETECTOR_NOISE GROUP: DETECTOR_NOISE_HIGH | Detector_Noise_ Level_B8_High | Dynamic | float32 array (32 values) | Standard deviation of image region data for each detector of Band 8, high gain Valid format: N.NNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE GROUP: DET_SHUTTER_NOISE_LOW | Det_Shutter_Noise_ Level_B1_Low | Dynamic | float32 array (16 values) | Standard deviation of shutter region data for each detector of Band 1, low gain Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE GROUP: DET_SHUTTER_NOISE_LOW | Det_Shutter_Noise_ Level_B2_Low | Dynamic | float32 array (16 values) | Standard deviation of shutter region data for each detector of Band 2, low gain Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE GROUP: DET_SHUTTER_NOISE_LOW | Det_Shutter_Noise_ Level_B3_Low | Dynamic | float32 array (16 values) | Standard deviation of shutter region data for each detector of Band 3, low gain Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE GROUP: DET_SHUTTER_NOISE_LOW | Det_Shutter_Noise_ Level_B4_Low | Dynamic | float32 array (16 values) | Standard deviation of shutter region data for each detector of Band 4, low gain Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE GROUP: DET_SHUTTER_NOISE_LOW | Det_Shutter_Noise_ Level_B5_Low | Dynamic | float32 array (16 values) | Standard deviation of shutter region data for each detector of Band 5, low gain Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE GROUP: DET_SHUTTER_NOISE_LOW | Det_Shutter_Noise_ Level_B6_Low | Dynamic | float32 array (8 values) | Standard deviation of shutter region data for each detector of Band 6, low gain Valid format: N.NNNNN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE GROUP: DET_SHUTTER_NOISE_LOW | Det_Shutter_Noise_Level_B7_Low | Dynamic | float32 array (16 values) | Standard deviation of shutter region data for each detector of Band 7, low gain<br>Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE GROUP: DET_SHUTTER_NOISE_LOW | Det_Shutter_Noise_Level_B8_Low | Dynamic | float32 array (32 values) | Standard deviation of shutter region data for each detector of Band 8, low gain<br>Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE GROUP: DET_SHUTTER_NOISE_HIGH | Det_Shutter_Noise_Level_B1_High | Dynamic | float32 array (16 values) | Standard deviation of shutter region data for each detector of Band 1, high gain<br>Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE GROUP: DET_SHUTTER_NOISE_HIGH | Det_Shutter_Noise_Level_B2_High | Dynamic | float32 array (16 values) | Standard deviation of shutter region data for each detector of Band 2, high gain<br>Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE GROUP: DET_SHUTTER_NOISE_HIGH | Det_Shutter_Noise_Level_B3_High | Dynamic | float32 array (16 values) | Standard deviation of shutter region data for each detector of Band 3, high gain<br>Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE GROUP: DET_SHUTTER_NOISE_HIGH | Det_Shutter_Noise_Level_B4_High | Dynamic | float32 array (16 values) | Standard deviation of shutter region data for each detector of Band 4, high gain<br>Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE GROUP: DET_SHUTTER_NOISE_HIGH | Det_Shutter_Noise_Level_B5_High | Dynamic | float32 array (16 values) | Standard deviation of shutter region data for each detector of Band 5, high gain<br>Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE GROUP: DET_SHUTTER_NOISE_HIGH | Det_Shutter_Noise_Level_B6_High | Dynamic | float32 array (8 values) | Standard deviation of shutter region data for each detector of Band 6, high gain<br>Valid format: N.NNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE GROUP: DET_SHUTTER_NOISE_HIGH | Det_Shutter_Noise_Level_B7_High | Dynamic | float32 array (16 values) | Standard deviation of shutter region data for each detector of Band 7, high gain<br>Valid format: N.NNNNNN, where N = 0 to 9 |
| GROUP: HISTOGRAM GROUP: DET_SHUTTER_NOISE GROUP: DET_SHUTTER_NOISE_HIGH | Det_Shutter_Noise_Level_B8_High | Dynamic | float32 array (32 values) | Standard deviation of shutter region data for each detector of Band 8, high gain<br>Valid format: N.NNNNN, where N = 0 to 9 |
| 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<br>Valid format: NN, where NN = 15 |

| Parameter<br>Group | Parameter<br>Name | Value<br>Type | Data<br>Type | Description |
|---|---|---|---|---|
| GROUP: HISTOGRAM<br>GROUP:<br>REFERENCE_DETECTORS | Reference_Detector_B2 | Dynamic | uint8 | Detector used as a reference when computing relative detector gains and biases (least noisy), Band 2<br>Valid format: NN, where NN = 12 |
| GROUP: HISTOGRAM<br>GROUP:<br>REFERENCE_DETECTORS | Reference_Detector_B3 | Dynamic | uint8 | Detector used as a reference when computing relative detector gains and biases (least noisy), Band 3<br>Valid format: NN, where NN = 08 |
| GROUP: HISTOGRAM<br>GROUP:<br>REFERENCE_DETECTORS | Reference_Detector_B4 | Dynamic | uint8 | Detector used as a reference when computing relative detector gains and biases (least noisy), Band 4<br>Valid format: NN, where NN = 07 |
| GROUP: HISTOGRAM<br>GROUP:<br>REFERENCE_DETECTORS | Reference_Detector_B5 | Dynamic | uint8 | Detector used as a reference when computing relative detector gains and biases (least noisy), Band 5<br>Valid format: NN, where NN = 14 |
| GROUP: HISTOGRAM<br>GROUP:<br>REFERENCE_DETECTORS | Reference_Detector_B6 | Dynamic | uint8 | Detector used as a reference when computing relative detector gains and biases (least noisy), Band 6<br>Valid format: NN, where NN = 01 |
| GROUP: HISTOGRAM<br>GROUP:<br>REFERENCE_DETECTORS | Reference_Detector_B7 | Dynamic | uint8 | Detector used as a reference when computing relative detector gains and biases (least noisy), Band 7<br>Valid format: NN, where NN = 10 |
| GROUP: HISTOGRAM<br>GROUP:<br>REFERENCE_DETECTORS | Reference_Detector_B8 | Dynamic | uint8 | Detector used as a reference when computing relative detector gains and biases (least noisy), Band 8<br>Valid format: NN, where NN = 27 |
| GROUP: HISTOGRAM<br>GROUP:<br>SATURATION_THRESHOLDS | Saturation_Bin_<br>Threshold_B1 | Dynamic | uint16 | Number of pixels that a bin must have to be tested as a saturation bin, Band 1<br>Valid format: NNNNN, where NNNNN = 1000 |
| GROUP: HISTOGRAM<br>GROUP:<br>SATURATION_THRESHOLDS | Saturation_Bin_<br>Threshold_B2 | Dynamic | uint16 | Number of pixels that a bin must have to be tested as a saturation bin, Band 2<br>Valid format: NNNNN, where NNNNN = 1000 |
| GROUP: HISTOGRAM<br>GROUP:<br>SATURATION_THRESHOLDS | Saturation_Bin_<br>Threshold_B3 | Dynamic | uint16 | Number of pixels that a bin must have to be tested as a saturation bin, Band 3<br>Valid format: NNNNN, where NNNNN = 1000 |
| GROUP: HISTOGRAM<br>GROUP:<br>SATURATION_THRESHOLDS | Saturation_Bin_<br>Threshold_B4 | Dynamic | uint16 | Number of pixels that a bin must have to be tested as a saturation bin, Band 4<br>Valid format: NNNNN, where NNNNN = 1000 |
| GROUP: HISTOGRAM<br>GROUP:<br>SATURATION_THRESHOLDS | Saturation_Bin_<br>Threshold_B5 | Dynamic | uint16 | Number of pixels that a bin must have to be tested as a saturation bin, Band 5<br>Valid format: NNNNN, where NNNNN = 1000 |
| GROUP: HISTOGRAM<br>GROUP:<br>SATURATION_THRESHOLDS | Saturation_Bin_<br>Threshold_B6 | Dynamic | uint16 | Number of pixels that a bin must have to be tested as a saturation bin, Band 6<br>Valid format: NNNNN, where NNNNN = 1000 |
| GROUP: HISTOGRAM<br>GROUP:<br>SATURATION_THRESHOLDS | Saturation_Bin_<br>Threshold_B7 | Dynamic | uint16 | Number of pixels that a bin must have to be tested as a saturation bin, Band 7<br>Valid format: NNNNN, where NNNNN = 1000 |
| GROUP: HISTOGRAM<br>GROUP:<br>SATURATION THRESHOLDS | Saturation_Bin_<br>Threshold_B8 | Dynamic | uint16 | Number of pixels that a bin must have to be tested as a saturation bin, Band 8<br>Valid format: NNNNN, where NNNNN = 1000 |
| GROUP: HISTOGRAM<br>GROUP: ADJACENT_BINS<br>GROUP: BIN_NUMBER | Adjacent_Bin_<br>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 saturation bin, Band 1<br>Valid format: N, where N = 2 (default) |

| Parameter Group | 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 possible bin as 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 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 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 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 saturation bin, Band 6 Valid format: N, where N = 2 (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 saturation bin, Band 7 Valid format: N, where N = 2 (default) |
| GROUP: HISTOGRAM GROUP: ADJACENT_BINS GROUP: BIN_NUMBER | Adjacent_Bin_ Number_B8 | Dynamic | uint8 | Bins adjacent to a possible saturation bin that must have fewer pixels than "adjacent bin threshold" to declare a possible bin as saturation bin, Band 8 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: NN, where NN = 10 (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: NN, where NN = 10 (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: NN, where NN = 10 (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: NN, where NN = 10 (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: NN, where NN = 10 (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: NN, where NN = 10 (default) |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: HISTOGRAM<br>GROUP: ADJACENT_BINS<br>GROUP: BIN_THRESHOLD | Adjacent_Bin_<br>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<br>Valid format: NN, where NN = 10 (default) |
| GROUP: HISTOGRAM<br>GROUP: ADJACENT_BINS<br>GROUP: BIN_THRESHOLD | Adjacent_Bin_<br>Threshold_B8 | Dynamic | uint8 | Number of adjacent bin pixels that cannot be exceeded for the Band 8 candidate saturation bin to be a valid saturation bin<br>Valid format: NN, where NN = 10 (default) |
| GROUP: HISTOGRAM<br>GROUP: STARTING_PIXEL | Start_pixel_B1 | Dynamic | uint8 | Leftmost pixel in the window to be tested, Band 1<br>Valid format: NNN, where NNN = 243 |
| GROUP: HISTOGRAM<br>GROUP: STARTING_PIXEL | Start_pixel_B2 | Dynamic | uint8 | Leftmost pixel in the window to be tested, Band 2<br>Valid format: NNN, where NNN = 218 |
| GROUP: HISTOGRAM<br>GROUP: STARTING_PIXEL | Start_pixel_B3 | Dynamic | uint8 | Leftmost pixel in the window to be tested, Band 3<br>Valid format: NNN, where NNN = 193 |
| GROUP: HISTOGRAM<br>GROUP: STARTING_PIXEL | Start_pixel_B4 | Dynamic | uint8 | Leftmost pixel in the window to be tested, Band 4<br>Valid format: NNN, where NNN = 168 |
| GROUP: HISTOGRAM<br>GROUP: STARTING_PIXEL | Start_pixel_B5 | Dynamic | uint8 | Leftmost pixel in the window to be tested, Band 5<br>Valid format: NNN, where NNN = 97 |
| GROUP: HISTOGRAM<br>GROUP: STARTING_PIXEL | Start_pixel_B6 | Dynamic | uint8 | Leftmost pixel in the window to be tested, Band 6<br>Valid format: NNN, where NNN = 31 |
| GROUP: HISTOGRAM<br>GROUP: STARTING_PIXEL | Start_pixel_B7 | Dynamic | uint8 | Leftmost pixel in the window to be tested, Band 7<br>Valid format: NNN, where NNN = 123 |
| GROUP: HISTOGRAM<br>GROUP: STARTING_PIXEL | Start_pixel_B8 | Dynamic | uint8 | Leftmost pixel in the window to be tested, Band 8<br>Valid format: NNN, where NNN = 536 |
| GROUP: HISTOGRAM<br>GROUP: WINDOW_WIDTH | Window_Samples_B1 | Dynamic | uint8 | Width of the window, in pixels, to be tested, Band 1<br>Valid format: NNNNN, where NNNNN = 5874 |
| GROUP: HISTOGRAM<br>GROUP: WINDOW_WIDTH | Window_Samples_B2 | Dynamic | uint8 | Width of the window, in pixels, to be tested, Band 2<br>Valid format: NNNNN, where NNNNN = 5874 |
| GROUP: HISTOGRAM<br>GROUP: WINDOW_WIDTH | Window_Samples_B3 | Dynamic | uint8 | Width of the window, in pixels, to be tested, Band 3<br>Valid format: NNNNN, where NNNNN = 5874 |
| GROUP: HISTOGRAM<br>GROUP: WINDOW_WIDTH | Window_Samples_B4 | Dynamic | uint8 | Width of the window, in pixels, to be tested, Band 4<br>Valid format: NNNNN, where NNNNN = 5874 |
| GROUP: HISTOGRAM<br>GROUP: WINDOW_WIDTH | Window_Samples_B5 | Dynamic | uint8 | Width of the window, in pixels, to be tested, Band 5<br>Valid format: NNNNN, where NNNNN = 5874 |
| GROUP: HISTOGRAM<br>GROUP: WINDOW_WIDTH | Window_Samples_B6 | Dynamic | uint8 | Width of the window, in pixels, to be tested, Band 6<br>Valid format: NNNNN, where NNNNN = 2937 |
| GROUP: HISTOGRAM<br>GROUP: WINDOW_WIDTH | Window_Samples_B7 | Dynamic | uint8 | Width of the window, in pixels, to be tested, Band 7<br>Valid format: NNNNN, where NNNNN = 5874 |
| GROUP: HISTOGRAM<br>GROUP: WINDOW_WIDTH | Window_Samples_B8 | Dynamic | uint8 | Width of the window, in pixels, to be tested, Band 8<br>Valid format: NNNNN, where NNNNN = 11748 |
| GROUP: HISTOGRAM<br>GROUP: WINDOW_LENGTH | Window_Scans_B1 | Dynamic | uint8 | Number of scans in the window to be tested, Band 1<br>Valid format: NNN, where NNN = 375 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: HISTOGRAM GROUP: WINDOW_LENGTH | Window_Scans_B2 | Dynamic | uint8 | Number of scans in the window to be tested, Band 2<br>Valid format: NNN, where NNN = 375 |
| GROUP: HISTOGRAM GROUP: WINDOW_LENGTH | Window_Scans_B3 | Dynamic | uint8 | Number of scans in the window to be tested, Band 3<br>Valid format: NNN, where NNN = 375 |
| GROUP: HISTOGRAM GROUP: WINDOW_LENGTH | Window_Scans_B4 | Dynamic | uint8 | Number of scans in the window to be tested, Band 4<br>Valid format: NNN, where NNN = 375 |
| GROUP: HISTOGRAM GROUP: WINDOW_LENGTH | Window_Scans_B5 | Dynamic | uint8 | Number of scans in the window to be tested, Band 5<br>Valid format: NNN, where NNN = 375 |
| GROUP: HISTOGRAM GROUP: WINDOW_LENGTH | Window_Scans_B6 | Dynamic | uint8 | Number of scans in the window to be tested, Band 6<br>Valid format: NNN, where NNN = 375 |
| GROUP: HISTOGRAM GROUP: WINDOW_LENGTH | Window_Scans_B7 | Dynamic | uint8 | Number of scans in the window to be tested, Band 7<br>Valid format: NNN, where NNN = 375 |
| GROUP: HISTOGRAM GROUP: WINDOW_LENGTH | Window_Scans_B8 | Dynamic | uint8 | Number of scans in the window to be tested, Band 8<br>Valid format: NNN, where NNN = 375 |
| GROUP: HISTOGRAM GROUP: OVERLAPPING_SCANS | Overlap_Scans_B1 | Dynamic | uint8 | Number of overlapping scans between the windows to be tested, Band 1<br>Valid format: NNN, where NNN = 0 |
| GROUP: HISTOGRAM GROUP: OVERLAPPING_SCANS | Overlap_Scans_B2 | Dynamic | uint8 | Number of overlapping scans between the windows to be tested, Band 2<br>Valid format: NNN, where NNN = 0 |
| GROUP: HISTOGRAM GROUP: OVERLAPPING_SCANS | Overlap_Scans_B3 | Dynamic | uint8 | Number of overlapping scans between the windows to be tested, Band 3<br>Valid format: NNN, where NNN = 0 |
| GROUP: HISTOGRAM GROUP: OVERLAPPING_SCANS | Overlap_Scans_B4 | Dynamic | uint8 | Number of overlapping scans between the windows to be tested, Band 4<br>Valid format: NNN, where NNN = 0 |
| GROUP: HISTOGRAM GROUP: OVERLAPPING_SCANS | Overlap_Scans_B5 | Dynamic | uint8 | Number of overlapping scans between the windows to be tested, Band 5<br>Valid format: NNN, where NNN = 0 |
| GROUP: HISTOGRAM GROUP: OVERLAPPING_SCANS | Overlap_Scans_B6 | Dynamic | uint8 | Number of overlapping scans between the windows to be tested, Band 6<br>Valid format: NNN, where NNN = 0 |
| GROUP: HISTOGRAM GROUP: OVERLAPPING_SCANS | Overlap_Scans_B7 | Dynamic | uint8 | Number of overlapping scans between the windows to be tested, Band 7<br>Valid format: NNN, where NNN = 0 |
| GROUP: HISTOGRAM GROUP: OVERLAPPING_SCANS | Overlap_Scans_B8 | Dynamic | uint8 | Number of overlapping scans between the windows to be tested, Band 8<br>Valid format: NNN, where NNN = 0 |
| GROUP: IMPULSE_NOISE | Median_Filter_Width | Static | uint8 | Width of median filter<br>Valid format: N, where N = 3 |
| GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD | B1L_Threshold | Dynamic | float32 array (16 values) | Band 1 low-gain noise threshold for an inequal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD | B2L_Threshold | Dynamic | float32 array (16 values) | Band 2 low-gain noise threshold for an inequal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD | B3L_Threshold | Dynamic | float32 array (16 values) | Band 3 low-gain noise threshold for an inequal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD | B4L_Threshold | Dynamic | float32 array (16 values) | Band 4 low-gain noise threshold for an inequal case<br>Valid format: NN.NN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD | B5L_Threshold | Dynamic | float32 array (16 values) | Band 5 low-gain noise threshold for an inequal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD | B6L_Threshold | Dynamic | float32 array (8 values) | Band 6 low-gain noise threshold for an inequal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD | B7L_Threshold | Dynamic | float32 array (16 values) | Band 7 low-gain noise threshold for an inequal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD | B8L_Threshold | Dynamic | float32 array (32 values) | Band 8 low-gain noise threshold for an inequal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD | B1H_Threshold | Dynamic | float32 array (16 values) | Band 1 high-gain noise threshold for an inequal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD | B2H_Threshold | Dynamic | float32 array (16 values) | Band 2 high-gain noise threshold for an inequal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD | B3H_Threshold | Dynamic | float32 array (16 values) | Band 3 high-gain noise threshold for an inequal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD | B4H_Threshold | Dynamic | float32 array (16 values) | Band 4 high-gain noise threshold for an inequal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD | B5H_Threshold | Dynamic | float32 array (16 values) | Band 5 high-gain noise threshold for an inequal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD | B6H_Threshold | Dynamic | float32 array (8 values) | Band 6 high-gain noise threshold for an inequal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD | B7H_Threshold | Dynamic | float32 array (16 values) | Band 7 high-gain noise threshold for an inequal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_THRESHOLD | B8H_Threshold | Dynamic | float32 array (32 values) | Band 8 high-gain noise threshold for an inequal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD | B1L_Sigma_Threshold | Dynamic | float32 array (16 values) | Band 1 low-gain noise threshold for an equal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD | B2L_Sigma_Threshold | Dynamic | float32 array (16 values) | Band 2 low-gain noise threshold for an equal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD | B3L_Sigma_Threshold | Dynamic | float32 array (16 values) | Band 3 low-gain noise threshold for an equal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD | B4L_Sigma_Threshold | Dynamic | float32 array (16 values) | Band 4 low-gain noise threshold for an equal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD | B5L_Sigma_Threshold | Dynamic | float32 array (16 values) | Band 5 low-gain noise threshold for an equal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD | B6L_Sigma_Threshold | Dynamic | float32 array (8 values) | Band 6 low-gain noise threshold for an equal case<br>Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD | B7L_Sigma_Threshold | Dynamic | float32 array (16 values) | Band 7 low-gain noise threshold for an equal case<br>Valid format: NN.NN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD | B8L_Sigma_Threshold | Dynamic | float32 array (32 values) | Band 8 low-gain noise threshold for an equal case Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD | B1H_Sigma_Threshold | Dynamic | float32 array (16 values) | Band 1 high-gain noise threshold for an equal case Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD | B2H_Sigma_Threshold | Dynamic | float32 array (16 values) | Band 2 high-gain noise threshold for an equal case Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD | B3H_Sigma_Threshold | Dynamic | float32 array (16 values) | Band 3 high-gain noise threshold for an equal case Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD | B4H_Sigma_Threshold | Dynamic | float32 array (16 values) | Band 4 high-gain noise threshold for an equal case Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD | B5H_Sigma_Threshold | Dynamic | float32 array (16 values) | Band 5 high-gain noise threshold for an equal case Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD | B6H_Sigma_Threshold | Dynamic | float32 array (8 values) | Band 6 high-gain noise threshold for an equal case Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD | B7H_Sigma_Threshold | Dynamic | float32 array (16 values) | Band 7 high-gain noise threshold for an equal case Valid format: NN.NN, where N = 0 to 9 |
| GROUP: IMPULSE_NOISE GROUP: IN_SIGMA_THRESHOLD | B8H_Sigma_Threshold | Dynamic | float32 array (32 values) | Band 8 high-gain noise threshold for an equal case Valid format: NN.NN, 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: NNNNNNN, 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: NNNNNNN, 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: NNNNNNN, 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: NNNNNNN, 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: NNNNNNN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| 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: NNNNNNN, 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: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_MEANS | B8_Frequency_Mean | Dynamic | float32 array (10 values) | Band 8 frequency means measured in inverse minor frames Valid format: NNNNNNN, 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: NNNNNNN, 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: NNNNNNN, 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: NNNNNNN, where N = 0 to 9 |
| 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: NNNNNNN, 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: NNNNNNN, 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: NNNNNNN, 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: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_SIGMAS | B8_Frequency_Sigma | Dynamic | float32 array (10 values) | Band 8 frequency sigmas measured in inverse minor frames Valid format: NNNNNNN, 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: NNNNNNN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| 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: NNNNNNN, 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: NNNNNNN, 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: NNNNNNN, 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: NNNNNNN, 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: NNNNNNN, 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: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_ MINIMUMS | B8_Frequency_Min | Dynamic | float32 array (10 values) | Band 8 frequency minimums measured in inverse minor frames Valid format: NNNNNNN, 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: NNNNNNN, where N = 0 to 9 |
| 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: NNNNNNN, 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: NNNNNNN, 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: NNNNNNN, 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: NNNNNNN, where N = 0 to 9 |

IAS-207
Version 6.0

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| 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: NNNNNNN, 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: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_FREQUENCY_ PARAMETERS GROUP: FREQUENCY_ MAXIMUMS | B8_Frequency_Max | Dynamic | float32 array (10 values) | Band 8 frequency maximums measured in inverse minor frames Valid format: NNNNNNN, 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: NNNNNNN, 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: NNNNNNN, 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: NNNNNNN, 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: NNNNNNN, 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: NNNNNNN, 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: NNNNNNN, 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: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MEANS | B8_Phase_Mean | Dynamic | float32 array (10 values) | Band 8 phase means measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_SIGMAS | B1_Phase_Sigma | Dynamic | float32 array (10 values) | Band 1 phase sigmas measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_SIGMAS | B2_Phase_Sigma | Dynamic | float32 array (10 values) | Band 2 phase sigmas measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_SIGMAS | B3_Phase_Sigma | Dynamic | float32 array (10 values) | Band 3 phase sigmas measured in radians Valid format: NNNNNNN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_SIGMAS | B4_Phase_Sigma | Dynamic | float32 array (10 values) | Band 4 phase sigmas measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_SIGMAS | B5_Phase_Sigma | Dynamic | float32 array (10 values) | Band 5 phase sigmas measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_SIGMAS | B6_Phase_Sigma | Dynamic | float32 array (10 values) | Band 6 phase sigmas measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_SIGMAS | B7_Phase_Sigma | Dynamic | float32 array (10 values) | Band 7 phase sigmas measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_SIGMAS | B8_Phase_Sigma | Dynamic | float32 array (10 values) | Band 8 phase sigmas measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MINIMUMS | B1_Phase_Min | Dynamic | float32 array (10 values) | Band 1 phase minimums measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MINIMUMS | B2_Phase_Min | Dynamic | float32 array (10 values) | Band 2 phase minimums measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MINIMUMS | B3_Phase_Min | Dynamic | float32 array (10 values) | Band 3 phase minimums measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MINIMUMS | B4_Phase_Min | Dynamic | float32 array (10 values) | Band 4 phase minimums measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MINIMUMS | B5_Phase_Min | Dynamic | float32 array (10 values) | Band 5 phase minimums measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MINIMUMS | B6_Phase_Min | Dynamic | float32 array (10 values) | Band 6 phase minimums measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MINIMUMS | B7_Phase_Min | Dynamic | float32 array (10 values) | Band 7 phase minimums measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MINIMUMS | B8_Phase_Min | Dynamic | float32 array (10 values) | Band 8 phase minimums measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MAXIMUMS | B1_Phase_Max | Dynamic | float32 array (10 values) | Band 1 phase maximums measured in radians Valid format: NNNNNNN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MAXIMUMS | B2_Phase_Max | Dynamic | float32 array (10 values) | Band 2 phase maximums measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MAXIMUMS | B3_Phase_Max | Dynamic | float32 array (10 values) | Band 3 phase maximums measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MAXIMUMS | B4_Phase_Max | Dynamic | float32 array (10 values) | Band 4 phase maximums measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MAXIMUMS | B5_Phase_Max | Dynamic | float32 array (10 values) | Band 5 phase maximums measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MAXIMUMS | B6_Phase_Max | Dynamic | float32 array (10 values) | Band 6 phase maximums measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MAXIMUMS | B7_Phase_Max | Dynamic | float32 array (10 values) | Band 7 phase maximums measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_PHASE_PARAMETERS GROUP: PHASE_MAXIMUMS | B8_Phase_Max | Dynamic | float32 array (10 values) | Band 8 phase maximums measured in radians Valid format: NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MEANS | B1_Magnitude_Mean | Dynamic | float32 array (10 values) | Band 1 magnitudes 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 magnitudes 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 magnitudes 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 magnitudes 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 magnitudes 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 magnitudes means measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MEANS | B7_Magnitude_Mean | Dynamic | float32 array (10 values) | Band 7 magnitudes means measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MEANS | B8_Magnitude_Mean | Dynamic | float32 array (10 values) | Band 8 magnitudes 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 magnitudes 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 magnitudes 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 magnitudes 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 magnitudes 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 magnitudes sigmas measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_SIGMAS | B6_Magnitude_Sigma | Dynamic | float32 array (10 values) | Band 6 magnitudes 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 magnitudes sigmas measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_SIGMAS | B8_Magnitude_Sigma | Dynamic | float32 array (10 values) | Band 8 magnitudes 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 magnitudes 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 magnitudes minimums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MINIMUMS | B3_Magnitude_Min | Dynamic | float32 array (10 values) | Band 3 magnitudes 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 magnitudes 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 magnitudes 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 magnitudes 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 magnitudes minimums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MINIMUMS | B8_Magnitude_Min | Dynamic | float32 array (10 values) | Band 8 magnitudes 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 magnitudes 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 magnitudes 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 magnitudes maximums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_ MAXIMUMS | B4_Magnitude_Max | Dynamic | float32 array (10 values) | Band 4 magnitudes 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 magnitudes 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 magnitudes maximums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MAXIMUMS | B7_Magnitude_Max | Dynamic | float32 array (10 values) | Band 7 magnitudes maximums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9 |
| GROUP: COHERENT_NOISE GROUP: CN_MAGNITUDE_ PARAMETERS GROUP: MAGNITUDE_MAXIMUMS | B8_Magnitude_Max | Dynamic | float32 array (10 values) | Band 8 magnitudes maximums measured in DNs Valid format: NNN.NNNNNNN, where N = 0 to 9 |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_LOW | High_AD_Level_B1_low | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the high end; Band 1, low gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_LOW | High_AD_Level_B2_low | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the high end; Band 2, low gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_LOW | High_AD_Level_B3_low | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the high end; Band 3, low gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_LOW | High_AD_Level_B4_low | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the high end; Band 4, low gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_LOW | High_AD_Level_B5_low | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the high end; Band 5, low gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_LOW | High_AD_Level_B6_low | Dynamic | uint8 array ( 8 values) | Digital count at which the analog-to-digital converter saturates at the high end; Band 6, low gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_LOW | High_AD_Level_B7_low | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the high end; Band 7, low gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_LOW | High_AD_Level_B8_low | Dynamic | uint8 array (32 values) | Digital count at which the analog-to-digital converter saturates at the high end; Band 8, low gain Valid format: NNN, where NNN = 255 (default) |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_LOW | Low_AD_Level_B1_low | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the low end; Band 1, low gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_LOW | Low_AD_Level_B2_low | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the low end; Band 2, low gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_LOW | Low_AD_Level_B3_low | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the low end; Band 3, low gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_LOW | Low_AD_Level_B4_low | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the low end; Band 4, low gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_LOW | Low_AD_Level_B5_low | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the low end; Band 5, low gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_LOW | Low_AD_Level_B6_low | Dynamic | uint8 array (8 values) | Digital count at which the analog-to-digital converter saturates at the low end; Band 6, low gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_LOW | Low_AD_Level_B7_low | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the low end; Band 7, low gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_LOW | Low_AD_Level_B8_low | Dynamic | uint8 array (32 values) | Digital count at which the analog-to-digital converter saturates at the low end; Band 8, low gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_HIGH | High_AD_Level_B1_high | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the high end; Band 1, high gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_HIGH | High_AD_Level_B2_high | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the high end; Band 2, high gain Valid format: NNN, where NNN = 255 (default) |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_HIGH | High_AD_Level_B3_high | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the high end; Band 3, high gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_HIGH | High_AD_Level_B4_high | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the high end; Band 4, high gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_HIGH | High_AD_Level_B5_high | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the high end; Band 5, high gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_HIGH | High_AD_Level_B6_high | Dynamic | uint8 array (8 values) | Digital count at which the analog-to-digital converter saturates at the high end; Band 6, high gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_HIGH | High_AD_Level_B7_high | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the high end; Band 7, high gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_HIGH | High_AD_Level_B8_high | Dynamic | uint8 array ( 32 values) | Digital count at which analog-to-digital converter saturates at high end; Band 8, gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_HIGH | Low_AD_Level_B1_high | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the low end; Band 1, high gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_HIGH | Low_AD_Level_B2_high | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the low end; Band 2, high gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_HIGH | Low_AD_Level_B3_high | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the low end; Band 3, high gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_HIGH | Low_AD_Level_B4_high | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the low end; Band 4, high gain Valid format: NNN, where NNN = 0 (default) |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_HIGH | Low_AD_Level_B5_high | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the low end; Band 5, high gain<br>Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_HIGH | Low_AD_Level_B6_high | Dynamic | uint8 array ( 8 values) | Digital count at which the analog-to-digital converter saturates at the low end; Band 6, high gain<br>Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_HIGH | Low_AD_Level_B7_high | Dynamic | uint8 array (16 values) | Digital count at which the analog-to-digital converter saturates at the low end; Band 7, high gain<br>Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: AD_CONVERTER_ SATURATION GROUP: AD_CONVERTER_ SATURATION_HIGH | Low_AD_Level_B8_high | Dynamic | uint8 array ( 32 values) | Digital count at which the analog-to-digital converter saturates at the low end; Band 8, gain<br>Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_LOW | High_Analog_Level_ B1_low | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the high end; Band 1, low gain<br>Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_LOW | High_Analog_Level_ B2_low | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the high end; Band 2, low gain<br>Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_LOW | High_Analog_Level_ B3_low | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the high end; Band 3, low gain<br>Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_LOW | High_Analog_Level_ B4_low | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the high end; Band 4, low gain<br>Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_LOW | High_Analog_Level_ B5_low | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the high end; Band 5, low gain<br>Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_LOW | High_Analog_Level_ B6_low | Dynamic | uint8 array (8 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the high end; Band 6, low gain<br>Valid format: NNN, where NNN = 255 (default) |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_LOW | High_Analog_Level_ B7_low | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the high end; Band 7, low gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_LOW | High_Analog_Level_ B8_low | Dynamic | uint8 array (32 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the high end; Band 8, low gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_LOW | Low_Analog_Level_ B1_low | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the low end; Band 1, low gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_LOW | Low_Analog_Level_ B2_low | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the low end; Band 2, low gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_LOW | Low_Analog_Level_ B3_low | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the low end; Band 3, low gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_LOW | Low_Analog_Level_ B4_low | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the low end; Band 4, low gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_LOW | Low_Analog_Level_ B5_low | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at low end; Band 5, low gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_LOW | Low_Analog_Level_ B6_low | Dynamic | uint8 array (8 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the low end; Band 6, low gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_LOW | Low_Analog_Level_ B7_low | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the low end; Band 7, low gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_LOW | Low_Analog_Level_ B8_low | Dynamic | uint8 array (32 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the low end; Band 8, low gain Valid format: NNN, where NNN = 0 (default) |

IAS-207
Version 6.0

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_HIGH | High_Analog_Level_ B1_high | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the high end; Band 1, high gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_HIGH | High_Analog_Level_ B2_high | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the high end; Band 2, high gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_HIGH | High_Analog_Level_ B3_high | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the high end; Band 3, high gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_HIGH | High_Analog_Level_ B4_high | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the high end; Band 4, high gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_HIGH | High_Analog_Level_ B5_high | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the high end; Band 5, high gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_HIGH | High_Analog_Level_ B6_high | Dynamic | uint8 array (8 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the high end; Band 6, high gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_HIGH | High_Analog_Level_ B7_high | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the high end; Band 7, high gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_HIGH | High_Analog_Level_ B8_high | Dynamic | uint8 array (32 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the high end; Band 8, high gain Valid format: NNN, where NNN = 255 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_HIGH | Low_Analog_Level_ B1_high | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the low end; Band 1, high gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_HIGH | Low_Analog_Level_ B2_high | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the low end; Band 2, high gain Valid format: NNN, where NNN = 0 (default) |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_HIGH | Low_Analog_Level_ B3_high | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the low end; Band 3, high gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_HIGH | Low_Analog_Level_ B4_high | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the low end; Band 4, high gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_HIGH | Low_Analog_Level_ B5_high | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the low end; Band 5, high gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_HIGH | Low_Analog_Level_ B6_high | Dynamic | uint8 array (8 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the low end; Band 6, high gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_HIGH | Low_Analog_Level_ B7_high | Dynamic | uint8 array (16 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the low end; Band 7, high gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: DETECTOR_SATURATION GROUP: ANALOG_SIGNAL_ SATURATION GROUP: ANALOG_SIGNAL_ SATURATION_HIGH | Low_Analog_Level_ B8_high | Dynamic | uint8 array (32 values) | Digital count corresponding to the signal level at which the analog portion of the signal chain saturates at the low end; Band 8, high gain Valid format: NNN, where NNN = 0 (default) |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B1L_RTemp_Prelaunch | Static | float64 | Band 1 prelaunch low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B1L_RTemp_Postlaunch | Static | float64 | Band 1 postlaunch low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B1L_RTemp_Current | Dynamic | float64 | Band 1 current low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B2L_RTemp_Prelaunch | Static | float64 | Band 2 prelaunch low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B2L_RTemp_Postlaunch | Static | float64 | Band 2 postlaunch low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B2L_RTemp_Current | Dynamic | float64 | Band 2 current low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B3L_RTemp_Prelaunch | Static | float64 | Band 3 prelaunch low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B3L_RTemp_Postlaunch | Static | float64 | Band 3 postlaunch low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B3L_RTemp_Current | Dynamic | float64 | Band 3 current low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B4L_RTemp_Prelaunch | Static | float64 | Band 4 prelaunch low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B4L_RTemp_Postlaunch | Static | float64 | Band 4 postlaunch low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B4L_RTemp_Current | Dynamic | float64 | Band 4 current low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B5L_RTemp_Prelaunch | Static | float64 | Band 5 prelaunch low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = -182.1 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B5L_RTemp_Postlaunch | Static | float64 | Band 5 postlaunch low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = -182.1 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B5L_RTemp_Current | Dynamic | float64 | Band 5 current low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = -182.1 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B6L_RTemp_Prelaunch | Static | float64 | Band 6 prelaunch low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = -182.1 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B6L_RTemp_Postlaunch | Static | float64 | Band 6 postlaunch low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = -182.1 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B6L_RTemp_Current | Dynamic | float64 | Band 6 current low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = -182.1 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B7L_RTemp_Prelaunch | Static | float64 | Band 7 prelaunch low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = -182.1 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B7L_RTemp_Postlaunch | Static | float64 | Band 7 postlaunch low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = -182.1 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B7L_RTemp_Current | Dynamic | float64 | Band 7 current low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = -182.1 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B8L_RTemp_Prelaunch | Static | float64 | Band 8 prelaunch low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B8L_RTemp_Postlaunch | Static | float64 | Band 8 postlaunch low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_LOW | B8L_RTemp_Current | Dynamic | float64 | Band 8 current low-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_HIGH | B1H_RTemp_Prelaunch | Static | float64 | Band 1 prelaunch high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_HIGH | B1H_RTemp_Postlaunch | Static | float64 | Band 1 postlaunch high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_HIGH | B1H_RTemp_Current | Dynamic | float64 | Band 1 current high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_HIGH | B2H_RTemp_Prelaunch | Static | float64 | Band 2 prelaunch high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_HIGH | B2H_RTemp_Postlaunch | Static | float64 | Band 2 postlaunch high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_HIGH | B2H_RTemp_Current | Dynamic | float64 | Band 2 current high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_HIGH | B3H_RTemp_Prelaunch | Static | float64 | Band 3 prelaunch high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_HIGH | B3H_RTemp_Postlaunch | Static | float64 | Band 3 postlaunch high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_HIGH | B3H_RTemp_Current | Dynamic | float64 | Band 3 current high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_HIGH | B4H_RTemp_Prelaunch | Static | float64 | Band 4 prelaunch high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_HIGH | B4H_RTemp_Postlaunch | Static | float64 | Band 4 postlaunch high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_ TEMPERATURES GROUP: REFERENCE_HIGH | B4H_RTemp_Current | Dynamic | float64 | Band 4 current high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFERENCE_TEMPERATURES GROUP: REFERENCE_HIGH | B5H_RTemp_Prelaunch | Static | float64 | Band 5 prelaunch high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = -182.1 |
| GROUP: REFERENCE_TEMPERATURES GROUP: REFERENCE_HIGH | B5H_RTemp_Postlaunch | Static | float64 | Band 5 postlaunch high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = -182.1 |
| GROUP: REFERENCE_TEMPERATURES GROUP: REFERENCE_HIGH | B5H_RTemp_Current | Dynamic | float64 | Band 5 current high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = -182.1 |
| GROUP: REFERENCE_TEMPERATURES GROUP: REFERENCE_HIGH | B6H_RTemp_Prelaunch | Static | float64 | Band 6 prelaunch high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = -182.1 |
| GROUP: REFERENCE_TEMPERATURES GROUP: REFERENCE_HIGH | B6H_RTemp_Postlaunch | Static | float64 | Band 6 postlaunch high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = -182.1 |
| GROUP: REFERENCE_TEMPERATURES GROUP: REFERENCE_HIGH | B6H_RTemp_Current | Dynamic | float64 | Band 6 current high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = -182.1 |
| GROUP: REFERENCE_TEMPERATURES GROUP: REFERENCE_HIGH | B7H_RTemp_Prelaunch | Static | float64 | Band 7 prelaunch high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = -182.1 |
| GROUP: REFERENCE_TEMPERATURES GROUP: REFERENCE_HIGH | B7H_RTemp_Postlaunch | Static | float64 | Band 7 postlaunch high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = -182.1 |
| GROUP: REFERENCE_TEMPERATURES GROUP: REFERENCE_HIGH | B7H_RTemp_Current | Dynamic | float64 | Band 7 current high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = -182.1 |
| GROUP: REFERENCE_TEMPERATURES GROUP: REFERENCE_HIGH | B8H_RTemp_Prelaunch | Static | float64 | Band 8 prelaunch high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_TEMPERATURES GROUP: REFERENCE_HIGH | B8H_RTemp_Postlaunch | Static | float64 | Band 8 postlaunch high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: REFERENCE_TEMPERATURES GROUP: REFERENCE_HIGH | B8H_RTemp_Current | Dynamic | float64 | Band 8 current high-gain calibration reference temperature in degrees C Valid format: SNNN.NNN, where SNNN.NNN = 25.00 |
| GROUP: SENSITIVITY_TEMPERATURES GROUP: SENSITIVITY_LOW | B1L_SCoeff_Prelaunch | Static | float64 array (16 values) | Band 1 prelaunch low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_TEMPERATURES GROUP: SENSITIVITY_LOW | B1L_SCoeff_Postlaunch | Static | float64 array (16 values) | Band 1 postlaunch low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_TEMPERATURES GROUP: SENSITIVITY_LOW | B1L_SCoeff_Current | Dynamic | float64 array (16 values) | Band 1 current low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B2L_SCoeff_Prelaunch | Static | float64 array (16 values) | Band 2 prelaunch low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B2L_SCoeff_Postlaunch | Static | float64 array (16 values) | Band 2 postlaunch low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B2L_SCoeff_Current | Dynamic | float64 array (16 values) | Band 2 current low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B3L_SCoeff_Prelaunch | Static | float64 array (16 values) | Band 3 prelaunch low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B3L_SCoeff_Postlaunch | Static | float64 array (16 values) | Band 3 postlaunch low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B3L_SCoeff_Current | Dynamic | float64 array (16 values) | Band 3 current low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B4L_SCoeff_Prelaunch | Static | float64 array (16 values) | Band 4 prelaunch low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B4L_SCoeff_Postlaunch | Static | float64 array (16 values) | Band 4 postlaunch low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B4L_SCoeff_Current | Dynamic | float64 array (16 values) | Band 4 current low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B5L_SCoeff_Prelaunch | Static | float64 array (16 values) | Band 5 prelaunch low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B5L_SCoeff_Postlaunch | Static | float64 array (16 values) | Band 5 postlaunch low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B5L_SCoeff_Current | Dynamic | float64 array (16 values) | Band 5 current low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B6L_SCoeff_Prelaunch | Static | float64 array (8 values) | Band 6 prelaunch low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B6L_SCoeff_Postlaunch | Static | float64 array (8 values) | Band 6 postlaunch low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B6L_SCoeff_Current | Dynamic | float64 array (8 values) | Band 6 current low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B6L_SCoeffOff_ Prelaunch | Static | float64 array (8 values) | Band 6 prelaunch offset calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B6L_SCoeffOff_ Postlaunch | Static | float64 array (8 values) | Band 6 postlaunch offset calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B6L_SCoeffOff_ Current | Dynamic | float64 array (8 values) | Band 6 current offset calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B7L_SCoeff_ Prelaunch | Static | float64 array (16 values) | Band 7 prelaunch low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B7L_SCoeff_ Postlaunch | Static | float64 array (16 values) | Band 7 postlaunch low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B7L_SCoeff_Current | Dynamic | float64 array (16 values) | Band 7 current low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B8L_SCoeff_ Prelaunch | Static | float64 array (32 values) | Band 8 prelaunch low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B8L_SCoeff_ Postlaunch | Static | float64 array (32 values) | Band 8 postlaunch low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_LOW | B8L_SCoeff_Current | Dynamic | float64 array (32 values) | Band 8 current low-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B1H_SCoeff_Prelaunch | Static | float64 array (16 values) | Band 1 prelaunch high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B1H_SCoeff_Postlaunch | Static | float64 array (16 values) | Band 1 postlaunch high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B1H_SCoeff_Current | Dynamic | float64 array (16 values) | Band 1 current high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B2H_SCoeff_Prelaunch | Static | float64 array (16 values) | Band 2 prelaunch high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B2H_SCoeff_Postlaunch | Static | float64 array (16 values) | Band 2 postlaunch high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B2H_SCoeff_Current | Dynamic | float64 array (16 values) | Band 2 current high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B3H_SCoeff_Prelaunch | Static | float64 array (16 values) | Band 3 prelaunch high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B3H_SCoeff_Postlaunch | Static | float64 array (16 values) | Band 3 postlaunch high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B3H_SCoeff_Current | Dynamic | float64 array (16 values) | Band 3 current high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B4H_SCoeff_Prelaunch | Static | float64 array (16 values) | Band 4 prelaunch high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B4H_SCoeff_Postlaunch | Static | float64 array (16 values) | Band 4 postlaunch high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B4H_SCoeff_Current | Dynamic | float64 array (16 values) | Band 4 current high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B5H_SCoeff_Prelaunch | Static | float64 array (16 values) | Band 5 prelaunch high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B5H_SCoeff_Postlaunch | Static | float64 array (16 values) | Band 5 postlaunch high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B5H_SCoeff_Current | Dynamic | float64 array (16 values) | Band 5 current high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B6H_SCoeff_Prelaunch | Static | float64 array (8 values) | Band 6 prelaunch high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B6H_SCoeff_Postlaunch | Static | float64 array (8 values) | Band 6 postlaunch high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B6H_SCoeff_Current | Dynamic | float64 array (8 values) | Band 6 current high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B6H_SCoeffOff_ Prelaunch | Static | float64 array (8 values) | Band 6 prelaunch offset calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B6H_SCoeffOff_ Postlaunch | Static | float64 array (8 values) | Band 6 postlaunch offset calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B6H_SCoeffOff_ Current | Dynamic | float64 array (8 values) | Band 6 current offset calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B7H_SCoeff_ Prelaunch | Static | float64 array (16 values) | Band 7 prelaunch high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B7H_SCoeff_ Postlaunch | Static | float64 array (16 values) | Band 7 postlaunch high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B7H_SCoeff_Current | Dynamic | float64 array (16 values) | Band 7 current high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B8H_SCoeff_ Prelaunch | Static | float64 array (32 values) | Band 8 prelaunch high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B8H_SCoeff_ Postlaunch | Static | float64 array (32 values) | Band 8 postlaunch high-gain calibration temperature sensitivity coefficient Valid format: SNNN.NNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: SENSITIVITY_ TEMPERATURES GROUP: SENSITIVITY_HIGH | B8H_SCoeff_Current | Dynamic | float64 array (32 values) | Band 8 current high-gain calibration temperature sensitivity coefficient 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 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 coefficients for Lamp 2 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B1L_Rad_State1_ Prelaunch | Static | float32 array (16 values) | Band 1 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 off; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B1L_Rad_State1_ Postlaunch | Static | float32 array (16 values) | Band 1 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 off; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B1L_Rad_State1_ Current | Dynamic | float32 array (16 values) | Band 1 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 off; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B1L_Rad_State2_ Prelaunch | Static | float32 array (16 values) | Band 1 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B1L_Rad_State2_ Postlaunch | Static | float32 array (16 values) | Band 1 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B1L_Rad_State2_ Current | Dynamic | float32 array (16 values) | Band 1 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B1L_Rad_State3_ Prelaunch | Static | float32 array (16 values) | Band 1 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: LAMP_RADIANCE<br>GROUP: LAMP_RADIANCE_LOW | B1L_Rad_State3_<br>Postlaunch | Static | float32 array (16 values) | Band 1 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode<br>Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE<br>GROUP: LAMP_RADIANCE_LOW | B1L_Rad_State3_<br>Current | Dynamic | float32 array (16 values) | Band 1 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode<br>Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE<br>GROUP: LAMP_RADIANCE_LOW | B2L_Rad_State1_<br>Prelaunch | Static | float32 array (16 values) | Band 2 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 off; low-gain mode<br>Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE<br>GROUP: LAMP_RADIANCE_LOW | B2L_Rad_State1_<br>Postlaunch | Static | float32 array (16 values) | Band 2 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 off; low-gain mode<br>Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE<br>GROUP: LAMP_RADIANCE_LOW | B2L_Rad_State1_<br>Current | Dynamic | float32 array (16 values) | Band 2 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 off; low-gain mode<br>Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE<br>GROUP: LAMP_RADIANCE_LOW | B2L_Rad_State2_<br>Prelaunch | Static | float32 array (16 values) | Band 2 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode<br>Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE<br>GROUP: LAMP_RADIANCE_LOW | B2L_Rad_State2_<br>Postlaunch | Static | float32 array (16 values) | Band 2 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode<br>Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE<br>GROUP: LAMP_RADIANCE_LOW | B2L_Rad_State2_<br>Current | Dynamic | float32 array (16 values) | Band 2 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode<br>Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE<br>GROUP: LAMP_RADIANCE_LOW | B2L_Rad_State3_<br>Prelaunch | Static | float32 array (16 values) | Band 2 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode<br>Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE<br>GROUP: LAMP_RADIANCE_LOW | B2L_Rad_State3_<br>Postlaunch | Static | float32 array (16 values) | Band 2 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode<br>Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE<br>GROUP: LAMP_RADIANCE_LOW | B2L_Rad_State3_<br>Current | Dynamic | float32 array (16 values) | Band 2 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode<br>Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE<br>GROUP: LAMP_RADIANCE_LOW | B3L_Rad_State1_<br>Prelaunch | Static | float32 array (16 values) | Band 3 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; low-gain mode<br>Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE<br>GROUP: LAMP_RADIANCE_LOW | B3L_Rad_State1_<br>Postlaunch | Static | float32 array (16 values) | Band 3 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; low-gain mode<br>Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE<br>GROUP: LAMP_RADIANCE_LOW | B3L_Rad_State1_<br>Current | Dynamic | float32 array (16 values) | Band 3 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; low-gain mode<br>Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE<br>GROUP: LAMP_RADIANCE_LOW | B3L_Rad_State2_<br>Prelaunch | Static | float32 array (16 values) | Band 3 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode<br>Valid format: NNN.NNN, where N = 0 to 9 |

IAS-207
Version 6.0

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B3L_Rad_State2_ Postlaunch | Static | float32 array (16 values) | Band 3 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B3L_Rad_State2_ Current | Dynamic | float32 array (16 values) | Band 3 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B3L_Rad_State3_ Prelaunch | Static | float32 array (16 values) | Band 3 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B3L_Rad_State3_ Postlaunch | Static | float32 array (16 values) | Band 3 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B3L_Rad_State3_ Current | Dynamic | float32 array (16 values) | Band 3 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B4L_Rad_State1_ Prelaunch | Static | float32 array (16 values) | Band 4 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B4L_Rad_State1_ Postlaunch | Static | float32 array (16 values) | Band 4 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B4L_Rad_State1_ Current | Dynamic | float32 array (16 values) | Band 4 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B4L_Rad_State2_ Prelaunch | Static | float32 array (16 values) | Band 4 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B4L_Rad_State2_ Postlaunch | Static | float32 array (16 values) | Band 4 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B4L_Rad_State2_ Current | Dynamic | float32 array (16 values) | Band 4 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B4L_Rad_State3_ Prelaunch | Static | float32 array (16 values) | Band 4 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B4L_Rad_State3_ Postlaunch | Static | float32 array (16 values) | Band 4 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B4L_Rad_State3_ Current | Dynamic | float32 array (16 values) | Band 4 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B5L_Rad_State1_ Prelaunch | Static | float32 array (16 values) | Band 5 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B5L_Rad_State1_Postlaunch | Static | float32 array (16 values) | Band 5 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B5L_Rad_State1_Current | Dynamic | float32 array (16 values) | Band 5 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B5L_Rad_State2_Prelaunch | Static | float32 array (16 values) | Band 5 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B5L_Rad_State2_Postlaunch | Static | float32 array (16 values) | Band 5 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B5L_Rad_State2_Current | Dynamic | float32 array (16 values) | Band 5 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B5L_Rad_State3_Prelaunch | Static | float32 array (16 values) | Band 5 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B5L_Rad_State3_Postlaunch | Static | float32 array (16 values) | Band 5 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B5L_Rad_State3_Current | Dynamic | float32 array (16 values) | Band 5 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B7L_Rad_State1_Prelaunch | Static | float32 array (16 values) | Band 7 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B7L_Rad_State1_Postlaunch | Static | float32 array (16 values) | Band 7 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B7L_Rad_State1_Current | Dynamic | float32 array (16 values) | Band 7 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B7L_Rad_State2_Prelaunch | Static | float32 array (16 values) | Band 7 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B7L_Rad_State2_Postlaunch | Static | float32 array (16 values) | Band 7 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B7L_Rad_State2_Current | Dynamic | float32 array (16 values) | Band 7 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B7L_Rad_State3_Prelaunch | Static | float32 array (16 values) | Band 7 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B7L_Rad_State3_ Postlaunch | Static | float32 array (16 values) | Band 7 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B7L_Rad_State3_ Current | Dynamic | float32 array (16 values) | Band 7 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B8L_Rad_State1_ Prelaunch | Static | float32 array (32 values) | Band 8 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B8L_Rad_State1_ Postlaunch | Static | float32 array (32 values) | Band 8 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B8L_Rad_State1_ Current | Dynamic | float32 array (32 values) | Band 8 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B8L_Rad_State2_ Prelaunch | Static | float32 array (32 values) | Band 8 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B8L_Rad_State2_ Postlaunch | Static | float32 array (32 values) | Band 8 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B8L_Rad_State2_ Current | Dynamic | float32 array (32 values) | Band 8 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B8L_Rad_State3_ Prelaunch | Static | float32 array (32 values) | Band 8 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B8L_Rad_State3_ Postlaunch | Static | float32 array (32 values) | Band 8 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_LOW | B8L_Rad_State3_ Current | Dynamic | float32 array (32 values) | Band 8 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; low-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B1H_Rad_State1_ Prelaunch | Static | float32 array (16 values) | Band 1 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 off; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B1H_Rad_State1_ Postlaunch | Static | float32 array (16 values) | Band 1 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 off; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B1H_Rad_State1_ Current | Dynamic | float32 array (16 values) | Band 1 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 off; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B1H_Rad_State2_ Prelaunch | Static | float32 array (16 values) | Band 1 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B1H_Rad_State2_ Postlaunch | Static | float32 array (16 values) | Band 1 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B1H_Rad_State2_ Current | Dynamic | float32 array (16 values) | Band 1 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B1H_Rad_State3_ Prelaunch | Static | float32 array (16 values) | Band 1 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B1H_Rad_State3_ Postlaunch | Static | float32 array (16 values) | Band 1 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B1H_Rad_State3_ Current | Dynamic | float32 array (16 values) | Band 1 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B2H_Rad_State1_ Prelaunch | Static | float32 array (16 values) | Band 2 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B2H_Rad_State1_ Postlaunch | Static | float32 array (16 values) | Band 2 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B2H_Rad_State1_ Current | Dynamic | float32 array (16 values) | Band 2 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B2H_Rad_State2_ Prelaunch | Static | float32 array (16 values) | Band 2 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B2H_Rad_State2_ Postlaunch | Static | float32 array (16 values) | Band 2 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B2H_Rad_State2_ Current | Dynamic | float32 array (16 values) | Band 2 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B2H_Rad_State3_ Prelaunch | Static | float32 array (16 values) | Band 2 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B2H_Rad_State3_ Postlaunch | Static | float32 array (16 values) | Band 2 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B2H_Rad_State3_ Current | Dynamic | float32 array (16 values) | Band 2 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B3H_Rad_State1_ Prelaunch | Static | float32 array (16 values) | Band 3 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B3H_Rad_State1_ Postlaunch | Static | float32 array (16 values) | Band 3 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B3H_Rad_State1_ Current | Dynamic | float32 array (16 values) | Band 3 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B3H_Rad_State2_ Prelaunch | Static | float32 array (16 values) | Band 3 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B3H_Rad_State2_ Postlaunch | Static | float32 array (16 values) | Band 3 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B3H_Rad_State2_ Current | Dynamic | float32 array (16 values) | Band 3 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B3H_Rad_State3_ Prelaunch | Static | float32 array (16 values) | Band 3 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B3H_Rad_State3_ Postlaunch | Static | float32 array (16 values) | Band 3 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B3H_Rad_State3_ Current | Dynamic | float32 array (16 values) | Band 3 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B4H_Rad_State1_ Prelaunch | Static | float32 array (16 values) | Band 4 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B4H_Rad_State1_ Postlaunch | Static | float32 array (16 values) | Band 4 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B4H_Rad_State1_ Current | Dynamic | float32 array (16 values) | Band 4 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B4H_Rad_State2_ Prelaunch | Static | float32 array (16 values) | Band 4 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B4H_Rad_State2_ Postlaunch | Static | float32 array (16 values) | Band 4 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B4H_Rad_State2_ Current | Dynamic | float32 array (16 values) | Band 4 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B4H_Rad_State3_ Prelaunch | Static | float32 array (16 values) | Band 4 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B4H_Rad_State3_ Postlaunch | Static | float32 array (16 values) | Band 4 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B4H_Rad_State3_ Current | Dynamic | float32 array (16 values) | Band 4 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B5H_Rad_State1_ Prelaunch | Static | float32 array (16 values) | Band 5 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B5H_Rad_State1_ Postlaunch | Static | float32 array (16 values) | Band 5 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B5H_Rad_State1_ Current | Dynamic | float32 array (16 values) | Band 5 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B5H_Rad_State2_ Prelaunch | Static | float32 array (16 values) | Band 5 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B5H_Rad_State2_ Postlaunch | Static | float32 array (16 values) | Band 5 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B5H_Rad_State2_ Current | Dynamic | float32 array (16 values) | Band 5 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B5H_Rad_State3_ Prelaunch | Static | float32 array (16 values) | Band 5 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B5H_Rad_State3_ Postlaunch | Static | float32 array (16 values) | Band 5 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B5H_Rad_State3_ Current | Dynamic | float32 array (16 values) | Band 5 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B7H_Rad_State1_ Prelaunch | Static | float32 array (16 values) | Band 7 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B7H_Rad_State1_ Postlaunch | Static | float32 array (16 values) | Band 7 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B7H_Rad_State1_ Current | Dynamic | float32 array (16 values) | Band 7 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B7H_Rad_State2_ Prelaunch | Static | float32 array (16 values) | Band 7 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B7H_Rad_State2_ Postlaunch | Static | float32 array (16 values) | Band 7 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B7H_Rad_State2_ Current | Dynamic | float32 array (16 values) | Band 7 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B7H_Rad_State3_ Prelaunch | Static | float32 array (16 values) | Band 7 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B7H_Rad_State3_ Postlaunch | Static | float32 array (16 values) | Band 7 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B7H_Rad_State3_ Current | Dynamic | float32 array (16 values) | Band 7 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B8H_Rad_State1_ Prelaunch | Static | float32 array (32 values) | Band 8 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B8H_Rad_State1_ Postlaunch | Static | float32 array (32 values) | Band 8 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B8H_Rad_State1_ Current | Dynamic | float32 array (32 values) | Band 8 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 1 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B8H_Rad_State2_ Prelaunch | Static | float32 array (32 values) | Band 8 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B8H_Rad_State2_ Postlaunch | Static | float32 array (32 values) | Band 8 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B8H_Rad_State2_ Current | Dynamic | float32 array (32 values) | Band 8 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 2 - lamp 1 off, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B8H_Rad_State3_ Prelaunch | Static | float32 array (32 values) | Band 8 prelaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B8H_Rad_State3_ Postlaunch | Static | float32 array (32 values) | Band 8 postlaunch internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |
| GROUP: LAMP_RADIANCE GROUP: LAMP_RADIANCE_HIGH | B8H_Rad_State3_ Current | Dynamic | float32 array (32 values) | Band 8 current internal calibrator lamp effective spectral radiance in W/m^2-ster-µm; State 3 - lamp 1 on, lamp 2 on; high-gain mode Valid format: NNN.NNN, where N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: LAMP_REFERENCE | Lmp_Rtemp_ PreLaunch | Static | float32 array (14 values) | Prelaunch internal calibrator lamp radiance reference temperatures in degrees C<br>Valid format: SNNN.NNN, where S = "+" or "-" and N = 0 to 9<br>T1 = Cal shutter flag temp<br>T2 = Backup shutter flag temp<br>T3 = Silicon focal plane array temp<br>T4 = Cold focal plane monitor temp<br>T5 = Cal lamp housing temp<br>T6 = Scan line corrector temp<br>T7 = Cal shutter hub temp<br>T8 = Ambient pre-amp temp (high)<br>T9 = Ambient pre-amp temp (low)<br>T10 = Cold pre-amp temp (B7)<br>T11 = Post-amp temp (B4)<br>T12 = Primary mirror amp temp<br>T13 = Secondary mirror temp<br>T14 = Pan band post-amp temp |
| GROUP: LAMP_REFERENCE | Lmp_Rtemp_ Postlaunch | Static | float32 array (14 values) | Postlaunch internal calibrator lamp radiance reference temperatures in degrees C<br>Valid format: SNNN.NNN, where S = "+" or "-" and N = 0 to 9<br>Descriptions of T1 through T14 are the same as above |
| GROUP: LAMP_REFERENCE | Lmp_Rtemp_Current | Dynamic | float32 array (14 values) | Current internal calibrator lamp radiance reference temperatures in degrees C<br>Valid format: SNNN.NNN, where S = "+" or "-" and N = 0 to 9<br>Descriptions of T1 through T14 are the same as above |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B1L_Coefficients_ Detector1 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, low gain, detector 1<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B1L_Coefficients_ Detector2 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, low gain, detector 2<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B1L_Coefficients_ Detector3 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, low gain, detector 3<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B1L_Coefficients_ Detector4 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, low gain, detector 4<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B1L_Coefficients_ Detector5 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, low gain, detector 5<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B1L_Coefficients_ Detector6 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, low gain, detector 6<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B1L_Coefficients_ Detector7 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, low gain, detector 7<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B1L_Coefficients_ Detector8 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, low gain, detector 8 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B1L_Coefficients_ Detector9 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, low gain, detector 9 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B1L_Coefficients_ Detector10 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, low gain, detector 10 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B1L_Coefficients_ Detector11 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, low gain, detector 11 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B1L_Coefficients_ Detector12 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, low gain, detector 12 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B1L_Coefficients_ Detector13 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, low gain, detector 13 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B1L_Coefficients_ Detector14 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, low gain, detector 14 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B1L_Coefficients_ Detector15 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, low gain, detector 15 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B1L_Coefficients_ Detector16 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, low gain, detector 16 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B2L_Coefficients_ Detector1 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, low gain, detector 1 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B2L_Coefficients_ Detector2 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, low gain, detector 2 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B2L_Coefficients_ Detector3 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, low gain, detector 3 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B2L_Coefficients_ Detector4 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, low gain, detector 4 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B2L_Coefficients_ Detector5 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, low gain, detector 5 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B2L_Coefficients_ Detector6 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, low gain, detector 6 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B2L_Coefficients_ Detector7 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, low gain, detector 7 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B2L_Coefficients_ Detector8 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, low gain, detector 8 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B2L_Coefficients_ Detector9 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, low gain, detector 9 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B2L_Coefficients_ Detector10 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, low gain, detector 10 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B2L_Coefficients_ Detector11 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, low gain, detector 11 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B2L_Coefficients_ Detector12 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, low gain, detector 12 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B2L_Coefficients_ Detector13 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, low gain, detector 13 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B2L_Coefficients_ Detector14 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, low gain, detector 14 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B2L_Coefficients_ Detector15 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, low gain, detector 15 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B2L_Coefficients_ Detector16 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, low gain, detector 16 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B3L_Coefficients_ Detector1 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, low gain, detector 1 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B3L_Coefficients_ Detector2 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, low gain, detector 2 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B3L_Coefficients_ Detector3 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, low gain, detector 3 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B3L_Coefficients_ Detector4 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, low gain, detector 4 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B3L_Coefficients_ Detector5 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, low gain, detector 5 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |

IAS-207
Version 6.0

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B3L_Coefficients_ Detector6 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, low gain, detector 6 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B3L_Coefficients_ Detector7 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, low gain, detector 7 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B3L_Coefficients_ Detector8 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, low gain, detector 8 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B3L_Coefficients_ Detector9 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, low gain, detector 9 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B3L_Coefficients_ Detector10 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, low gain, detector 10 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B3L_Coefficients_ Detector11 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, low gain, detector 11 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B3L_Coefficients_ Detector12 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, low gain, detector 12 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B3L_Coefficients_ Detector13 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, low gain, detector 13 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B3L_Coefficients_ Detector14 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, low gain, detector 14 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B3L_Coefficients_ Detector15 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, low gain, detector 15 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B3L_Coefficients_ Detector16 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, low gain, detector 16 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B4L_Coefficients_ Detector1 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, low gain, detector 1 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B4L_Coefficients_ Detector2 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, low gain, detector 2 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B4L_Coefficients_ Detector3 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, low gain, detector 3 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B4L_Coefficients_ Detector4 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, low gain, detector 4 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B4L_Coefficients_Detector5 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, low gain, detector 5 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B4L_Coefficients_Detector6 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, low gain, detector 6 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B4L_Coefficients_Detector7 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, low gain, detector 7 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B4L_Coefficients_Detector8 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, low gain, detector 8 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B4L_Coefficients_Detector9 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, low gain, detector 9 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B4L_Coefficients_Detector10 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, low gain, detector 10 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B4L_Coefficients_Detector11 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, low gain, detector 11 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B4L_Coefficients_Detector12 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, low gain, detector 12 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B4L_Coefficients_Detector13 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, low gain, detector 13 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B4L_Coefficients_Detector14 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, low gain, detector 14 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B4L_Coefficients_Detector15 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, low gain, detector 15 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B4L_Coefficients_Detector16 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, low gain, detector 16 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B5L_Coefficients_Detector1 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, low gain, detector 1 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B5L_Coefficients_Detector2 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, low gain, detector 2 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B5L_Coefficients_Detector3 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, low gain, detector 3 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B5L_Coefficients_ Detector4 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, low gain, detector 4 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B5L_Coefficients_ Detector5 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, low gain, detector 5 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B5L_Coefficients_ Detector6 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, low gain, detector 6 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B5L_Coefficients_ Detector7 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, low gain, detector 7 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B5L_Coefficients_ Detector8 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, low gain, detector 8 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B5L_Coefficients_ Detector9 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, low gain, detector 9 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B5L_Coefficients_ Detector10 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, low gain, detector 10 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B5L_Coefficients_ Detector11 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, low gain, detector 11 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B5L_Coefficients_ Detector12 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, low gain, detector 12 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B5L_Coefficients_ Detector13 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, low gain, detector 13 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B5L_Coefficients_ Detector14 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, low gain, detector 14 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B5L_Coefficients_ Detector15 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, low gain, detector 15 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B5L_Coefficients_ Detector16 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, low gain, detector 16 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B7L_Coefficients_ Detector1 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, low gain, detector 1 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B7L_Coefficients_ Detector2 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, low gain, detector 2 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B7L_Coefficients_Detector3 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, low gain, detector 3 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B7L_Coefficients_Detector4 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, low gain, detector 4 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B7L_Coefficients_Detector5 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, low gain, detector 5 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B7L_Coefficients_Detector6 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, low gain, detector 6 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B7L_Coefficients_Detector7 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, low gain, detector 7 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B7L_Coefficients_Detector8 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, low gain, detector 8 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B7L_Coefficients_Detector9 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, low gain, detector 9 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B7L_Coefficients_Detector10 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, low gain, detector 10 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B7L_Coefficients_Detector11 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, low gain, detector 11 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B7L_Coefficients_Detector12 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, low gain, detector 12 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B7L_Coefficients_Detector13 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, low gain, detector 13 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B7L_Coefficients_Detector14 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, low gain, detector 14 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B7L_Coefficients_Detector15 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, low gain, detector 15 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B7L_Coefficients_Detector16 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, low gain, detector 16 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_Detector1 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 1 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector2 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 2 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector3 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 3 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector4 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 4 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector5 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 5 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients _Detector6 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 6 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector7 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 7 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector8 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 8 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector9 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 9 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector10 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 10 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector11 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 11 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector12 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 12 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector13 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 13 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector14 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 14 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector15 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 15 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector16 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 16 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector17 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 17 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector18 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 18 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector19 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 19 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector20 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 20 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector21 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 21 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector22 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 22 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector23 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 23 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector24 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 24 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector25 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 25 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector26 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 26 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector27 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 27 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector28 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 28 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector29 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 29 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector30 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 30 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector31 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 31 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_LOW | B8L_Coefficients_ Detector32 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, low gain, detector 32 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B1H_Coefficients_ Detector1 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, high gain, detector 1 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B1H_Coefficients_ Detector2 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, high gain, detector 2 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B1H_Coefficients_ Detector3 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, high gain, detector 3 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B1H_Coefficients_ Detector4 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, high gain, detector 4 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B1H_Coefficients_ Detector5 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, high gain, detector 5 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B1H_Coefficients_ Detector6 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, high gain, detector 6 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B1H_Coefficients_ Detector7 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, high gain, detector 7 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B1H_Coefficients_ Detector8 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, high gain, detector 8 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B1H_Coefficients_ Detector9 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, high gain, detector 9 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B1H_Coefficients_ Detector10 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, high gain, detector 10 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B1H_Coefficients_ Detector11 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, high gain, detector 11 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B1H_Coefficients_ Detector12 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, high gain, detector 12 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B1H_Coefficients_ Detector13 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, high gain, detector 13 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B1H_Coefficients_ Detector14 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, high gain, detector 14 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B1H_Coefficients_Detector15 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, high gain, detector 15 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B1H_Coefficients_Detector16 | Dynamic | float32 array (18 values) | IC coefficients for Band 1, high gain, detector 16 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B2H_Coefficients_Detector1 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, high gain, detector 1 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B2H_Coefficients_Detector2 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, high gain, detector 2 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B2H_Coefficients_Detector3 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, high gain, detector 3 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B2H_Coefficients_Detector4 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, high gain, detector 4 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B2H_Coefficients_Detector5 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, high gain, detector 5 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B2H_Coefficients_Detector6 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, high gain, detector 6 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B2H_Coefficients_Detector7 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, high gain, detector 7 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B2H_Coefficients_Detector8 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, high gain, detector 8 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B2H_Coefficients_Detector9 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, high gain, detector 9 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B2H_Coefficients_Detector10 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, high gain, detector 10 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B2H_Coefficients_Detector11 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, high gain, detector 11 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B2H_Coefficients_Detector12 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, high gain, detector 12 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B2H_Coefficients_Detector13 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, high gain, detector 13 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B2H_Coefficients_Detector14 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, high gain, detector 14 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B2H_Coefficients_Detector15 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, high gain, detector 15 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B2H_Coefficients_Detector16 | Dynamic | float32 array (18 values) | IC coefficients for Band 2, high gain, detector 16 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B3H_Coefficients_Detector1 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, high gain, detector 1 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B3H_Coefficients_Detector2 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, high gain, detector 2 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B3H_Coefficients_Detector3 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, high gain, detector 3 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B3H_Coefficients_Detector4 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, high gain, detector 4 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B3H_Coefficients_Detector5 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, high gain, detector 5 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B3H_Coefficients_Detector6 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, high gain, detector 6 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B3H_Coefficients_Detector7 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, high gain, detector 7 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B3H_Coefficients_Detector8 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, high gain, detector 8 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B3H_Coefficients_Detector9 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, high gain, detector 9 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B3H_Coefficients_Detector10 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, high gain, detector 10 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B3H_Coefficients_Detector11 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, high gain, detector 11 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B3H_Coefficients_Detector12 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, high gain, detector 12 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B3H_Coefficients_ Detector13 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, high gain, detector 13 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B3H_Coefficients_ Detector14 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, high gain, detector 14 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B3H_Coefficients_ Detector15 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, high gain, detector 15 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B3H_Coefficients_ Detector16 | Dynamic | float32 array (18 values) | IC coefficients for Band 3, high gain, detector 16 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B4H_Coefficients_ Detector1 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, high gain, detector 1 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B4H_Coefficients_ Detector2 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, high gain, detector 2 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B4H_Coefficients_ Detector3 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, high gain, detector 3 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B4H_Coefficients_ Detector4 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, high gain, detector 4 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B4H_Coefficients_ Detector5 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, high gain, detector 5 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B4H_Coefficients_ Detector6 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, high gain, detector 6 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B4H_Coefficients_ Detector7 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, high gain, detector 7 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B4H_Coefficients_ Detector8 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, high gain, detector 8 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B4H_Coefficients_ Detector9 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, high gain, detector 9 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B4H_Coefficients_ Detector10 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, high gain, detector 10 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B4H_Coefficients_ Detector11 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, high gain, detector 11 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B4H_Coefficients_Detector12 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, high gain, detector 12 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B4H_Coefficients_Detector13 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, high gain, detector 13 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B4H_Coefficients_Detector14 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, high gain, detector 14 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B4H_Coefficients_Detector15 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, high gain, detector 15 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B4H_Coefficients_Detector16 | Dynamic | float32 array (18 values) | IC coefficients for Band 4, high gain, detector 16 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B5H_Coefficients_Detector1 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, high gain, detector 1 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B5H_Coefficients_Detector2 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, high gain, detector 2 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B5H_Coefficients_Detector3 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, high gain, detector 3 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B5H_Coefficients_Detector4 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, high gain, detector 4 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B5H_Coefficients_Detector5 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, high gain, detector 5 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B5H_Coefficients_Detector6 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, high gain, detector 6 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B5H_Coefficients_Detector7 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, high gain, detector 7 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B5H_Coefficients_Detector8 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, high gain, detector 8 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B5H_Coefficients_Detector9 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, high gain, detector 9 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B5H_Coefficients_Detector10 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, high gain, detector 10 Valid format: SNNN.NNNNNNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B5H_Coefficients_ Detector11 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, high gain, detector 11 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B5H_Coefficients_ Detector12 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, high gain, detector 12 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B5H_Coefficients_ Detector13 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, high gain, detector 13 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B5H_Coefficients_ Detector14 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, high gain, detector 14 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B5H_Coefficients_ Detector15 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, high gain, detector 15 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B5H_Coefficients_ Detector16 | Dynamic | float32 array (18 values) | IC coefficients for Band 5, high gain, detector 16 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B7H_Coefficients_ Detector1 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, high gain, detector 1 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B7H_Coefficients_ Detector2 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, high gain, detector 2 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B7H_Coefficients_ Detector3 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, high gain, detector 3 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B7H_Coefficients_ Detector4 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, high gain, detector 4 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B7H_Coefficients_ Detector5 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, high gain, detector 5 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B7H_Coefficients_ Detector6 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, high gain, detector 6 Valid format: SNNN.NNNNNNN, where = + or - and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B7H_Coefficients_ Detector7 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, high gain, detector 7 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B7H_Coefficients_ Detector8 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, high gain, detector 8 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B7H_Coefficients_ Detector9 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, high gain, detector 9 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B7H_Coefficients_ Detector10 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, high gain, detector 10 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B7H_Coefficients_ Detector11 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, high gain, detector 11 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B7H_Coefficients_ Detector12 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, high gain, detector 12 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B7H_Coefficients_ Detector13 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, high gain, detector 13 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B7H_Coefficients_ Detector14 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, high gain, detector 14 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B7H_Coefficients_ Detector15 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, high gain, detector 15 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B7H_Coefficients_ Detector16 | Dynamic | float32 array (18 values) | IC coefficients for Band 7, high gain, detector 16 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients _Detector1 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 1 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector2 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 2 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector3 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 3 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector4 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 4 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector5 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 5 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector6 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 6 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector7 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 7 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector8 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 8 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector9 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 9 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector10 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 10 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector11 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 11 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector12 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 12 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector13 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 13 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector14 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 14 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector15 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 15 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector16 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 16 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector17 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 17 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector18 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 18 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector19 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 19 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector20 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 20 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector21 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 21 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector22 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 22 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_ Detector23 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 23 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_Detector24 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 24 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_Detector25 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 25 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_Detector26 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 26 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_Detector27 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 27 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_Detector28 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 28 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_Detector29 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 29 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_Detector30 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 30 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_Detector31 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 31 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: REFLECTIVE_IC_COEFFS GROUP: REFLECT_IC_COEFFS_HIGH | B8H_Coefficients_Detector32 | Dynamic | float32 array (18 values) | IC coefficients for Band 8, high gain, detector 32 Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: B6_VIEW_COEFFS | B6_View_Coefficients_Detector1 | Static | float32 array (15 values) | View factor coefficients for Band 6, detector 1 Valid format: SNNN.NNNNNNNNN, where S = "+" or "-" and N = 0 to 9 a1 = Scan line corrector view factor a2 = Central baffles (heater) a3 = Secondary mirror and mask view factor a4 = Primary mirror and mask view factor a5 = Scan mirror view factor a6 = Black body (isolated) view factor a7 = Black body (control) view factor a8 = Cold focal plane control view factor a9 = Cold focal plane monitor view factor a10 = Baffle (tube) view factor a11 = Baffle (support) view factor a12 = Telescope housing view factor frb = Integrated instrument view factor Vbb = Blocked aperture black body view factor Vsh = Blocked aperture shutter view factor |
| GROUP: B6_VIEW_COEFFS | B6_View_Coefficients_Detector2 | Static | float32 array (15 values) | View factor coefficients for Band 6, detector 2 Valid format: SNNN.NNNNNNNNN, where S = "+" or "-" and N = 0 to 9 Descriptions of the 15 coefficients are the same as above |

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: B6_VIEW_COEFFS | B6_View_Coefficients_ Detector3 | Static | float32 array ( 15 values) | View factor coefficients for Band 6, detector 3<br>Valid format: SNNN.NNNNNNNNN, where S = "+" or "-" and N = 0 to 9<br>Descriptions of the 15 coefficients are the same as above |
| GROUP: B6_VIEW_COEFFS | B6_View_Coefficients_ Detector4 | Static | float32 array ( 15 values) | View factor coefficients for Band 6, detector 4<br>Valid format: SNNN.NNNNNNNNN, where S = "+" or "-" and N = 0 to 9<br>Descriptions of the 15 coefficients are the same as above |
| GROUP: B6_VIEW_COEFFS | B6_View_Coefficients_ Detector5 | Static | float32 array ( 15 values) | View factor coefficients for Band 6, detector 5<br>Valid format: SNNN.NNNNNNNNN, where S = "+" or "-" and N = 0 to 9<br>Descriptions of the 15 coefficients are the same as above |
| GROUP: B6_VIEW_COEFFS | B6_View_Coefficients_ Detector6 | Static | float32 array ( 15 values) | View factor coefficients for Band 6, detector 6<br>Valid format: SNNN.NNNNNNNNN, where S = "+" or "-" and N = 0 to 9<br>Descriptions of the 15 coefficients are the same as above |
| GROUP: B6_VIEW_COEFFS | B6_View_Coefficients_ Detector7 | Static | float32 array ( 15 values) | View factor coefficients for Band 6, detector 7<br>Valid format: SNNN.NNNNNNNNN, where S = "+" or "-" and N = 0 to 9<br>Descriptions of the 15 coefficients are the same as above |
| GROUP: B6_VIEW_COEFFS | B6_View_Coefficients_ Detector8 | Static | float32 array ( 15 values) | View factor coefficients for Band 6, detector 8<br>Valid format: SNNN.NNNNNNNNN, where S = "+" or "-" and N = 0 to 9<br>Descriptions of the 15 coefficients are the same as above |
| GROUP: B6_TEMP_MODEL_COEFFS | B6_Temp_Model_Parm | Dynamic | float32 array (6 values) | Coefficients used to calculate scan mirror temperature where (a1) = Scan mirror/secondary mirror adjustment factor, (a2) = Average secondary mirror temperature, and (a3) - (a6) = reserved<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and<br>SNNN.NNNNNNN = +1.0178 (a1)<br>SNNN.NNNNNNN = +0.0 (a2)<br>SNNN.NNNNNNN = +0.0 (a3)<br>SNNN.NNNNNNN = +0.0 (a4)<br>SNNN.NNNNNNN = +0.0 (a5)<br>SNNN.NNNNNNN = +0.0 (a6) |
| GROUP: THERMISTOR_COEFFS | Black_Body_Isolated_ Temp | Static | float32 array (6 values) | Calibration coefficients for raw data<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Black_Body_Control_ Temp | Static | float32 array (6 values) | Calibration coefficients for raw data<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Cold_FP_Control_ Temp | Static | float32 array (6 values) | Calibration coefficients for raw data<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Cold_FP_Monitor_ Temp | Static | float32 array (6 values) | Calibration coefficients for raw data<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Cal_Shutter_Flag_ Temp | Static | float32 array (6 values) | Calibration coefficients for raw data<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Backup_Shutter_ Flag_Temp | Static | float32 array (6 values) | Calibration coefficients for raw data<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |

IAS-207
Version 6.0

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: THERMISTOR_COEFFS | Baffle_Heater_Temp | Static | float32 array (6 values) | Calibration coefficients for raw data Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Silicon_FP_Array_ Temp | Static | float32 array (6 values) | Calibration coefficients for raw data Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Primary_Mirror_Temp | Static | float32 array (6 values) | Calibration coefficients for raw data Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Secondary_Mirror_ Temp | Static | float32 array (6 values) | Calibration coefficients for raw data Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Scan_Line_Corrector_ Temp | Static | float32 array (6 values) | Calibration coefficients for raw data Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Baffle3_Tube_Temp | Static | float32 array (6 values) | Calibration coefficients for raw data Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Baffle2_Support_ Temp | Static | float32 array (6 values) | Calibration coefficients for raw data Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Cal_Lamp_Housing_ Temp | Static | float32 array (6 values) | Calibration coefficients for raw data Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Cal_Shutter_Hub_ Temp | Static | float32 array (6 values) | Calibration coefficients for raw data Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Ambient_Preamp_ HighCh_Temp | Static | float32 array (6 values) | Calibration coefficients for raw data Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Ambient_Preamp_ LowCh_Temp | Static | float32 array (6 values) | Calibration coefficients for raw data Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Postamp_Temp_B4 | Static | float32 array (6 values) | Calibration coefficients for raw data Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Cold_Preamp_B7_ Temp | Static | float32 array (6 values) | Calibration coefficients for raw data Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Pan_Band_Postamp_ Temp | Static | float32 array (6 values) | Calibration coefficients for raw data Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Telescope_Housing_ Temp | Static | float32 array (6 values) | Calibration coefficients for raw data Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Primary_Mirror_ Mask_Temp | Static | float32 array (6 values) | Calibration coefficients for raw data Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Secondary_Mirror_ Mask_Temp | Static | float32 array (6 values) | Calibration coefficients for raw data Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Telescope_ Baseplate_Temp | Static | float32 array (6 values) | Calibration coefficients for raw data Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Mem_Heat_Sink_ Power_Supply1_Temp | Static | float32 array (6 values) | Calibration coefficients for raw data Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |

IAS-207
Version 6.0

| Parameter Group | Parameter Name | Value Type | Data Type | Description |
|---|---|---|---|---|
| GROUP: THERMISTOR_COEFFS | Mem_Heat_Sink_ Power_Supply2_Temp | Static | float32 array (6 values) | Calibration coefficients for raw data<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Mux1_Power_Supply_ Temp | Static | float32 array (6 values) | Calibration coefficients for raw data (telemetry value contains the power supply temperature for "active" Mux, which could be either Mux 1 or Mux 2)<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: THERMISTOR_COEFFS | Mux1_Electronics_ Temp | Static | float32 array (6 values) | Calibration coefficients for raw data (telemetry value contains the power supply temperature for "active" Mux, which could be either Mux 1 or Mux 2)<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: LAMP_CURRENTS | Tec_Lamp_i1 | Static | float32 array (2 values) | Calibration coefficients for raw data (telemetry value contains current in mA of primary on-board calibration lamp, telemetry name = TECLAMP1I)<br>Valid format: SNNN.NNNNNNN, where S = "+" or "-" and N = 0 to 9 |
| GROUP: LAMP_CURRENTS | Tec_Lamp_i2 | Static | float32 array (2 values) | Calibration coefficients for raw data (telemetry value contains current in mA of primary on-board calibration lamp, telemetry name = TECLAMP2I)<br>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 to fill erroneous or missing image data minor frames<br>Valid format: NNN, where NNN = (0, 255) (alternating 0, 255's) |

***Table 2-1. Landsat 7 CPF Parameters***

# Section 3    CPF ODL

## 3.1    Introduction

The ODL syntax employs the following conventions:

- Parameter definition is in the form of parameter = value.
- 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 parameters Effective_Date_Begin and Effective_Date_End also do not have quotation marks. ODL recognizes dates if they follow prescribed formats.
- In general for ODL, case is not significant. However, for the CPF, the case is significant for keyword and group names. All group names are in all capital letters and keywords are in mixed case.
- Indentation is not significant but is used for readability.
- The reserved word END concludes the file.
- Most parameter values have been derived during prelaunch instrument and spacecraft testing and analysis. Formats for CPF numerical parameters are accurate; however, negative signs are not explicitly stated. A data dictionary that declares each parameter's data type and value range has been defined.

## 3.2    Sample ETM+ CPF ODL File

The following is a prototype of CPF file that contains valid parameter values for the first calendar quarter of 2007. To present the format structure, the hypothetical bumper mode specific parameters are also included in this example.

```
GROUP = FILE_ATTRIBUTES
  Spacecraft_Name = "Landsat_7"
  Sensor_Name = "Enhanced_Thematic_Mapper_Plus"
  Effective_Date_Begin = 2007-01-01
  Effective_Date_End = 2007-03-31
  CPF_File_Name = "L7CPF20070101_20070331.01"
END_GROUP = FILE_ATTRIBUTES
GROUP = EARTH_CONSTANTS
  Ellipsoid_Name = "WGS84"
  Semi_Major_Axis = 6378137.000
  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 = "09:45"
   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_60 = 8
   Lines_Per_Scan_15 = 32
   Scans_Per_Scene = 375
   Swath_Angle = .26868
   Scan_Rate = 2.21095
   Dwell_Time_30 = 9.6110206
   Dwell_Time_60 = 19.222041
   Dwell_Time_15 = 4.8055103
   IC_Line_Length_30 = 1150
   IC_Line_Length_60 = 575
   IC_Line_Length_15 = 2300
   Scan_Line_Length_30 = 6320
   Scan_Line_Length_60 = 3160
   Scan_Line_Length_15 = 12640
   Filter_Frequency_30 = 52.02
   Filter_Frequency_60 = 26.01
   Filter_Frequency_15 = 115.00
   IFOV_B1234 = 42.5
   IFOV_B57_along_scan = 39.4
   IFOV_B57_across_scan = 42.5
   IFOV_B6 = 85.0
   IFOV_B8_along_scan = 18.5
   IFOV_B8_across_scan = 21.25
   Scan_Period = 143.58
   Scan_Frequency = 6.96476
   Active_Scan_Time = 60743.346
   Turn_Around_Time = 11.055
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 = 74.914
END_GROUP = SPACECRAFT_PARAMETERS
GROUP = MIRROR_PARAMETERS
   Error_Conversion_Factor = 0.18845139
   GROUP = ANGLES_SME1_SAM
     Forward_Along_SME1_SAM = (0.000000E+00,-2.188024E-03,3.507066E-01,-
1.638834E+01,3.070082E+02,-2.016646E+03)
     Forward_Cross_SME1_SAM = (-8.926001E-07,2.945449E-04,-2.799967E-02,1.024417E+00,-
1.579172E+01,8.644595E+01)
     Forward_Angle1_SME1_SAM = 67166.9
     Forward_Angle2_SME1_SAM = 67145.9
     Reverse_Along_SME1_SAM = (0.000000E+00,2.717297E-03,-3.610215E-01,1.637412E+01,-
3.045525E+02,1.987221E+03)
     Reverse_Cross_SME1_SAM = (-7.702087E-07,1.318691E-04,-4.507913E-03,-8.416380E-
02,5.421192E+00,-5.563424E+01)
     Reverse_Angle1_SME1_SAM = 67142.8
     Reverse_Angle2_SME1_SAM = 67169.9
   END_GROUP = ANGLES_SME1_SAM
   GROUP = ANGLES_SME1_BUMP
```

```
    Forward_Along_SME1_Bump = (1.177376E-19, -2.713081E-03, 3.605800E-01, -1.618500E+01,
3.001900E+02, -1.965000E+03)
    Forward_Cross_SME1_Bump = (-3.159000E-07, 4.831800E-06, -1.336000E-03, 6.273300E-02, -
1.174500E+00, 7.932400E+00)
    Forward_Angle1_SME1_Bump =
(68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,
68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,6
8302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68
302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,683
02.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,6830
2.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302
.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.
9,68302.9,68302.9,68302.9,68302.9,68302.9,68302.9)
    Forward_Angle2_SME1_Bump =
(69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,
69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,6
9050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69
050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,690
50.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,6905
0.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050
.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.5,69050.
5,69050.5,69050.5,69050.5,69050.5,69050.5)
    Forward_FHSERR_SME1_Bump =
(878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,
878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,8
78,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,87
8,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878)
    Forward_SHSERR_SME1_Bump = (-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-
883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-
883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-
883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-
883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-
883,-883)
    Reverse_Along_SME1_Bump = (-4.065758E-2, 2.074688E-03, -3.345100E-01, 1.567300E+01, -
2.953100E+02, 1.954000E+03)
    Reverse_Cross_SME1_Bump = (-5.611700E-07, -1.018300E-06, -1.553500E-04, 2.048200E-03,
1.075500E-01, -1.450700E+00)
    Reverse_Angle1_SME1_Bump =
(68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,
68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,6
8234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68
234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,682
34.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,6823
4.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234
.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.3,68234.
3,68234.3,68234.3,68234.3,68234.3,68234.3)
    Reverse_Angle2_SME1_Bump =
(68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,
68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,6
8889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68
889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,688
89.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,6888
9.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889
.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.3,68889.
3,68889.3,68889.3,68889.3,68889.3,68889.3)
    Reverse_FHSERR_SME1_Bump =
(785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,
785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,7
85,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,78
5,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785)
    Reverse_SHSERR_SME1_Bump = (-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-
790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-
790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-
790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-
790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-
790,-790)
  END_GROUP = ANGLES_SME1_BUMP
  GROUP = ANGLES_SME2_SAM
    Forward_Along_SME2_SAM = (0.000000E+00, -2.100656E-03, 3.401124E-01, -1.558871E+01,
2.878695E+02, -1.877441E+03)
```

```
      Forward_Cross_SME2_SAM = (-2.374600E-09, -8.188300E-06, 1.072700E-04, -3.646200E-03,
1.456200E-01, -1.486700E+00)
      Forward_Angle1_SME2_SAM = 67162.7
      Forward_Angle2_SME2_SAM = 67162.8
      Reverse_Along_SME2_SAM = (0.000000E+00, 2.746938E-03, -3.415100E-01, 1.534667E+01, -
2.872800E+02, 1.892100E+03)
      Reverse_Cross_SME2_SAM = (-6.351600E-07, 1.258700E-05, -7.787700E-04, 1.767400E-02, -
1.108500E-01, -1.597100E-01)
      Reverse_Angle1_SME2_SAM = 67162.8
      Reverse_Angle2_SME2_SAM = 67162.7
  END_GROUP = ANGLES_SME2_SAM
  GROUP = ANGLES_SME2_BUMP
      Forward_Along_SME2_Bump = (0.000000E+00, -2.463915E-03, 3.546100E-01, -1.609400E+01,
2.987000E+02, -1.956800E+03)
      Forward_Cross_SME2_Bump = (-3.344900E-07, 7.778000E-06, -1.768700E-03, 8.061500E-02, -
1.463400E+00, 9.512300E+00)
      Forward_Angle1_SME2_Bump =
(67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,
67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,6
7162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67
162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,671
62.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,6716
2.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162
.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.
7,67162.7,67162.7,67162.7,67162.7,67162.7)
      Forward_Angle2_SME2_Bump =
(67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,
67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,6
7162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67
162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,671
62.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,6716
2.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162
.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.
8,67162.8,67162.8,67162.8,67162.8,67162.8)
      Forward_FHSERR_SME2_Bump =
(878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,
878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,8
78,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,87
8,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878,878)
      Forward_SHSERR_SME2_Bump = (-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-
883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-
883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-
883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-
883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-883,-
883,-883)
      Reverse_Along_SME2_Bump = (0.000000E+00, 2.234071E-03, -3.347900E-01, 1.554200E+01, -
2.927500E+02, 1.936900E+03)
      Reverse_Cross_SME2_Bump = (-6.024100E-07, 6.736100E-06, -1.153000E-03, 5.158900E-02, -
9.145700E-01, 5.977300E+00)
      Reverse_Angle1_SME2_Bump =
(67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,
67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,6
7162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67
162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,671
62.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,6716
2.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162
.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.8,67162.
8,67162.8,67162.8,67162.8,67162.8,67162.8)
      Reverse_Angle2_SME2_Bump =
(67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,
67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,6
7162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67
162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,671
62.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,6716
2.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162
.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.7,67162.
7,67162.7,67162.7,67162.7,67162.7,67162.7)
      Reverse_FHSERR_SME2_Bump =
(785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,
785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,7
```

85,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,78
5,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785,785)
    Reverse_SHSERR_SME2_Bump = (-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-
790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-
790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-
790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-
790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-790,-
790,-790)
  END_GROUP = ANGLES_SME2_BUMP
END_GROUP = MIRROR_PARAMETERS

GROUP = BUMPER_MODE_PARAMETERS
  SME1_BumperA_Dwell_Time =
(9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.
8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770
.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,977
0.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,97
70.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9
770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,
9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8)
  SME1_BumperA_Pickoff_Time =
(511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,
511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,5
11.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,51
1.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511
.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.
0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0)
  SME1_BumperA_Offset_Time = 10110.0
  SME1_BumperA_Angle = -69000.0
  SME1_BumperB_Dwell_Time =
(9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.
7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801
.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,980
1.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,98
01.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9
801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,
9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7)
  SME1_BumperB_Pickoff_Time =
(439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,
439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,4
39.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,43
9.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439
.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.
6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6)
  SME1_BumperB_Offset_Time = 10110.0
  SME1_BumperB_Angle = 69000.0
  SME2_BumperA_Dwell_Time =
(9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.
8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770
.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,977
0.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,97
70.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9
770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,9770.8,
9770.8,9770.8,9770.8,9770.8,9770.8,9770.8)
  SME2_BumperA_Pickoff_Time =
(511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,
511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,5
11.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,51
1.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511
.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.
0,511.0,511.0,511.0,511.0,511.0,511.0,511.0,511.0)
  SME2_BumperA_Offset_Time = 10110.0
  SME2_BumperA_Angle = -69000.0
  SME2_BumperB_Dwell_Time =
(9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.
7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801
.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,980
1.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,98
01.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9
801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,
9801.7,9801.7,9801.7,9801.7,9801.7,9801.7,9801.7)

```
  SME2_BumperB_Pickoff_Time =
(439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,
439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,4
39.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,43
9.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439
.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.
6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6,439.6)
  SME2_BumperB_Offset_Time = 10110.0
  SME2_BumperB_Angle = 69000.0
END_GROUP = BUMPER_MODE_PARAMETERS

GROUP = SCAN_LINE_CORRECTOR
  Primary_Angular_Velocity = 0.0
  Secondary_Angular_Velocity = 0.0
  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 = 0.0000427
END_GROUP = SCAN_LINE_CORRECTOR
GROUP = FOCAL_PLANE_PARAMETERS
  GROUP = BAND_OFFSETS
    Along_Scan_Band_Offsets = (+3627.944,+2564.567,+1501.002,+438.166,-2577.619,-4072.538,-
1473.263,+4692.000)
    Across_Scan_Band_Offsets = (+1.280,+0.537,-0.270,-1.447,+15.828,+31.762,+15.683,+0.000)
    Forward_Focal_Plane_Offsets = (+25.0,+50.0,+75.0,+100.0,+171.0,+206.0,+145.0,+0.0)
    Reverse_Focal_Plane_Offsets = (-25.0,-50.0,-75.0,-100.0,-171.0,-206.0,-145.0,+0.5)
  END_GROUP = BAND_OFFSETS
  GROUP = DETECTOR_OFFSETS
    Forward_Along_Scan_DO_B1 = (1.250, 1.298, 1.324, 1.250, 1.253, 1.241, 1.254, 1.271, 1.207,
1.260, 1.247, 1.251, 1.234, 1.227, 1.264, 1.243)
    Reverse_Along_Scan_DO_B1 = (1.241, 1.278, 1.320, 1.230, 1.244, 1.224, 1.256, 1.260, 1.223,
1.255, 1.262, 1.252, 1.251, 1.234, 1.287, 1.258)
    Forward_Along_Scan_DO_B2 = (1.260, 1.278, 1.263, 1.276, 1.269, 1.246, 1.233, 1.232, 1.281,
1.229, 1.237, 1.240, 1.228, 1.246, 1.221, 1.267)
    Reverse_Along_Scan_DO_B2 = (1.249, 1.262, 1.257, 1.258, 1.260, 1.230, 1.234, 1.221, 1.294,
1.224, 1.253, 1.242, 1.244, 1.252, 1.244, 1.282)
    Forward_Along_Scan_DO_B3 = (1.252, 1.256, 1.267, 1.225, 1.240, 1.261, 1.236, 1.297, 1.203,
1.222, 1.214, 1.193, 1.218, 1.210, 1.269, 1.221)
    Reverse_Along_Scan_DO_B3 = (1.242, 1.237, 1.260, 1.207, 1.231, 1.244, 1.235, 1.287, 1.213,
1.218, 1.229, 1.195, 1.238, 1.214, 1.294, 1.237)
    Forward_Along_Scan_DO_B4 = (1.263, 1.259, 1.286, 1.268, 1.265, 1.269, 1.257, 1.300, 1.281,
1.265, 1.234, 1.253, 1.232, 1.226, 1.256, 1.301)
    Reverse_Along_Scan_DO_B4 = (1.253, 1.238, 1.278, 1.246, 1.256, 1.251, 1.257, 1.291, 1.292,
1.263, 1.249, 1.259, 1.250, 1.233, 1.281, 1.317)
    Forward_Along_Scan_DO_B5 = (1.163, 1.165, 1.144, 1.137, 1.150, 1.120, 1.109, 1.109, 1.100,
1.095, 1.067, 1.069, 1.058, 1.053, 1.058, 1.027)
    Reverse_Along_Scan_DO_B5 = (1.047, 1.037, 1.045, 1.068, 1.078, 1.077, 1.075, 1.103, 1.108,
1.125, 1.107, 1.133, 1.133, 1.151, 1.168, 1.166)
    Forward_Along_Scan_DO_B6 = (1.904, 2.058, 1.890, 2.055, 1.899, 1.946, 1.820, 1.924)
    Reverse_Along_Scan_DO_B6 = (1.952, 1.899, 1.951, 1.924, 1.964, 1.907, 1.974, 1.924)
    Forward_Along_Scan_DO_B7 = (1.202, 1.190, 1.217, 1.165, 1.185, 1.116, 1.168, 1.117, 1.121,
1.092, 1.110, 1.091, 1.079, 1.058, 1.076, 1.042)
    Reverse_Along_Scan_DO_B7 = (1.034, 1.064, 1.098, 1.074, 1.104, 1.064, 1.135, 1.113, 1.133,
1.133, 1.163, 1.168, 1.170, 1.171, 1.206, 1.199)
    Forward_Along_Scan_DO_B8 = (0.511, 0.508, 0.505, 0.514, 0.513, 0.523, 0.521, 0.511, 0.509,
0.499, 0.517, 0.513, 0.508, 0.508, 0.516, 0.512, 0.507, 0.523, 0.522, 0.541, 0.499, 0.527, 0.510,
0.528, 0.518, 0.519, 0.515, 0.518, 0.514, 0.521, 0.499, 0.523)
    Reverse_Along_Scan_DO_B8 = (0.511, 0.508, 0.505, 0.514, 0.513, 0.523, 0.521, 0.511, 0.509,
0.499, 0.517, 0.513, 0.507, 0.508, 0.516, 0.512, 0.514, 0.489, 0.525, 0.504, 0.515, 0.497, 0.522,
0.505, 0.535, 0.497, 0.539, 0.505, 0.544, 0.516, 0.537, 0.523)
    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, 0.000, 0.000, 0.000, 0.000, 0.000)
      Reverse_Across_Scan_DO_B6 = (0.000, 0.000, 0.000, 0.000, 0.000, 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)
      Forward_Across_Scan_DO_B8 = (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, 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_B8 = (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, 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 = (31.0,56.0,81.0,106.0,177.0,101.0,151.0,14.0)
      Forward_Odd_Detector_Shift = (33.0,58.0,83.0,108.0,179.0,104.0,153.0,18.0)
      Reverse_Even_Detector_Shift = (27.0,52.0,77.0,102.0,173.0,105.0,147.0,4.0)
      Reverse_Odd_Detector_Shift = (30.0,55.0,80.0,105.0,176.0,107.0,150.0,8.0)
    END_GROUP = ODD_EVEN_OFFSETS
END_GROUP = FOCAL_PLANE_PARAMETERS
GROUP = ATTITUDE_PARAMETERS
  Gyro_To_Attitude_Matrix = (9.99999900E-01, -3.68543600E-04, 2.43062700E-05, 2.43207600E-05, -
2.22997000E-04, -1.00000000E+00, 3.68785400E-04, 9.99999900E-01, -2.22941100E-04)
  ADSA_To_ETM_Matrix = (9.99999800E-01, 1.65108100E-04, 6.51893000E-04, 6.78739000E-05,
9.39659800E-01, -3.42110300E-01, -6.69042800E-04, 3.42110300E-01, 9.39659600E-01)
  Attitude_To_ETM_Matrix = (9.99999845E-01, 1.18363752E-04, 5.43986578E-04, -1.18213574E-04,
9.99999955E-01, -2.76092898E-04, -5.44019232E-04, 2.76028548E-04, 9.99999814E-01)
  Spacecraft_Roll_Bias = 0.00000000E+00
  Spacecraft_Pitch_Bias = 0.00000000E+00
  Spacecraft_Yaw_Bias = 0.00000000E+00
  IMU_Drift_Bias_XA = -2.23500000E-06
  IMU_Drift_Bias_YA = -2.23500000E-06
  IMU_Drift_Bias_ZA = 1.68230000E-06
  IMU_Drift_Bias_XB = 1.86665000E-06
  IMU_Drift_Bias_YB = -6.35100000E-07
  IMU_Drift_Bias_ZB = 4.84810000E-08
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 = 3.3113091
    Zeta = 0.66882924
    Tau = -1.6086176E-2
    P = -4.1138195E-3
    Ak = 1.0103061
  END_GROUP = IMU
  GROUP = ADS
    ADS_num =
(0.0,0.0,0.0,+1.1381628E3,+2.0062211E2,+1.00000E0,0.0,0.0,0.0,1.1381628E3,+2.0062211E2,+1.00000E0
,0.0,0.0,0.0,+9.2111049E2,+1.9766902E2,+1.00000E0)
    ADS_den =
(+1.4195461E5,+1.6961111E5,+4.7877303E4,+4.8108434E3,+2.3739131E2,+1.0000000E0,+1.4168390E5,+1.83
26650E5,+5.2674623E4,+5.1999651E3,+2.3909029E2,+1.0000000E0,+1.1459413E5,+1.4727717E5,+4.7786443E
4,+4.3224093E3,+2.3570742E2,+1.0000000E0)
    ADS_num_temp = (0.0,0.0,0.0,+6.4416984E1,-1.3578067E0,-5.0789831E-
3,0.0,0.0,0.0,+2.0618135E2,+4.7466808E0,-2.9005228E-
3,0.0,0.0,0.0,+9.1603744E1,+2.0285055E0,+4.0783070E-2)
```

ADS_den_temp = (+7.6388956E3,+8.7276441E3,+7.5038775E2,+3.2855210E0,-2.1966002E0,-4.6355589E-3,+1.9779418E4,+3.3575148E4,+4.6478372E3,+2.6281609E2,+4.3279161E0,-2.7584826E-3,+9.9464208E3,+1.3229420E4,+1.8093952E3,+9.2350092E1,+2.9068940E0,+4.2219584E-2)
  END_GROUP = ADS
  GROUP = PREFILTER
    ADSPre_W = (0.000670695,0.000427279,0.000667499,0.000946530,0.001221428)
    ADSPre_H = (-0.0748,0.0133,0.7994,0.1824,1.00157)
    ADSPre_T = (0.0010191,0.000015,0.0,0.0,0.0)
  END_GROUP = PREFILTER
END_GROUP = TRANSFER_FUNCTION
GROUP = UT1_TIME_PARAMETERS
  UT1_Year =
(2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2006,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007,2007)
  UT1_Month =
("Nov","Nov","Nov","Nov","Nov","Nov","Nov","Nov","Nov","Nov","Nov","Nov","Nov","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Dec","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Jan","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Feb","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Mar","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","Apr","May","May","May","May","May","May","May","May","May","May","May","May","May","May","May")
  UT1_Day =
(18,19,20,21,22,23,24,25,26,27,28,29,30,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16)
  UT1_Modified_Julian =
(54057,54058,54059,54060,54061,54062,54063,54064,54065,54066,54067,54068,54069,54070,54071,54072,54073,54074,54075,54076,54077,54078,54079,54080,54081,54082,54083,54084,54085,54086,54087,54088,54089,54090,54091,54092,54093,54094,54095,54096,54097,54098,54099,54100,54101,54102,54103,54104,54105,54106,54107,54108,54109,54110,54111,54112,54113,54114,54115,54116,54117,54118,54119,54120,54121,54122,54123,54124,54125,54126,54127,54128,54129,54130,54131,54132,54133,54134,54135,54136,54137,54138,54139,54140,54141,54142,54143,54144,54145,54146,54147,54148,54149,54150,54151,54152,54153,54154,54155,54156,54157,54158,54159,54160,54161,54162,54163,54164,54165,54166,54167,54168,54169,54170,54171,54172,54173,54174,54175,54176,54177,54178,54179,54180,54181,54182,54183,54184,54185,54186,54187,54188,54189,54190,54191,54192,54193,54194,54195,54196,54197,54198,54199,54200,54201,54202,54203,54204,54205,54206,54207,54208,54209,54210,54211,54212,54213,54214,54215,54216,54217,54218,54219,54220,54221,54222,54223,54224,54225,54226,54227,54228,54229,54230,54231,54232,54233,54234,54235,54236)
  UT1_X = (-0.02126,-0.02269,-0.02414,-0.02561,-0.02684,-0.02808,-0.02957,-0.03118,-0.03286,-0.03446,-0.03572,-0.03663,-0.03742,-0.03814,-0.03848,-0.03856,-0.03863,-0.03876,-0.03921,-0.04011,-0.04122,-0.04235,-0.04350,-0.04465,-0.04579,-0.04691,-0.04802,-0.04910,-0.05016,-0.05120,-0.05220,-0.05318,-0.05413,-0.05504,-0.05593,-0.05679,-0.05761,-0.05840,-0.05917,-0.05989,-0.06059,-0.06125,-0.06187,-0.06247,-0.06303,-0.06355,-0.06404,-0.06449,-0.06492,-0.06530,-0.06564,-0.06595,-0.06623,-0.06646,-0.06666,-0.06681,-0.06693,-0.06702,-0.06706,-0.06706,-0.06703,-0.06696,-0.06684,-0.06669,-0.06650,-0.06626,-0.06599,-0.06568,-0.06532,-0.06493,-0.06450,-0.06402,-0.06351,-0.06295,-0.06235,-0.06171,-0.06104,-0.06032,-0.05956,-0.05876,-0.05792,-0.05704,-0.05612,-0.05516,-0.05416,-0.05312,-0.05204,-0.05093,-0.04977,-0.04858,-0.04734,-0.04607,-0.04476,-0.04342,-0.04203,-0.04061,-0.03916,-0.03767,-0.03614,-0.03458,-0.03298,-0.03134,-0.02968,-0.02798,-0.02625,-0.02448,-0.02269,-0.02086,-0.01900,-0.01711,-0.01519,-0.01324,-0.01126,-0.00926,-0.00723,-0.00517,-0.00308,-0.00097,0.00117,0.00333,0.00551,0.00772,0.00995,0.01220,0.01447,0.01676,0.01907,0.02140,0.02375,0.02612,0.02850,0.03090,0.03331,0.03574,0.03818,0.04063,0.04310,0.04558,0.04807,0.05057,0.05308,0.05559,0.05812,0.06065,0.06318,0.06573,0.06827,0.07082,0.07338,0.07594,0.07849,0.08105,0.08361,0.08617,0.08873,0.09128,0.09383,0.09638,0.09892,0.10146,0.10400,0.10652,0.10904,0.11155,0.11406,0.11

655,0.11903,0.12151,0.12397,0.12642,0.12885,0.13128,0.13369,0.13608,0.13846,0.14083,0.14317,0.145
50,0.14782,0.15011)
  UT1_Y =
(0.29018,0.29078,0.29127,0.29199,0.29307,0.29435,0.29579,0.29730,0.29881,0.30016,0.30114,0.30189,
0.30287,0.30385,0.30490,0.30642,0.30828,0.31009,0.31195,0.31392,0.31583,0.31768,0.31948,0.32125,0
.32298,0.32470,0.32640,0.32810,0.32979,0.33147,0.33316,0.33486,0.33656,0.33827,0.33998,0.34171,0.
34345,0.34520,0.34696,0.34873,0.35051,0.35230,0.35410,0.35591,0.35774,0.35957,0.36141,0.36326,0.3
6512,0.36699,0.36886,0.37074,0.37263,0.37453,0.37643,0.37833,0.38024,0.38216,0.38407,0.38599,0.38
792,0.38984,0.39176,0.39369,0.39562,0.39754,0.39947,0.40139,0.40331,0.40523,0.40711,0.40903,0.410
94,0.41284,0.41474,0.41664,0.41852,0.42040,0.42228,0.42414,0.42600,0.42784,0.42968,0.43150,0.4333
2,0.43512,0.43691,0.43868,0.44045,0.44220,0.44393,0.44565,0.44735,0.44904,0.45071,0.45236,0.45400
,0.45561,0.45721,0.45879,0.46035,0.46188,0.46340,0.46489,0.46636,0.46780,0.46923,0.47063,0.47200,
0.47335,0.47467,0.47596,0.47723,0.47847,0.47969,0.48087,0.48203,0.48315,0.48425,0.48531,0.48635,0
.48735,0.48832,0.48926,0.49016,0.49103,0.49187,0.49268,0.49345,0.49418,0.49488,0.49554,0.49617,0.
49676,0.49732,0.49783,0.49831,0.49875,0.49916,0.49952,0.49985,0.50013,0.50038,0.50059,0.50076,0.5
0089,0.50098,0.50103,0.50104,0.50101,0.50093,0.50082,0.50066,0.50047,0.50023,0.49995,0.49963,0.49
927,0.49887,0.49843,0.49794,0.49741,0.49685,0.49624,0.49559,0.49489,0.49416,0.49339,0.49257,0.491
72,0.49082,0.48988,0.48890,0.48789,0.48683,0.48573,0.48459,0.48342,0.48220,0.48095)
  UT1_UTC =
(0.08566,0.08458,0.08362,0.08277,0.08206,0.08147,0.08094,0.08033,0.07952,0.07842,0.07703,0.07539,
0.07361,0.07180,0.07013,0.06875,0.06767,0.06679,0.06607,0.06538,0.06459,0.06360,0.06245,0.06121,0
.05987,0.05846,0.05700,0.05561,0.05436,0.05331,0.05246,0.05181,0.05130,0.05087,0.05039,0.04977,0.
04891,0.04781,0.04648,0.04502,0.04354,0.04218,0.04106,0.04022,0.03963,0.03924,0.03893,0.03857,0.0
3807,0.03738,0.03650,0.03547,0.03435,0.03321,0.03209,0.03105,0.03013,0.02936,0.02873,0.02824,0.02
780,0.02733,0.02670,0.02583,0.02465,0.02318,0.02153,0.01983,0.01825,0.01690,0.01583,0.01504,0.014
45,0.01395,0.01343,0.01279,0.01198,0.01098,0.00980,0.00850,0.00714,0.00580,0.00453,0.00338,0.0023
9,0.00155,0.00086,0.00025,-0.00036,-0.00109,-0.00205,-0.00334,-0.00497,-0.00686,-0.00888,-
0.01085,-0.01260,-0.01405,-0.01520,-0.01613,-0.01693,-0.01773,-0.01863,-0.01971,-0.02099,-
0.02246,-0.02409,-0.02581,-0.02754,-0.02921,-0.03076,-0.03216,-0.03339,-0.03446,-0.03542,-
0.03632,-0.03728,-0.03840,-0.03980,-0.04154,-0.04360,-0.04588,-0.04818,-0.05033,-0.05218,-
0.05366,-0.05484,-0.05583,-0.05677,-0.05778,-0.05893,-0.06025,-0.06174,-0.06337,-0.06510,-
0.06684,-0.06854,-0.07012,-0.07155,-0.07281,-0.07389,-0.07485,-0.07573,-0.07663,-0.07764,-
0.07886,-0.08038,-0.08223,-0.08434,-0.08658,-0.08875,-0.09069,-0.09229,-0.09355,-0.09454,-
0.09542,-0.09631,-0.09732,-0.09848,-0.09980,-0.10124,-0.10277,-0.10431,-0.10580,-0.10718,-
0.10840,-0.10943,-0.11028,-0.11096,-0.11155,-0.11211,-0.11274,-0.11353,-0.11455,-0.11584,-
0.11739,-0.11911,-0.12083,-0.12241,-0.12368)
END_GROUP = UT1_TIME_PARAMETERS
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","00000","00000","00000","00000")
  Status_Band7 =
("00000","00000","00000","00000","00000","00000","00000","00000","00000","00000","00000","00000",
"00000","00000","00000","00000")
  Status_Band8 =
("00000","00000","00000","00000","00000","00000","00000","00000","00000","00000","00000","00000",
"00000","00000","00000","00000","00000","00000","00000","00000","00000","00000","00000","00000","
00000","00000","00000","00000","00000","00000")
END_GROUP = DETECTOR_STATUS
GROUP = DETECTOR_GAINS
  GROUP = DETECTOR_GAINS_LOW
    B1L_Prelaunch =
(0.81539,0.81569,0.80851,0.81656,0.80959,0.81726,0.81510,0.81726,0.81972,0.82364,0.81647,0.81569,
0.81558,0.81421,0.81637,0.82413)
    B1L_Postlaunch =
(0.81823,0.81783,0.80966,0.81754,0.81015,0.81693,0.81472,0.81488,0.81880,0.82097,0.81406,0.81251,
0.81488,0.81594,0.81815,0.82591)

```
    B1L_Current =
(0.81799,0.81750,0.80957,0.81749,0.81003,0.81677,0.81467,0.81498,0.81855,0.82083,0.81422,0.81251,
0.81532,0.81641,0.81808,0.82585)
    B2L_Prelaunch =
(0.79631,0.79482,0.78627,0.79980,0.79164,0.79352,0.79342,0.78984,0.78915,0.80556,0.79114,0.79323,
0.79721,0.79393,0.79909,0.78627)
    B2L_Postlaunch =
(0.79776,0.79609,0.78776,0.80101,0.79164,0.79403,0.79284,0.78974,0.78839,0.80499,0.79077,0.79244,
0.79657,0.79395,0.79720,0.78602)
    B2L_Current =
(0.79746,0.78745,0.78744,0.79996,0.79186,0.79381,0.79329,0.78996,0.78878,0.80521,0.79057,0.79210,
0.79651,0.79394,0.79806,0.78682)
    B3L_Prelaunch =
(1.02746,1.02044,1.02350,1.02469,1.02370,1.03171,1.03417,1.02360,1.01866,1.02785,1.01728,1.02884,
1.02192,1.02578,1.01966,1.02212)
    B3L_Postlaunch =
(1.02799,1.02041,1.02390,1.02421,1.02328,1.03097,1.03486,1.02379,1.01888,1.02687,1.01693,1.02830,
1.02298,1.02656,1.01940,1.02205)
    B3L_Current =
(1.02779,1.02074,1.02399,1.02526,1.02314,1.03146,1.03397,1.02363,1.01817,1.02742,1.01668,1.02899,
1.02203,1.02624,1.01991,1.02282)
    B4L_Prelaunch =
(1.00155,0.99885,1.00308,0.98557,1.00135,1.00001,0.99761,1.00491,0.99087,0.99626,0.98750,0.99693,
0.99405,0.99751,0.98913,1.00578)
    B4L_Postlaunch =
(1.00200,0.99891,1.00320,0.98575,1.00160,0.99990,0.99711,1.00359,0.99203,0.99691,0.98724,0.99661,
0.99402,0.99761,0.98844,1.00608)
    B4L_Current =
(1.00257,0.99977,1.00358,0.98599,1.00195,1.00017,0.99770,1.00433,0.99132,0.99636,0.98717,0.99667,
0.99341,0.99711,0.98861,1.00640)
    B5L_Prelaunch =
(5.03398,5.06663,5.07855,5.05421,5.08496,5.02657,5.04109,5.08426,5.06803,5.08837,5.04810,5.04560,
5.03738,5.05932,5.03949,5.09518)
    B5L_Postlaunch =
(5.03903,5.07388,5.07588,5.05826,5.09772,5.02991,5.04964,5.07949,5.07899,5.10122,5.04353,5.03692,
5.02130,5.04453,5.03642,5.08500)
    B5L_Current =
(5.04091,5.07587,5.07901,5.05802,5.09723,5.03077,5.05160,5.08235,5.07142,5.09768,5.04620,5.03239,
5.02648,5.05153,5.03273,5.07932)
    B6L_Prelaunch = (12.283,12.474,13.150,12.511,12.805,12.646,13.108,12.794)
    B6L_Postlaunch = (12.426,12.614,13.270,12.625,12.899,12.893,13.217,12.969)
    B6L_Current = (12.435,12.620,13.276,12.628,12.898,12.758,13.211,12.951)
    B7L_Prelaunch =
(14.54238,14.52680,14.58439,14.51162,14.55705,14.59233,14.60841,14.48228,14.52429,14.53584,14.622
07,14.51916,14.21294,14.68861,14.51303,14.49303)
    B7L_Postlaunch =
(14.51063,14.46411,14.60801,14.52661,14.55857,14.57595,14.62399,14.52219,14.53385,14.53676,14.653
03,14.54550,14.19237,14.68207,14.50772,14.47285)
    B7L_Current =
(14.50706,14.48280,14.61788,14.53490,14.54235,14.58111,14.59652,14.50995,14.53182,14.54217,14.643
89,14.52400,14.21692,14.69388,14.51501,14.47046)
    B8L_Prelaunch =
(0.98287,0.99414,0.98206,0.99334,0.99072,0.99545,0.98679,0.99656,0.98277,0.98146,0.98821,0.98096,
0.98861,0.98468,0.98438,0.99464,0.99344,0.97783,0.99504,0.98428,0.99122,0.98589,0.99092,0.98166,0
.99636,0.98719,0.98780,0.98337,0.99313,0.99575,0.99344,0.98831)
    B8L_Postlaunch =
(0.98400,0.99576,0.98222,0.99448,0.99022,0.99586,0.98529,0.99645,0.98271,0.98083,0.98657,0.98103,
0.99092,0.98558,0.98400,0.99389,0.99448,0.98004,0.99418,0.98479,0.99122,0.98251,0.99122,0.98044,0
.99527,0.98667,0.98647,0.98351,0.99448,0.99487,0.99418,0.98914)
    B8L_Current =
(0.98400,0.99576,0.98222,0.99448,0.99022,0.99586,0.98529,0.99645,0.98271,0.98083,0.98657,0.98103,
0.99092,0.98558,0.98400,0.99389,0.99448,0.98004,0.99418,0.98479,0.99122,0.98251,0.99122,0.98044,0
.99527,0.98667,0.98647,0.98351,0.99448,0.99487,0.99418,0.98914)
  END_GROUP = DETECTOR_GAINS_LOW
  GROUP = DETECTOR_GAINS_HIGH
    B1H_Prelaunch =
(1.22405,1.22336,1.21383,1.22582,1.21451,1.22641,1.22287,1.22630,1.23112,1.23643,1.22523,1.22365,
1.22483,1.22306,1.22473,1.23711)
    B1H_Postlaunch =
(1.22807,1.22746,1.21521,1.22703,1.21594,1.22611,1.22281,1.22304,1.22892,1.23218,1.22182,1.21949,
1.22304,1.22464,1.22795,1.23959)
```

```
    B1H_Current =
(1.22803,1.22712,1.21536,1.22724,1.21611,1.22630,1.22281,1.22306,1.22901,1.23215,1.22207,1.21956,
1.22259,1.22435,1.22788,1.23963)
    B2H_Prelaunch =
(1.19510,1.19271,1.18036,1.20027,1.18664,1.18942,1.18893,1.18444,1.18424,1.20845,1.18723,1.18992,
1.19510,1.19092,1.19779,1.17906)
    B2H_Postlaunch =
(1.19657,1.19406,1.18157,1.20144,1.18739,1.19097,1.18918,1.18453,1.18252,1.20740,1.18608,1.18858,
1.19478,1.19085,1.19573,1.17895)
    B2H_Current =
(1.19663,1.18166,1.18154,1.20040,1.18714,1.19041,1.18950,1.18471,1.18307,1.20780,1.18624,1.18886,
1.19419,1.19057,1.19641,1.17941)
    B3H_Prelaunch =
(1.54197,1.53259,1.53429,1.53718,1.53629,1.54845,1.55294,1.53678,1.52820,1.54297,1.52711,1.54436,
1.53459,1.54107,1.52980,1.53289)
    B3H_Postlaunch =
(1.54289,1.53152,1.53674,1.53721,1.53582,1.54736,1.55319,1.53659,1.52922,1.54121,1.52629,1.54336,
1.53537,1.54075,1.52999,1.53398)
    B3H_Current =
(1.54333,1.53257,1.53633,1.53773,1.53541,1.54789,1.55255,1.53701,1.52843,1.54193,1.52569,1.54413,
1.53439,1.54101,1.53007,1.53442)
    B4H_Prelaunch =
(1.50174,1.49785,1.50437,1.47818,1.50252,1.49989,1.49531,1.50642,1.48850,1.49610,1.48227,1.49668,
1.49259,1.49785,1.48461,1.50963)
    B4H_Postlaunch =
(1.50351,1.49886,1.50530,1.47912,1.50290,1.50036,1.49616,1.50589,1.48854,1.49587,1.48136,1.49542,
1.49153,1.49692,1.48315,1.50963)
    B4H_Current =
(1.50367,1.49969,1.50557,1.47950,1.50283,1.50040,1.49576,1.50606,1.48807,1.49590,1.48144,1.49577,
1.49192,1.49748,1.48375,1.51046)
    B5H_Prelaunch =
(7.55469,7.59878,7.62118,7.58419,7.63018,7.54119,7.55799,7.61848,7.59718,7.63598,7.57749,7.57069,
7.54699,7.58149,7.56809,7.64298)
    B5H_Postlaunch =
(7.55854,7.61082,7.61382,7.58738,7.64657,7.54487,7.57446,7.61923,7.61848,7.65183,7.56530,7.55538,
7.53195,7.56680,7.55463,7.62749)
    B5H_Current =
(7.55944,7.61343,7.61647,7.58572,7.64538,7.54615,7.57231,7.61890,7.61047,7.64933,7.56928,7.55076,
7.53328,7.57133,7.55801,7.62922)
    B6H_Prelaunch = (23.953,24.325,25.642,24.397,24.969,24.659,25.561,24.949)
    B6H_Postlaunch = (24.231,24.597,25.876,24.618,25.153,25.142,25.774,25.289)
    B6H_Current = (24.257,24.700,26.097,24.783,25.211,24.980,25.851,25.424)
    B7H_Prelaunch =
(21.82563,21.80364,21.87966,21.77120,21.83747,21.89717,21.93090,21.74115,21.78742,21.79995,21.957
57,21.78373,21.30363,22.02712,21.76921,21.74891)
    B7H_Postlaunch =
(21.77175,21.70194,21.91786,21.79573,21.84368,21.86976,21.94183,21.78909,21.80658,21.81096,21.985
41,21.82407,21.29424,22.02898,21.76738,21.71506)
    B7H_Current =
(21.75815,21.72167,21.93070,21.80577,21.81935,21.86660,21.92634,21.79395,21.80156,21.81776,21.989
36,21.81000,21.31337,22.03100,21.77424,21.70837)
    B8H_Prelaunch =
(1.47469,1.49009,1.47114,1.49009,1.48593,1.49231,1.48026,1.49565,1.47530,1.47246,1.48420,1.47226,
1.48269,1.47732,1.47610,1.49231,1.49130,1.46730,1.49423,1.47550,1.48715,1.47803,1.48735,1.47459,1
.49697,1.48127,1.48300,1.47630,1.48958,1.49423,1.49049,1.48249)
    B8H_Postlaunch =
(1.47639,1.49404,1.47373,1.49212,1.48573,1.49419,1.47833,1.49508,1.47446,1.47164,1.48025,1.47194,
1.48677,1.47876,1.47639,1.49123,1.49212,1.47045,1.49167,1.47757,1.48722,1.47416,1.48722,1.47105,1
.49330,1.48040,1.48010,1.47565,1.49212,1.49270,1.49167,1.48411)
    B8H_Current =
(1.47639,1.49404,1.47373,1.49212,1.48573,1.49419,1.47833,1.49508,1.47446,1.47164,1.48025,1.47194,
1.48677,1.47876,1.47639,1.49123,1.49212,1.47045,1.49167,1.47757,1.48722,1.47416,1.48722,1.47105,1
.49330,1.48040,1.48010,1.47565,1.49212,1.49270,1.49167,1.48411)
  END_GROUP = DETECTOR_GAINS_HIGH
END_GROUP = DETECTOR_GAINS
GROUP = BIAS_LOCATIONS
  Forward_Bias_Location_30 = 143
  Forward_Bias_Length_30 = 500
  Forward_IC_Region_30 = 814
  Reverse_Bias_Location_30 = 780
  Reverse_Bias_Length_30 = 500
```

```
    Reverse_IC_Region_30 = 780
    Forward_Bias_Location_60 = 85
    Forward_Bias_Length_60 = 275
    Forward_IC_Region_60 = 380
    Reverse_Bias_Location_60 = 380
    Reverse_Bias_Length_60 = 275
    Reverse_IC_Region_60 = 380
    Forward_Bias_Location_15 = 286
    Forward_Bias_Length_15 = 1000
    Forward_IC_Region_15 = 1635
    Reverse_Bias_Location_15 = 1580
    Reverse_Bias_Length_15 = 1000
    Reverse_IC_Region_15 = 1580
END_GROUP = BIAS_LOCATIONS
GROUP = DETECTOR_BIASES_B6
  GROUP = DETECTOR_BIASES_B6_LOW
    B6L_Bias_Prelaunch = (31.51,30.12,25.27,29.86,27.84,28.91,25.65,27.87)
    B6L_Bias_Postlaunch = (25.96,24.86,20.14,24.76,22.76,22.93,20.57,22.43)
    B6L_Bias_Current = (29.825,28.782,24.286,28.685,26.768,26.941,24.678,26.463)
  END_GROUP = DETECTOR_BIASES_B6_LOW
  GROUP = DETECTOR_BIASES_B6_HIGH
    B6H_Bias_Prelaunch = (-66.23,-68.95,-78.39,-69.44,-73.38,-71.30,-77.66,-73.33)
    B6H_Bias_Postlaunch = (-77.10,-79.26,-88.45,-79.44,-83.35,-83.01,-87.62,-83.98)
    B6H_Bias_Current = (-69.566,-72.634,-82.61,-73.2228,-76.1918,-76.3519,-80.6467,-77.7089)
  END_GROUP = DETECTOR_BIASES_B6_HIGH
END_GROUP = DETECTOR_BIASES_B6
GROUP = ACCA_BIASES
  GROUP = ACCA_BIASES_LOW
    B1L_ACCA_Bias =
(9.91,9.87,10.11,10.02,10.06,10.02,10.12,10.03,10.00,9.97,10.08,10.09,10.02,10.07,9.96,10.03)
    B2L_ACCA_Bias =
(9.95,10.12,9.95,10.09,9.87,10.05,9.98,10.11,9.90,10.14,9.87,10.13,9.83,10.11,9.88,10.16)
    B3L_ACCA_Bias =
(10.20,9.79,10.18,9.70,10.08,9.66,10.20,9.84,10.23,9.84,10.24,9.83,10.13,9.72,10.11,9.74)
    B4L_ACCA_Bias =
(10.06,9.99,9.97,9.88,10.00,9.90,10.03,9.92,10.00,9.94,9.77,9.74,9.81,9.78,9.99,9.95)
    B5L_ACCA_Bias =
(10.02,10.03,10.00,9.98,10.06,10.07,10.01,10.07,10.09,10.09,9.90,9.97,10.10,10.08,10.07,10.06)
    B6L_ACCA_Bias = (29.825,28.782,24.286,28.685,26.768,26.941,24.678,26.463)
    B7L_ACCA_Bias =
(10.23,10.16,10.08,10.08,10.14,10.16,10.20,10.09,10.00,10.02,10.12,10.11,10.04,10.17,9.96,10.10)
    B8L_ACCA_Bias =
(10.49,9.40,10.62,9.48,10.25,9.75,10.26,9.64,9.90,10.37,10.02,10.26,9.10,10.37,9.49,9.90,9.28,9.4
5,9.05,9.02,8.60,10.19,8.53,10.02,8.29,8.82,7.99,9.02,7.34,8.98,7.45,8.95)
  END_GROUP = ACCA_BIASES_LOW
  GROUP = ACCA_BIASES_HIGH
    B1H_ACCA_Bias =
(14.86,14.80,15.13,15.00,15.14,15.08,15.23,15.09,14.94,14.93,15.07,15.09,15.08,15.15,14.99,15.08)
    B2H_ACCA_Bias =
(14.92,15.19,14.90,15.13,14.70,15.00,14.93,15.13,14.82,15.18,14.79,15.17,14.71,15.13,14.76,15.18)
    B3H_ACCA_Bias =
(15.19,14.58,15.33,14.61,15.07,14.46,15.32,14.80,15.23,14.64,15.24,14.63,15.18,14.55,15.22,14.66)
    B4H_ACCA_Bias =
(14.97,14.84,14.88,14.72,15.03,14.84,15.10,14.88,15.08,14.95,14.64,14.57,14.70,14.66,14.92,14.85)
    B5H_ACCA_Bias =
(14.98,15.00,15.05,15.02,15.02,15.04,15.01,15.10,15.18,15.19,14.81,14.93,15.04,15.01,15.04,15.03)
    B6H_ACCA_Bias = (-69.566,-72.634,-82.61,-73.2228,-76.1918,-76.3519,-80.6467,-77.7089)
    B7H_ACCA_Bias =
(15.32,15.15,15.13,15.11,15.14,15.16,15.31,15.12,15.03,15.04,15.11,15.10,15.02,15.22,14.91,15.13)
    B8H_ACCA_Bias =
(15.85,14.28,16.29,14.39,15.51,14.87,15.49,14.54,14.99,15.60,15.12,15.25,13.80,15.59,14.27,14.82,
14.06,14.18,13.46,13.52,12.70,15.49,12.56,15.21,12.21,13.14,11.91,13.21,10.91,13.12,11.16,13.00)
  END_GROUP = ACCA_BIASES_HIGH
END_GROUP = ACCA_BIASES
GROUP = ACCA_THRESHOLDS
  Thresh_B3 = 0.0800
  Thresh_B3_Lower = 0.07
  Thresh_B56_High = 225.000
  Thresh_B56_Low = 210.000
  Thresh_B6 = 300.000
  Thresh_B45_Ratio = 1.0000
```

```
      Thresh_B42_Ratio = 2.16248
      Thresh_B43_Ratio = 2.3500
      Thresh_NDSI_Max = 0.7000
      Thresh_NDSI_Min = -0.2500
      Thresh_NDSI_Snow = 0.8000
      Cloud_Percent_Min = 0.4000
      Desert_Index = 0.500
      Thresh_Snow_Percent = 1.0000
      Thermal_Effect_High = 35.0000
      Thermal_Effect_Low = 25.000
      B6Max_Maxthresh_Diff = 2.000
  END_GROUP = ACCA_THRESHOLDS
  GROUP = SOLAR_SPECTRAL_IRRADIANCES
      B1_Solar_Irradiance = 1969.000
      B2_Solar_Irradiance = 1840.000
      B3_Solar_Irradiance = 1551.000
      B4_Solar_Irradiance = 1044.000
      B5_Solar_Irradiance = 225.7
      B7_Solar_Irradiance = 82.07
      B8_Solar_Irradiance = 1368.000
  END_GROUP = SOLAR_SPECTRAL_IRRADIANCES
  GROUP = THERMAL_CONSTANTS
      K1_Constant = 666.09
      K2_Constant = 1282.71
  END_GROUP = THERMAL_CONSTANTS
  GROUP = SCALING_PARAMETERS
      GROUP = SCALING_PARAMETERS_LOW
        B1L_Lmin_Lmax = (-6.2,293.7)
        B2L_Lmin_Lmax = (-6.4,300.9)
        B3L_Lmin_Lmax = (-5.0,234.4)
        B4L_Lmin_Lmax = (-5.1,241.1)
        B5L_Lmin_Lmax = (-1.0,47.57)
        B6L_Lmin_Lmax = (0.0,17.04)
        B7L_Lmin_Lmax = (-0.35,16.54)
        B8L_Lmin_Lmax = (-4.7,243.1)
      END_GROUP = SCALING_PARAMETERS_LOW
      GROUP = SCALING_PARAMETERS_HIGH
        B1H_Lmin_Lmax = (-6.2,191.6)
        B2H_Lmin_Lmax = (-6.4,196.5)
        B3H_Lmin_Lmax = (-5.0,152.9)
        B4H_Lmin_Lmax = (-5.1,157.4)
        B5H_Lmin_Lmax = (-1.0,31.06)
        B6H_Lmin_Lmax = (3.2,12.65)
        B7H_Lmin_Lmax = (-0.35,10.80)
        B8H_Lmin_Lmax = (-4.7,158.3)
      END_GROUP = SCALING_PARAMETERS_HIGH
  END_GROUP = SCALING_PARAMETERS
  GROUP = MTF_COMPENSATION
      B1_weights_along = (1.56766583,0.00000000,-1.56766583,-0.01966520,-0.01966520)
      B1_weights_across = (1.45063128,0.00000000,-1.45063128,0.00257381,0.00257381)
      B2_weights_along = (1.61050310,0.00000000,-1.61050310,-0.02774139,-0.02774139)
      B2_weights_across = (1.49221631,0.00000000,-1.49221631,-0.00535953,-0.00535953)
      B3_weights_along = (1.65047774,0.00000000,-1.65047774,-0.03525043,-0.03525043)
      B3_weights_across = (1.52298447,0.00000000,-1.52298447,-0.01120648,-0.01120648)
      B4_weights_along = (1.73786071,0.00000000,-1.73786071,-0.05158080,-0.05158080)
      B4_weights_across = (1.55814152,0.00000000,-1.55814152,-0.01786521,-0.01786521)
      B5_weights_along = (1.54118459,0.00000000,-1.54118459,-0.01465645,-0.01465645)
      B5_weights_across = (1.43789226,0.00000000,-1.43789226,0.00501156,0.00501156)
      B6_weights_along = (1.74511478,0.00000000,-1.74511478,-0.05293163,-0.05293163)
      B6_weights_across = (1.24858736,0.00000000,-1.24858736,0.04172298,0.04172298)
      B7_weights_along = (1.47951767,0.00000000,-1.47951767,-0.00294082,-0.00294082)
      B7_weights_across = (1.42261190,0.00000000,-1.42261190,0.00794044,0.00794044)
      B8_weights_along = (1.94052085,0.00000000,-1.94052085,-0.08907866,-0.08907866)
      B8_weights_across = (2.11745387,0.00000000,-2.11745387,-0.12147250,-0.12147250)
  END_GROUP = MTF_COMPENSATION
  GROUP = MEMORY_EFFECT
      GROUP = ME_MAGNITUDES
        B1_ME_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)
        B2_ME_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)
        B3_ME_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)
        B4_ME_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)
```

```
    B5_ME_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_ME_Magnitude = (0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)
    B7_ME_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)
    B8_ME_Magnitude =
(0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)
  END_GROUP = ME_MAGNITUDES
  GROUP = ME_TIME_CONSTANTS
    B1_ME_Time_Constant =
(1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.
0,1100.0,1100.0)
    B2_ME_Time_Constant =
(1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.
0,1100.0,1100.0)
    B3_ME_Time_Constant =
(1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.
0,1100.0,1100.0)
    B4_ME_Time_Constant =
(1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.
0,1100.0,1100.0)
    B5_ME_Time_Constant =
(1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.
0,1100.0,1100.0)
    B6_ME_Time_Constant = (1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0)
    B7_ME_Time_Constant =
(1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.
0,1100.0,1100.0)
    B8_ME_Time_Constant =
(1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.
0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100.0,1100
.0,1100.0,1100.0,1100.0,1100.0)
  END_GROUP = ME_TIME_CONSTANTS
END_GROUP = MEMORY_EFFECT
GROUP = GHOST_PULSE
  Ghost_Pulse_Endpoints = (0.00,0.00)
END_GROUP = GHOST_PULSE
GROUP = SCAN_CORRELATED_SHIFT
  SCS_Reference_Detectors = (1,1,1,1,1,1,1)
  GROUP = SCS_LOW
    B1L_SCS_Magnitudes = (0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)
    B2L_SCS_Magnitudes = (0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)
    B3L_SCS_Magnitudes = (0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)
    B4L_SCS_Magnitudes = (0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)
    B5L_SCS_Magnitudes = (0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)
    B7L_SCS_Magnitudes = (0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)
    B8L_SCS_Magnitudes =
(0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)
  END_GROUP = SCS_LOW
  GROUP = SCS_HIGH
    B1H_SCS_Magnitudes = (0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)
    B2H_SCS_Magnitudes = (0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)
    B3H_SCS_Magnitudes = (0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)
    B4H_SCS_Magnitudes = (0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)
    B5H_SCS_Magnitudes = (0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)
    B7H_SCS_Magnitudes = (0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)
    B8H_SCS_Magnitudes =
(0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)
  END_GROUP = SCS_HIGH
END_GROUP = SCAN_CORRELATED_SHIFT
GROUP = STRIPING
  GROUP = STRIPING_FLAG_LOW
    Correction_Reference_B1_Low = 2
    Correction_Reference_B2_Low = 2
    Correction_Reference_B3_Low = 2
    Correction_Reference_B4_Low = 2
    Correction_Reference_B5_Low = 2
    Correction_Reference_B6_Low = 0
    Correction_Reference_B7_Low = 2
    Correction_Reference_B8_Low = 2
```

```
      END_GROUP = STRIPING_FLAG_LOW
    GROUP = STRIPING_FLAG_HIGH
      Correction_Reference_B1_High = 2
      Correction_Reference_B2_High = 2
      Correction_Reference_B3_High = 2
      Correction_Reference_B4_High = 2
      Correction_Reference_B5_High = 2
      Correction_Reference_B6_High = 0
      Correction_Reference_B7_High = 2
      Correction_Reference_B8_High = 2
    END_GROUP = STRIPING_FLAG_HIGH
  END_GROUP = STRIPING
  GROUP = HISTOGRAM
    GROUP = DETECTOR_NOISE
      GROUP = DETECTOR_NOISE_LOW
        Detector_Noise_Level_B1_Low =
(0.779460,0.772285,0.728671,0.763302,0.786943,0.776291,0.770672,0.739825,0.804123,0.737660,0.7495
92,0.794155,0.765984,0.780631,0.743702,0.759316)
        Detector_Noise_Level_B2_Low =
(0.601029,0.576235,0.578696,0.572740,0.587501,0.591210,0.577901,0.596892,0.612274,0.599762,0.6019
85,0.572298,0.605657,0.588833,0.586422,0.574523)
        Detector_Noise_Level_B3_Low =
(0.814260,0.797001,0.806253,0.825438,0.805607,0.835023,0.802352,0.759353,0.814861,0.804235,0.8303
29,0.835847,0.795850,0.819125,0.760094,0.808781)
        Detector_Noise_Level_B4_Low =
(0.373855,0.357412,0.410801,0.401166,0.386735,0.385534,0.351177,0.388469,0.298362,0.346134,0.4544
76,0.467169,0.430189,0.458503,0.309542,0.349836)
        Detector_Noise_Level_B5_Low =
(0.541758,0.538805,0.564040,0.528059,0.567236,0.564582,0.557496,0.559557,0.576319,0.563746,0.5487
35,0.554186,0.578348,0.541598,0.546643,0.543750)
        Detector_Noise_Level_B6_Low =
(0.38498,0.40307,0.39786,0.37927,0.38451,0.38594,0.37989,0.37353)
        Detector_Noise_Level_B7_Low =
(0.882830,0.841372,0.840801,0.836801,0.918675,0.888781,0.868970,0.833833,0.889666,0.816621,0.8723
48,0.823312,0.844422,0.839332,0.818657,0.838030)
        Detector_Noise_Level_B8_Low = (1.411792, 1.429218, 1.475390, 1.417629, 1.426976, 1.455102,
1.447059, 1.445895, 1.428523, 1.518779, 1.410482, 1.546677, 1.412992, 1.568844, 1.432523,
1.406441, 1.466322, 1.523012, 1.488277, 1.538726, 1.419587, 1.438650, 1.437786, 1.435104,
1.447014, 1.411219, 1.420777, 1.483956, 1.469533, 1.473144, 1.436488, 1.435460)
      END_GROUP = DETECTOR_NOISE_LOW
      GROUP = DETECTOR_NOISE_HIGH
        Detector_Noise_Level_B1_High =
(1.115767,1.105730,1.034492,1.088057,1.122772,1.109671,1.093771,1.045665,1.135931,1.051208,1.0683
69,1.137889,1.088930,1.113233,1.060574,1.083090)
        Detector_Noise_Level_B2_High =
(0.840525,0.802531,0.811894,0.802952,0.821014,0.828448,0.799075,0.826498,0.848278,0.828466,0.8339
49,0.787364,0.842703,0.814721,0.810008,0.789331)
        Detector_Noise_Level_B3_High =
(1.171068,1.145406,1.154020,1.185309,1.152540,1.197513,1.145797,1.079392,1.168113,1.152881,1.1973
19,1.204024,1.142140,1.177719,1.081946,1.158203)
        Detector_Noise_Level_B4_High =
(0.571712,0.568938,0.601790,0.577678,0.585365,0.570769,0.564726,0.575154,0.523646,0.549596,0.5781
82,0.564183,0.560860,0.580182,0.528458,0.559510)
        Detector_Noise_Level_B5_High =
(0.758784,0.753994,0.791165,0.746657,0.796592,0.791935,0.769362,0.774648,0.810945,0.794850,0.7724
70,0.783913,0.811670,0.761202,0.770891,0.766004)
        Detector_Noise_Level_B6_High =
(0.63424,0.64766,0.63594,0.62451,0.62835,0.64975,0.62390,0.60717)
        Detector_Noise_Level_B7_High =
(1.292273,1.226330,1.215933,1.208056,1.337354,1.291756,1.261227,1.208343,1.296625,1.183313,1.2693
91,1.193715,1.221458,1.216385,1.179451,1.211407)
        Detector_Noise_Level_B8_High =
(2.01844,2.03392,2.10339,1.99754,2.63946,2.07794,2.46647,2.03946,2.51454,2.12575,2.02078,2.16917,
1.98090,2.18632,2.01149,1.98286,2.07499,2.11379,2.10509,2.14305,2.00573,2.09848,2.03111,2.08271,2
.06095,2.01814,1.98271,2.17801,2.44790,2.15665,2.25689,2.24758)
      END_GROUP = DETECTOR_NOISE_HIGH
    END_GROUP = DETECTOR_NOISE
    GROUP = DET_SHUTTER_NOISE
      GROUP = DET_SHUTTER_NOISE_LOW
```

Det_Shutter_Noise_Level_B1_Low =
(0.779460,0.772285,0.728671,0.763302,0.786943,0.776291,0.770672,0.739825,0.804123,0.737660,0.7495
92,0.794155,0.765984,0.780631,0.743702,0.759316)
Det_Shutter_Noise_Level_B2_Low =
(0.601029,0.576235,0.578696,0.572740,0.587501,0.591210,0.577901,0.596892,0.612274,0.599762,0.6019
85,0.572298,0.605657,0.588833,0.586422,0.574523)
Det_Shutter_Noise_Level_B3_Low =
(0.814260,0.797001,0.806253,0.825438,0.805607,0.835023,0.802352,0.759353,0.814861,0.804235,0.8303
29,0.835847,0.795850,0.819125,0.760094,0.808781)
Det_Shutter_Noise_Level_B4_Low =
(0.373855,0.357412,0.410801,0.401166,0.386735,0.385534,0.351177,0.388469,0.298362,0.346134,0.4544
76,0.467169,0.430189,0.458503,0.309542,0.349836)
Det_Shutter_Noise_Level_B5_Low =
(0.541758,0.538805,0.564040,0.528059,0.567236,0.564582,0.557496,0.559557,0.576319,0.563746,0.5487
35,0.554186,0.578348,0.541598,0.546643,0.543750)
Det_Shutter_Noise_Level_B6_Low =
(0.38498,0.40307,0.39786,0.37927,0.38451,0.38594,0.37989,0.37353)
Det_Shutter_Noise_Level_B7_Low =
(0.882830,0.841372,0.840801,0.836801,0.918675,0.888781,0.868970,0.833833,0.889666,0.816621,0.8723
48,0.823312,0.844422,0.839332,0.818657,0.838030)
Det_Shutter_Noise_Level_B8_Low = (1.411792, 1.429218, 1.475390, 1.417629, 1.426976,
1.455102, 1.447059, 1.445895, 1.428523, 1.518779, 1.410482, 1.546677, 1.412992, 1.568844,
1.432523, 1.406441, 1.466322, 1.523012, 1.488277, 1.538726, 1.419587, 1.438650, 1.437786,
1.435104, 1.447014, 1.411219, 1.420777, 1.483956, 1.469533, 1.473144, 1.436488, 1.435460)
END_GROUP = DET_SHUTTER_NOISE_LOW
GROUP = DET_SHUTTER_NOISE_HIGH
Det_Shutter_Noise_Level_B1_High =
(1.115767,1.105730,1.034492,1.088057,1.122772,1.109671,1.093771,1.045665,1.135931,1.051208,1.0683
69,1.137889,1.088930,1.113233,1.060574,1.083090)
Det_Shutter_Noise_Level_B2_High =
(0.840525,0.802531,0.811894,0.802952,0.821014,0.828448,0.799075,0.826498,0.848278,0.828466,0.8339
49,0.787364,0.842703,0.814721,0.810008,0.789331)
Det_Shutter_Noise_Level_B3_High =
(1.171068,1.145406,1.154020,1.185309,1.152540,1.197513,1.145797,1.079392,1.168113,1.152881,1.1973
19,1.204024,1.142140,1.177719,1.081946,1.158203)
Det_Shutter_Noise_Level_B4_High =
(0.571712,0.568938,0.601790,0.577678,0.585365,0.570769,0.564726,0.575154,0.523646,0.549596,0.5781
82,0.564183,0.560860,0.580182,0.528458,0.559510)
Det_Shutter_Noise_Level_B5_High =
(0.758784,0.753994,0.791165,0.746657,0.796592,0.791935,0.769362,0.774648,0.810945,0.794850,0.7724
70,0.783913,0.811670,0.761202,0.770891,0.766004)
Det_Shutter_Noise_Level_B6_High =
(0.63424,0.64766,0.63594,0.62451,0.62835,0.64975,0.62390,0.60717)
Det_Shutter_Noise_Level_B7_High =
(1.292273,1.226330,1.215933,1.208056,1.337354,1.291756,1.261227,1.208343,1.296625,1.183313,1.2693
91,1.193715,1.221458,1.216385,1.179451,1.211407)
Det_Shutter_Noise_Level_B8_High =
(2.01844,2.03392,2.10339,1.99754,2.63946,2.07794,2.46647,2.03946,2.51454,2.12575,2.02078,2.16917,
1.98090,2.18632,2.01149,1.98286,2.07499,2.11379,2.10509,2.14305,2.00573,2.09848,2.03111,2.08271,2
.06095,2.01814,1.98271,2.17801,2.44790,2.15665,2.25689,2.24758)
END_GROUP = DET_SHUTTER_NOISE_HIGH
END_GROUP = DET_SHUTTER_NOISE
GROUP = REFERENCE_DETECTORS
Reference_Detector_B1 = 15
Reference_Detector_B2 = 12
Reference_Detector_B3 = 08
Reference_Detector_B4 = 07
Reference_Detector_B5 = 14
Reference_Detector_B6 = 01
Reference_Detector_B7 = 10
Reference_Detector_B8 = 27
END_GROUP = REFERENCE_DETECTORS
GROUP = SATURATION_THRESHOLDS
Saturation_Bin_Threshold_B1 = 1000
Saturation_Bin_Threshold_B2 = 1000
Saturation_Bin_Threshold_B3 = 1000
Saturation_Bin_Threshold_B4 = 1000
Saturation_Bin_Threshold_B5 = 1000
Saturation_Bin_Threshold_B6 = 1000
Saturation_Bin_Threshold_B7 = 1000
Saturation_Bin_Threshold_B8 = 1000

```
    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 = 2
        Adjacent_Bin_Number_B7 = 2
        Adjacent_Bin_Number_B8 = 2
      END_GROUP = BIN_NUMBER
      GROUP = BIN_THRESHOLD
        Adjacent_Bin_Threshold_B1 = 10
        Adjacent_Bin_Threshold_B2 = 10
        Adjacent_Bin_Threshold_B3 = 10
        Adjacent_Bin_Threshold_B4 = 10
        Adjacent_Bin_Threshold_B5 = 10
        Adjacent_Bin_Threshold_B6 = 10
        Adjacent_Bin_Threshold_B7 = 10
        Adjacent_Bin_Threshold_B8 = 10
      END_GROUP = BIN_THRESHOLD
    END_GROUP = ADJACENT_BINS
    GROUP = STARTING_PIXEL
      Start_pixel_B1 = 243
      Start_pixel_B2 = 218
      Start_pixel_B3 = 193
      Start_pixel_B4 = 168
      Start_pixel_B5 = 97
      Start_pixel_B6 = 31
      Start_pixel_B7 = 123
      Start_pixel_B8 = 536
    END_GROUP = STARTING_PIXEL
    GROUP = WINDOW_WIDTH
      Window_Samples_B1 = 5874
      Window_Samples_B2 = 5874
      Window_Samples_B3 = 5874
      Window_Samples_B4 = 5874
      Window_Samples_B5 = 5874
      Window_Samples_B6 = 2937
      Window_Samples_B7 = 5874
      Window_Samples_B8 = 11748
    END_GROUP = WINDOW_WIDTH
    GROUP = WINDOW_LENGTH
      Window_Scans_B1 = 375
      Window_Scans_B2 = 375
      Window_Scans_B3 = 375
      Window_Scans_B4 = 375
      Window_Scans_B5 = 375
      Window_Scans_B6 = 375
      Window_Scans_B7 = 375
      Window_Scans_B8 = 375
    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
      Overlap_Scans_B8 = 0
    END_GROUP = OVERLAPPING_SCANS
  END_GROUP = HISTOGRAM
GROUP = IMPULSE_NOISE
  Median_Filter_Width = 3
  GROUP = IN_THRESHOLD
    B1L_Threshold = (10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33,
10.33, 10.33, 10.33, 10.33, 10.33)
    B2L_Threshold = (10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33,
10.33, 10.33, 10.33, 10.33, 10.33)
```

```
    B3L_Threshold = (10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33,
10.33, 10.33, 10.33, 10.33, 10.33)
    B4L_Threshold = (20.67, 20.67, 20.67, 20.67, 20.67, 20.67, 20.67, 20.67, 20.67, 20.67, 20.67,
20.67, 20.67, 20.67, 20.67, 20.67)
    B5L_Threshold = (10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33,
10.33, 10.33, 10.33, 10.33, 10.33)
    B6L_Threshold = (20.67, 20.67, 20.67, 20.67, 20.67, 20.67, 20.67)
    B7L_Threshold = (10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33,
10.33, 10.33, 10.33, 10.33, 10.33)
    B8L_Threshold = (6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 5.17, 6.89, 5.17,
6.89, 5.17, 6.89, 6.89, 6.89, 5.17, 6.89, 5.17, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89,
6.89, 6.89, 6.89, 6.89)
    B1H_Threshold = (6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89,
6.89, 6.89, 6.89, 6.89)
    B2H_Threshold = (10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33,
10.33, 10.33, 10.33, 10.33)
    B3H_Threshold = (6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89,
6.89, 6.89, 6.89, 6.89)
    B4H_Threshold = (10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33,
10.33, 10.33, 10.33, 10.33, 10.33)
    B5H_Threshold = (10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33,
10.33, 10.33, 10.33, 10.33, 10.33)
    B6H_Threshold = (10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33, 10.33)
    B7H_Threshold = (6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89, 6.89,
6.89, 6.89, 6.89, 6.89)
    B8H_Threshold = (4.13, 4.13, 4.13, 5.17, 3.44, 4.13, 4.13, 4.13, 3.44, 4.13, 4.13, 4.13,
5.17, 4.13, 4.13, 5.17, 4.13, 4.13, 4.13, 4.13, 4.13, 4.13, 4.13, 4.13, 4.13, 4.13, 5.17, 4.13,
4.13, 4.13, 4.13, 4.13)
  END_GROUP = IN_THRESHOLD
  GROUP = IN_SIGMA_THRESHOLD
    B1L_Sigma_Threshold = (13.26, 13.38, 14.18, 13.54, 13.13, 13.31, 13.41, 13.97, 12.85, 14.01,
13.78, 13.01, 13.49, 13.24, 13.89, 13.61)
    B2L_Sigma_Threshold = (17.19, 17.93, 17.86, 18.04, 17.59, 17.48, 17.88, 17.31, 16.88, 17.23,
17.16, 18.06, 17.06, 17.55, 17.62, 17.99)
    B3L_Sigma_Threshold = (12.69, 12.96, 12.82, 12.52, 12.83, 12.37, 12.88, 13.61, 12.68, 12.85,
12.44, 12.36, 12.98, 12.61, 13.59, 12.78)
    B4L_Sigma_Threshold = (27.64, 28.91, 25.15, 25.76, 26.72, 26.80, 29.42, 26.60, 34.63, 29.85,
22.74, 22.12, 24.02, 22.54, 33.38, 29.54)
    B5L_Sigma_Threshold = (19.07, 19.18, 18.32, 19.57, 18.22, 18.30, 18.53, 18.47, 17.93, 18.33,
18.83, 18.65, 17.87, 19.08, 18.90, 19.00)
    B6L_Sigma_Threshold = (26.84, 25.64, 25.97, 27.24, 26.87, 26.77, 27.20, 27.66)
    B7L_Sigma_Threshold = (11.70, 12.28, 12.29, 12.35, 11.25, 11.63, 11.89, 12.39, 11.61, 12.65,
11.85, 12.55, 12.24, 12.31, 12.62, 12.33)
    B8L_Sigma_Threshold = (7.32, 7.23, 7.00, 7.29, 7.24, 7.10, 7.14, 7.15, 7.23, 6.80, 7.33,
6.68, 7.31, 6.59, 7.21, 7.35, 7.05, 6.78, 6.94, 6.72, 7.28, 7.18, 7.19, 7.20, 7.14, 7.32, 7.27,
6.96, 7.03, 7.01, 7.19, 7.20)
    B1H_Sigma_Threshold = (9.26, 9.34, 9.99, 9.50, 9.20, 9.31, 9.45, 9.88, 9.10, 9.83, 9.67,
9.08, 9.49, 9.28, 9.74, 9.54)
    B2H_Sigma_Threshold = (12.29, 12.88, 12.73, 12.87, 12.59, 12.47, 12.93, 12.50, 12.18, 12.47,
12.39, 13.12, 12.26, 12.68, 12.76, 13.09)
    B3H_Sigma_Threshold = (8.82, 9.02, 8.95, 8.72, 8.97, 8.63, 9.02, 9.57, 8.85, 8.96, 8.63,
8.58, 9.05, 8.77, 9.55, 8.92)
    B4H_Sigma_Threshold = (18.07, 18.16, 17.17, 17.89, 17.65, 18.10, 18.30, 17.97, 19.73, 18.80,
17.87, 18.31, 18.42, 17.81, 19.55, 18.47)
    B5H_Sigma_Threshold = (13.62, 13.70, 13.06, 13.84, 12.97, 13.05, 13.43, 13.34, 12.74, 13.00,
13.38, 13.18, 12.73, 13.57, 13.40, 13.49)
    B6H_Sigma_Threshold = (16.29, 15.95, 16.25, 16.55, 16.44, 15.90, 16.56, 17.02)
    B7H_Sigma_Threshold = (8.00, 8.43, 8.50, 8.55, 7.73, 8.00, 8.19, 8.55, 7.97, 8.73, 8.14,
8.66, 8.46, 8.49, 8.76, 8.53)
    B8H_Sigma_Threshold = (5.12, 5.08, 4.91, 5.17, 3.91, 4.97, 4.19, 5.07, 4.11, 4.86, 5.11,
4.76, 5.22, 4.73, 5.14, 5.21, 4.98, 4.89, 4.91, 4.82, 5.15, 4.92, 5.09, 4.96, 5.01, 5.12, 5.21,
4.74, 4.22, 4.79, 4.58, 4.60)
  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)
      B8_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)
      B8_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)
      B8_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)
      B8_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_PARAMETERS
    GROUP = 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)
      B8_Phase_Mean = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    END_GROUP = PHASE_MEANS
    GROUP = PHASE_SIGMAS
      B1_Phase_Sigma = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B2_Phase_Sigma = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B3_Phase_Sigma = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B4_Phase_Sigma = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B5_Phase_Sigma = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B6_Phase_Sigma = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B7_Phase_Sigma = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B8_Phase_Sigma = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    END_GROUP = PHASE_SIGMAS
    GROUP = PHASE_MINIMUMS
      B1_Phase_Min = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B2_Phase_Min = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B3_Phase_Min = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B4_Phase_Min = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B5_Phase_Min = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B6_Phase_Min = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B7_Phase_Min = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B8_Phase_Min = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
    END_GROUP = PHASE_MINIMUMS
    GROUP = PHASE_MAXIMUMS
      B1_Phase_Max = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
      B2_Phase_Max = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
```

```
            B3_Phase_Max = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
            B4_Phase_Max = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
            B5_Phase_Max = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
            B6_Phase_Max = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
            B7_Phase_Max = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
            B8_Phase_Max = (0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00)
          END_GROUP = PHASE_MAXIMUMS
        END_GROUP = CN_PHASE_PARAMETERS
        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)
            B8_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)
            B8_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)
            B8_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)
            B8_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 = DETECTOR_SATURATION
        GROUP = AD_CONVERTER_SATURATION
          GROUP = AD_CONVERTER_SATURATION_LOW
            High_AD_Level_B1_low = (255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
            High_AD_Level_B2_low = (255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
            High_AD_Level_B3_low = (255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
            High_AD_Level_B4_low = (255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
            High_AD_Level_B5_low = (255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
            High_AD_Level_B6_low = (255,255,255,255,255,255,255,255)
            High_AD_Level_B7_low = (255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
            High_AD_Level_B8_low =
(255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255)
            Low_AD_Level_B1_low = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
            Low_AD_Level_B2_low = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
            Low_AD_Level_B3_low = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
            Low_AD_Level_B4_low = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
            Low_AD_Level_B5_low = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
            Low_AD_Level_B6_low = (0,0,0,0,0,0,0,0)
            Low_AD_Level_B7_low = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
```

```
        Low_AD_Level_B8_low = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
      END_GROUP = AD_CONVERTER_SATURATION_LOW
      GROUP = AD_CONVERTER_SATURATION_HIGH
        High_AD_Level_B1_high = (255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
        High_AD_Level_B2_high = (255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
        High_AD_Level_B3_high = (255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
        High_AD_Level_B4_high = (255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
        High_AD_Level_B5_high = (255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
        High_AD_Level_B6_high = (255,255,255,255,255,255,255,255)
        High_AD_Level_B7_high = (255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
        High_AD_Level_B8_high =
(255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255)
        Low_AD_Level_B1_high = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
        Low_AD_Level_B2_high = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
        Low_AD_Level_B3_high = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
        Low_AD_Level_B4_high = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
        Low_AD_Level_B5_high = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
        Low_AD_Level_B6_high = (0,0,0,0,0,0,0,0)
        Low_AD_Level_B7_high = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
        Low_AD_Level_B8_high = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
      END_GROUP = AD_CONVERTER_SATURATION_HIGH
    END_GROUP = AD_CONVERTER_SATURATION
    GROUP = ANALOG_SIGNAL_SATURATION
      GROUP = ANALOG_SIGNAL_SATURATION_LOW
        High_Analog_Level_B1_low =
(255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
        High_Analog_Level_B2_low =
(255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
        High_Analog_Level_B3_low =
(255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
        High_Analog_Level_B4_low =
(255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
        High_Analog_Level_B5_low =
(255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
        High_Analog_Level_B6_low = (255,255,255,255,255,255,255,255)
        High_Analog_Level_B7_low =
(255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
        High_Analog_Level_B8_low =
(255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255)
        Low_Analog_Level_B1_low = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
        Low_Analog_Level_B2_low = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
        Low_Analog_Level_B3_low = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
        Low_Analog_Level_B4_low = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
        Low_Analog_Level_B5_low = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
        Low_Analog_Level_B6_low = (0,0,0,0,0,0,0,0)
        Low_Analog_Level_B7_low = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
        Low_Analog_Level_B8_low = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
      END_GROUP = ANALOG_SIGNAL_SATURATION_LOW
      GROUP = ANALOG_SIGNAL_SATURATION_HIGH
        High_Analog_Level_B1_high =
(255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
        High_Analog_Level_B2_high =
(255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
        High_Analog_Level_B3_high =
(255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
        High_Analog_Level_B4_high =
(255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
        High_Analog_Level_B5_high =
(255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
        High_Analog_Level_B6_high = (255,255,255,255,255,255,255,255)
        High_Analog_Level_B7_high =
(255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255)
        High_Analog_Level_B8_high =
(255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255)
        Low_Analog_Level_B1_high = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
        Low_Analog_Level_B2_high = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
        Low_Analog_Level_B3_high = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
        Low_Analog_Level_B4_high = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
```

```
        Low_Analog_Level_B5_high = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
        Low_Analog_Level_B6_high = (0,0,0,0,0,0,0,0)
        Low_Analog_Level_B7_high = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
        Low_Analog_Level_B8_high =
(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
      END_GROUP = ANALOG_SIGNAL_SATURATION_HIGH
    END_GROUP = ANALOG_SIGNAL_SATURATION
END_GROUP = DETECTOR_SATURATION
GROUP = REFERENCE_TEMPERATURES
  GROUP = REFERENCE_LOW
    B1L_RTemp_Prelaunch = 25.00
    B1L_RTemp_Postlaunch = 25.00
    B1L_RTemp_Current = 25.00
    B2L_RTemp_Prelaunch = 25.00
    B2L_RTemp_Postlaunch = 25.00
    B2L_RTemp_Current = 25.00
    B3L_RTemp_Prelaunch = 25.00
    B3L_RTemp_Postlaunch = 25.00
    B3L_RTemp_Current = 25.00
    B4L_RTemp_Prelaunch = 25.00
    B4L_RTemp_Postlaunch = 25.00
    B4L_RTemp_Current = 25.00
    B5L_RTemp_Prelaunch = -182.1
    B5L_RTemp_Postlaunch = -182.1
    B5L_RTemp_Current = -182.1
    B6L_RTemp_Prelaunch = -182.2
    B6L_RTemp_Postlaunch = -182.1
    B6L_RTemp_Current = -182.2
    B7L_RTemp_Prelaunch = -182.1
    B7L_RTemp_Postlaunch = -182.1
    B7L_RTemp_Current = -182.1
    B8L_RTemp_Prelaunch = 25.00
    B8L_RTemp_Postlaunch = 25.00
    B8L_RTemp_Current = 25.00
  END_GROUP = REFERENCE_LOW
  GROUP = REFERENCE_HIGH
    B1H_RTemp_Prelaunch = 25.00
    B1H_RTemp_Postlaunch = 25.00
    B1H_RTemp_Current = 25.00
    B2H_RTemp_Prelaunch = 25.00
    B2H_RTemp_Postlaunch = 25.00
    B2H_RTemp_Current = 25.00
    B3H_RTemp_Prelaunch = 25.00
    B3H_RTemp_Postlaunch = 25.00
    B3H_RTemp_Current = 25.00
    B4H_RTemp_Prelaunch = 25.00
    B4H_RTemp_Postlaunch = 25.00
    B4H_RTemp_Current = 25.00
    B5H_RTemp_Prelaunch = -182.1
    B5H_RTemp_Postlaunch = -182.1
    B5H_RTemp_Current = -182.1
    B6H_RTemp_Prelaunch = -182.2
    B6H_RTemp_Postlaunch = -182.1
    B6H_RTemp_Current = -182.2
    B7H_RTemp_Prelaunch = -182.1
    B7H_RTemp_Postlaunch = -182.1
    B7H_RTemp_Current = -182.1
    B8H_RTemp_Prelaunch = 25.00
    B8H_RTemp_Postlaunch = 25.00
    B8H_RTemp_Current = 25.00
  END_GROUP = REFERENCE_HIGH
END_GROUP = REFERENCE_TEMPERATURES
GROUP = SENSITIVITY_TEMPERATURES
  GROUP = SENSITIVITY_LOW
    B1L_SCoeff_Prelaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
    B1L_SCoeff_Postlaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
    B1L_SCoeff_Current = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0 )
```

```
   B2L_SCoeff_Prelaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
   B2L_SCoeff_Postlaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
   B2L_SCoeff_Current = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0 )
   B3L_SCoeff_Prelaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
   B3L_SCoeff_Postlaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
   B3L_SCoeff_Current = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0 )
   B4L_SCoeff_Prelaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
   B4L_SCoeff_Postlaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
   B4L_SCoeff_Current = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0 )
   B5L_SCoeff_Prelaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
   B5L_SCoeff_Postlaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
   B5L_SCoeff_Current = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0 )
   B6L_SCoeff_Prelaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 )
   B6L_SCoeff_Postlaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 )
   B6L_SCoeff_Current = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 )
   B6L_SCoeffOff_Prelaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 )
   B6L_SCoeffOff_Postlaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 )
   B6L_SCoeffOff_Current = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 )
   B7L_SCoeff_Prelaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
   B7L_SCoeff_Postlaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
   B7L_SCoeff_Current = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0 )
   B8L_SCoeff_Prelaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 )
   B8L_SCoeff_Postlaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 )
   B8L_SCoeff_Current = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 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_LOW
  GROUP = SENSITIVITY_HIGH
   B1H_SCoeff_Prelaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
   B1H_SCoeff_Postlaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
   B1H_SCoeff_Current = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0 )
   B2H_SCoeff_Prelaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
   B2H_SCoeff_Postlaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
   B2H_SCoeff_Current = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0 )
   B3H_SCoeff_Prelaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
   B3H_SCoeff_Postlaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
   B3H_SCoeff_Current = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0 )
   B4H_SCoeff_Prelaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
   B4H_SCoeff_Postlaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
   B4H_SCoeff_Current = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0 )
   B5H_SCoeff_Prelaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
```

```
    B5H_SCoeff_Postlaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
    B5H_SCoeff_Current = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0 )
    B6H_SCoeff_Prelaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 )
    B6H_SCoeff_Postlaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 )
    B6H_SCoeff_Current = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 )
    B6H_SCoeffOff_Prelaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 )
    B6H_SCoeffOff_Postlaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 )
    B6H_SCoeffOff_Current = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 )
    B7H_SCoeff_Prelaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
    B7H_SCoeff_Postlaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0 )
    B7H_SCoeff_Current = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0 )
    B8H_SCoeff_Prelaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 )
    B8H_SCoeff_Postlaunch = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 )
    B8H_SCoeff_Current = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 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_HIGH
END_GROUP = SENSITIVITY_TEMPERATURES
GROUP = LAMP_RADIANCE
  GROUP = TRENDING_COEFFS
    Lamp1_Coeffs = (+0.0,+0.0)
    Lamp2_Coeffs = (+0.0,+0.0)
  END_GROUP = TRENDING_COEFFS
  GROUP = LAMP_RADIANCE_LOW
    B1L_Rad_State1_Prelaunch =
(45.787,45.377,46.026,45.784,46.332,45.894,46.752,45.929,46.900,46.087,46.742,45.694,46.361,45.56
1,46.177,45.732)
    B1L_Rad_State1_Postlaunch =
(50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0)
    B1L_Rad_State1_Current =
(45.787,45.377,46.026,45.784,46.332,45.894,46.752,45.929,46.900,46.087,46.742,45.694,46.361,45.56
1,46.177,45.732)
    B1L_Rad_State2_Prelaunch =
(45.803,45.365,45.935,45.555,46.116,45.726,46.623,45.706,46.806,45.923,46.497,45.639,46.194,45.33
5,45.981,45.577)
    B1L_Rad_State2_Postlaunch =
(50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0)
    B1L_Rad_State2_Current =
(45.803,45.365,45.935,45.555,46.116,45.726,46.623,45.706,46.806,45.923,46.497,45.639,46.194,45.33
5,45.981,45.577)
    B1L_Rad_State3_Prelaunch =
(81.684,80.810,81.903,81.323,82.394,81.590,83.226,81.523,83.745,82.168,83.184,81.248,82.648,80.83
9,82.374,81.345)
    B1L_Rad_State3_Postlaunch =
(100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0)
    B1L_Rad_State3_Current =
(81.684,80.810,81.903,81.323,82.394,81.590,83.226,81.523,83.745,82.168,83.184,81.248,82.648,80.83
9,82.374,81.345)
    B2L_Rad_State1_Prelaunch =
(92.855,86.584,93.161,87.519,94.752,86.906,95.384,86.465,95.068,88.235,94.897,86.732,94.539,86.82
6,93.658,85.806)
    B2L_Rad_State1_Postlaunch =
(100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0)
    B2L_Rad_State1_Current =
(92.855,86.584,93.161,87.519,94.752,86.906,95.384,86.465,95.068,88.235,94.897,86.732,94.539,86.82
6,93.658,85.806)
    B2L_Rad_State2_Prelaunch =
(100.787,95.042,101.110,95.845,102.845,95.216,103.303,94.719,102.990,96.648,102.735,94.994,102.41
3,95.003,101.319,93.884)
    B2L_Rad_State2_Postlaunch =
(100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0)
    B2L_Rad_State2_Current =
(100.787,95.042,101.110,95.845,102.845,95.216,103.303,94.719,102.990,96.648,102.735,94.994,102.41
3,95.003,101.319,93.884)
```

```
      B2L_Rad_State3_Prelaunch =
(183.710,171.503,184.161,173.274,187.684,171.997,188.732,171.010,188.371,174.781,187.716,171.468,
186.974,171.568,185.010,169.558)
      B2L_Rad_State3_Postlaunch =
(200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0)
      B2L_Rad_State3_Current =
(183.710,171.503,184.161,173.274,187.684,171.997,188.732,171.010,188.371,174.781,187.716,171.468,
186.974,171.568,185.010,169.558)
      B3L_Rad_State1_Prelaunch =
(74.771,68.694,75.603,68.942,76.300,69.277,77.123,69.013,75.981,69.171,75.813,69.290,75.248,68.87
4,74.268,68.658)
      B3L_Rad_State1_Postlaunch =
(60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0)
      B3L_Rad_State1_Current =
(74.771,68.694,75.603,68.942,76.300,69.277,77.123,69.013,75.981,69.171,75.813,69.290,75.248,68.87
4,74.268,68.658)
      B3L_Rad_State2_Prelaunch =
(83.835,78.103,84.806,78.339,85.510,78.794,86.371,78.474,85.090,78.442,84.790,78.771,84.271,78.17
7,83.013,77.903)
      B3L_Rad_State2_Postlaunch =
(60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0)
      B3L_Rad_State2_Current =
(83.835,78.103,84.806,78.339,85.510,78.794,86.371,78.474,85.090,78.442,84.790,78.771,84.271,78.17
7,83.013,77.903)
      B3L_Rad_State3_Prelaunch =
(148.358,136.935,150.558,137.800,151.652,138.445,153.268,137.681,150.913,138.026,150.303,138.203,
149.181,137.271,147.210,136.848)
      B3L_Rad_State3_Postlaunch =
(120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0)
      B3L_Rad_State3_Current =
(148.358,136.935,150.558,137.800,151.652,138.445,153.268,137.681,150.913,138.026,150.303,138.203,
149.181,137.271,147.210,136.848)
      B4L_Rad_State1_Prelaunch =
(90.684,86.813,91.648,85.361,91.916,86.890,91.548,87.355,91.100,86.758,90.371,86.732,90.606,86.79
4,89.926,87.610)
      B4L_Rad_State1_Postlaunch =
(100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0)
      B4L_Rad_State1_Current =
(90.684,86.813,91.648,85.361,91.916,86.890,91.548,87.355,91.100,86.758,90.371,86.732,90.606,86.79
4,89.926,87.610)
      B4L_Rad_State2_Prelaunch =
(99.545,97.781,100.581,96.103,100.861,97.858,100.429,98.329,99.894,97.626,99.123,97.587,99.371,97
.661,98.474,98.477)
      B4L_Rad_State2_Postlaunch =
(100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0)
      B4L_Rad_State2_Current =
(99.545,97.781,100.581,96.103,100.861,97.858,100.429,98.329,99.894,97.626,99.123,97.587,99.371,97
.661,98.474,98.477)
      B4L_Rad_State3_Prelaunch =
(180.297,174.745,182.339,171.777,182.794,174.990,182.165,176.045,180.939,174.523,179.635,174.865,
180.368,174.777,178.348,176.381)
      B4L_Rad_State3_Postlaunch =
(200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0)
      B4L_Rad_State3_Current =
(180.297,174.745,182.339,171.777,182.794,174.990,182.165,176.045,180.939,174.523,179.635,174.865,
180.368,174.777,178.348,176.381)
      B5L_Rad_State1_Prelaunch =
(22.307,21.710,22.166,21.616,22.084,21.632,22.074,21.576,22.134,21.496,22.005,21.409,22.028,21.53
3,22.030,21.432)
      B5L_Rad_State1_Postlaunch =
(20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0)
      B5L_Rad_State1_Current =
(22.307,21.710,22.166,21.616,22.084,21.632,22.074,21.576,22.134,21.496,22.005,21.409,22.028,21.53
3,22.030,21.432)
      B5L_Rad_State2_Prelaunch =
(23.397,23.405,23.270,23.231,23.191,23.271,23.182,23.288,23.190,23.108,23.053,22.976,23.089,23.20
4,23.054,23.045)
      B5L_Rad_State2_Postlaunch =
(20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0)
```

```
    B5L_Rad_State2_Current =
(23.397,23.405,23.270,23.231,23.191,23.271,23.182,23.288,23.190,23.108,23.053,22.976,23.089,23.20
4,23.054,23.045)
    B5L_Rad_State3_Prelaunch =
(43.679,43.113,43.429,42.911,43.374,42.978,43.272,42.897,43.283,42.604,43.140,42.467,43.155,42.83
9,43.069,42.487)
    B5L_Rad_State3_Postlaunch =
(50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0)
    B5L_Rad_State3_Current =
(43.679,43.113,43.429,42.911,43.374,42.978,43.272,42.897,43.283,42.604,43.140,42.467,43.155,42.83
9,43.069,42.487)
    B7L_Rad_State1_Prelaunch =
(12.224,11.010,12.122,10.999,12.138,10.970,12.140,11.023,12.146,10.957,12.103,10.979,12.128,10.95
3,12.035,10.990)
    B7L_Rad_State1_Postlaunch =
(10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0)
    B7L_Rad_State1_Current =
(12.224,11.010,12.122,10.999,12.138,10.970,12.140,11.023,12.146,10.957,12.103,10.979,12.128,10.95
3,12.035,10.990)
    B7L_Rad_State2_Prelaunch =
(12.661,11.945,12.529,11.926,12.564,11.878,12.545,11.915,12.532,11.850,12.498,11.884,12.504,11.80
1,12.390,11.835)
    B7L_Rad_State2_Postlaunch =
(10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0)
    B7L_Rad_State2_Current =
(12.661,11.945,12.529,11.926,12.564,11.878,12.545,11.915,12.532,11.850,12.498,11.884,12.504,11.80
1,12.390,11.835)
    B7L_Rad_State3_Prelaunch =
(24.885,22.955,24.651,22.925,24.702,22.848,24.685,22.938,24.678,22.807,24.601,22.863,24.632,22.75
4,24.425,22.825)
    B7L_Rad_State3_Postlaunch =
(30.0,30.0,30.0,30.0,30.0,30.0,30.0,30.0,30.0,30.0,30.0,30.0,30.0,30.0,30.0,30.0)
    B7L_Rad_State3_Current =
(24.885,22.955,24.651,22.925,24.702,22.848,24.685,22.938,24.678,22.807,24.601,22.863,24.632,22.75
4,24.425,22.825)
    B8L_Rad_State1_Prelaunch =
(99.913,88.876,102.090,89.352,103.174,89.855,103.613,90.034,103.929,91.642,104.850,90.456,104.540
,91.315,104.027,90.476,105.156,89.298,105.865,89.710,105.337,90.632,105.169,90.085,104.852,90.168
,103.097,90.113,102.637,90.968,101.805,89.732)
    B8L_Rad_State1_Postlaunch =
(110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,
110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0)
    B8L_Rad_State1_Current =
(99.913,88.876,102.090,89.352,103.174,89.855,103.613,90.034,103.929,91.642,104.850,90.456,104.540
,91.315,104.027,90.476,105.156,89.298,105.865,89.710,105.337,90.632,105.169,90.085,104.852,90.168
,103.097,90.113,102.637,90.968,101.805,89.732)
    B8L_Rad_State2_Prelaunch =
(93.948,84.423,96.089,84.763,96.898,85.256,97.194,85.395,97.565,87.087,98.395,86.116,97.894,86.73
9,97.360,85.834,98.402,84.873,99.018,85.050,98.837,85.956,98.510,85.452,98.185,85.574,96.513,85.2
76,96.015,86.142,95.205,85.060)
    B8L_Rad_State2_Postlaunch =
(110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,
110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0)
    B8L_Rad_State2_Current =
(93.948,84.423,96.089,84.763,96.898,85.256,97.194,85.395,97.565,87.087,98.395,86.116,97.894,86.73
9,97.360,85.834,98.402,84.873,99.018,85.050,98.837,85.956,98.510,85.452,98.185,85.574,96.513,85.2
76,96.015,86.142,95.205,85.060)
    B8L_Rad_State3_Prelaunch =
(182.440,163.589,186.632,164.440,189.408,165.374,190.111,165.489,190.873,166.669,192.347,165.231,
192.365,165.990,191.352,165.694,193.326,164.103,194.511,164.837,193.731,165.855,193.256,165.002,1
93.115,165.292,190.031,164.790,189.484,166.400,187.450,164.442)
    B8L_Rad_State3_Postlaunch =
(220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,
220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0)
    B8L_Rad_State3_Current =
(182.440,163.589,186.632,164.440,189.408,165.374,190.111,165.489,190.873,166.669,192.347,165.231,
192.365,165.990,191.352,165.694,193.326,164.103,194.511,164.837,193.731,165.855,193.256,165.002,1
93.115,165.292,190.031,164.790,189.484,166.400,187.450,164.442)
  END_GROUP = LAMP_RADIANCE_LOW
  GROUP = LAMP_RADIANCE_HIGH
```

```
    B1H_Rad_State1_Prelaunch =
(45.787,45.377,46.026,45.784,46.332,45.894,46.752,45.929,46.900,46.087,46.742,45.694,46.361,45.56
1,46.177,45.732)
    B1H_Rad_State1_Postlaunch =
(50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0)
    B1H_Rad_State1_Current =
(45.787,45.377,46.026,45.784,46.332,45.894,46.752,45.929,46.900,46.087,46.742,45.694,46.361,45.56
1,46.177,45.732)
    B1H_Rad_State2_Prelaunch =
(45.803,45.365,45.935,45.555,46.116,45.726,46.623,45.706,46.806,45.923,46.497,45.639,46.194,45.33
5,45.981,45.577)
    B1H_Rad_State2_Postlaunch =
(50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0)
    B1H_Rad_State2_Current =
(45.803,45.365,45.935,45.555,46.116,45.726,46.623,45.706,46.806,45.923,46.497,45.639,46.194,45.33
5,45.981,45.577)
    B1H_Rad_State3_Prelaunch =
(81.684,80.810,81.903,81.323,82.394,81.590,83.226,81.523,83.745,82.168,83.184,81.248,82.648,80.83
9,82.374,81.345)
    B1H_Rad_State3_Postlaunch =
(100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0)
    B1H_Rad_State3_Current =
(81.684,80.810,81.903,81.323,82.394,81.590,83.226,81.523,83.745,82.168,83.184,81.248,82.648,80.83
9,82.374,81.345)
    B2H_Rad_State1_Prelaunch =
(92.855,86.584,93.161,87.519,94.752,86.906,95.384,86.465,95.068,88.235,94.897,86.732,94.539,86.82
6,93.658,85.806)
    B2H_Rad_State1_Postlaunch =
(100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0)
    B2H_Rad_State1_Current =
(92.855,86.584,93.161,87.519,94.752,86.906,95.384,86.465,95.068,88.235,94.897,86.732,94.539,86.82
6,93.658,85.806)
    B2H_Rad_State2_Prelaunch =
(100.787,95.042,101.110,95.845,102.845,95.216,103.303,94.719,102.990,96.648,102.735,94.994,102.41
3,95.003,101.319,93.884)
    B2H_Rad_State2_Postlaunch =
(100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0)
    B2H_Rad_State2_Current =
(100.787,95.042,101.110,95.845,102.845,95.216,103.303,94.719,102.990,96.648,102.735,94.994,102.41
3,95.003,101.319,93.884)
    B2H_Rad_State3_Prelaunch =
(183.710,171.503,184.161,173.274,187.684,171.997,188.732,171.010,188.371,174.781,187.716,171.468,
186.974,171.568,185.010,169.558)
    B2H_Rad_State3_Postlaunch =
(200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0)
    B2H_Rad_State3_Current =
(183.710,171.503,184.161,173.274,187.684,171.997,188.732,171.010,188.371,174.781,187.716,171.468,
186.974,171.568,185.010,169.558)
    B3H_Rad_State1_Prelaunch =
(74.771,68.694,75.603,68.942,76.300,69.277,77.123,69.013,75.981,69.171,75.813,69.290,75.248,68.87
4,74.268,68.658)
    B3H_Rad_State1_Postlaunch =
(60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0)
    B3H_Rad_State1_Current =
(74.771,68.694,75.603,68.942,76.300,69.277,77.123,69.013,75.981,69.171,75.813,69.290,75.248,68.87
4,74.268,68.658)
    B3H_Rad_State2_Prelaunch =
(83.835,78.103,84.806,78.339,85.510,78.794,86.371,78.474,85.090,78.442,84.790,78.771,84.271,78.17
7,83.013,77.903)
    B3H_Rad_State2_Postlaunch =
(60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0,60.0)
    B3H_Rad_State2_Current =
(83.835,78.103,84.806,78.339,85.510,78.794,86.371,78.474,85.090,78.442,84.790,78.771,84.271,78.17
7,83.013,77.903)
    B3H_Rad_State3_Prelaunch =
(148.358,136.935,150.558,137.800,151.652,138.445,153.268,137.681,150.913,138.026,150.303,138.203,
149.181,137.271,147.210,136.848)
    B3H_Rad_State3_Postlaunch =
(120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0,120.0)
```

```
    B3H_Rad_State3_Current =
(148.358,136.935,150.558,137.800,151.652,138.445,153.268,137.681,150.913,138.026,150.303,138.203,
149.181,137.271,147.210,136.848)
    B4H_Rad_State1_Prelaunch =
(90.684,86.813,91.648,85.361,91.916,86.890,91.548,87.355,91.100,86.758,90.371,86.732,90.606,86.79
4,89.926,87.610)
    B4H_Rad_State1_Postlaunch =
(100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0)
    B4H_Rad_State1_Current =
(90.684,86.813,91.648,85.361,91.916,86.890,91.548,87.355,91.100,86.758,90.371,86.732,90.606,86.79
4,89.926,87.610)
    B4H_Rad_State2_Prelaunch =
(99.545,97.781,100.581,96.103,100.861,97.858,100.429,98.329,99.894,97.626,99.123,97.587,99.371,97
.661,98.474,98.477)
    B4H_Rad_State2_Postlaunch =
(100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0,100.0)
    B4H_Rad_State2_Current =
(99.545,97.781,100.581,96.103,100.861,97.858,100.429,98.329,99.894,97.626,99.123,97.587,99.371,97
.661,98.474,98.477)
    B4H_Rad_State3_Prelaunch =
(180.297,174.745,182.339,171.777,182.794,174.990,182.165,176.045,180.939,174.523,179.635,174.865,
180.368,174.777,178.348,176.381)
    B4H_Rad_State3_Postlaunch =
(200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0,200.0)
    B4H_Rad_State3_Current =
(180.297,174.745,182.339,171.777,182.794,174.990,182.165,176.045,180.939,174.523,179.635,174.865,
180.368,174.777,178.348,176.381)
    B5H_Rad_State1_Prelaunch =
(22.307,21.710,22.166,21.616,22.084,21.632,22.074,21.576,22.134,21.496,22.005,21.409,22.028,21.53
3,22.030,21.432)
    B5H_Rad_State1_Postlaunch =
(20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0)
    B5H_Rad_State1_Current =
(22.307,21.710,22.166,21.616,22.084,21.632,22.074,21.576,22.134,21.496,22.005,21.409,22.028,21.53
3,22.030,21.432)
    B5H_Rad_State2_Prelaunch =
(23.397,23.405,23.270,23.231,23.191,23.271,23.182,23.288,23.190,23.108,23.053,22.976,23.089,23.20
4,23.054,23.045)
    B5H_Rad_State2_Postlaunch =
(20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0,20.0)
    B5H_Rad_State2_Current =
(23.397,23.405,23.270,23.231,23.191,23.271,23.182,23.288,23.190,23.108,23.053,22.976,23.089,23.20
4,23.054,23.045)
    B5H_Rad_State3_Prelaunch =
(43.679,43.113,43.429,42.911,43.374,42.978,43.272,42.897,43.283,42.604,43.140,42.467,43.155,42.83
9,43.069,42.487)
    B5H_Rad_State3_Postlaunch =
(50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0,50.0)
    B5H_Rad_State3_Current =
(43.679,43.113,43.429,42.911,43.374,42.978,43.272,42.897,43.283,42.604,43.140,42.467,43.155,42.83
9,43.069,42.487)
    B7H_Rad_State1_Prelaunch =
(12.224,11.010,12.122,10.999,12.138,10.970,12.140,11.023,12.146,10.957,12.103,10.979,12.128,10.95
3,12.035,10.990)
    B7H_Rad_State1_Postlaunch =
(10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0)
    B7H_Rad_State1_Current =
(12.224,11.010,12.122,10.999,12.138,10.970,12.140,11.023,12.146,10.957,12.103,10.979,12.128,10.95
3,12.035,10.990)
    B7H_Rad_State2_Prelaunch =
(12.661,11.945,12.529,11.926,12.564,11.878,12.545,11.915,12.532,11.850,12.498,11.884,12.504,11.80
1,12.390,11.835)
    B7H_Rad_State2_Postlaunch =
(10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0,10.0)
    B7H_Rad_State2_Current =
(12.661,11.945,12.529,11.926,12.564,11.878,12.545,11.915,12.532,11.850,12.498,11.884,12.504,11.80
1,12.390,11.835)
    B7H_Rad_State3_Prelaunch =
(24.885,22.955,24.651,22.925,24.702,22.848,24.685,22.938,24.678,22.807,24.601,22.863,24.632,22.75
4,24.425,22.825)
```

```
    B7H_Rad_State3_Postlaunch =
(30.0,30.0,30.0,30.0,30.0,30.0,30.0,30.0,30.0,30.0,30.0,30.0,30.0,30.0,30.0,30.0)
    B7H_Rad_State3_Current =
(24.885,22.955,24.651,22.925,24.702,22.848,24.685,22.938,24.678,22.807,24.601,22.863,24.632,22.75
4,24.425,22.825)
    B8H_Rad_State1_Prelaunch =
(99.913,88.876,102.090,89.352,103.174,89.855,103.613,90.034,103.929,91.642,104.850,90.456,104.540
,91.315,104.027,90.476,105.156,89.298,105.865,89.710,105.337,90.632,105.169,90.085,104.852,90.168
,103.097,90.113,102.637,90.968,101.805,89.732)
    B8H_Rad_State1_Postlaunch =
(110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,
110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0)
    B8H_Rad_State1_Current =
(99.913,88.876,102.090,89.352,103.174,89.855,103.613,90.034,103.929,91.642,104.850,90.456,104.540
,91.315,104.027,90.476,105.156,89.298,105.865,89.710,105.337,90.632,105.169,90.085,104.852,90.168
,103.097,90.113,102.637,90.968,101.805,89.732)
    B8H_Rad_State2_Prelaunch =
(93.948,84.423,96.089,84.763,96.898,85.256,97.194,85.395,97.565,87.087,98.395,86.116,97.894,86.73
9,97.360,85.834,98.402,84.873,99.018,85.050,98.837,85.956,98.510,85.452,98.185,85.574,96.513,85.2
76,96.015,86.142,95.205,85.060)
    B8H_Rad_State2_Postlaunch =
(110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,
110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0,110.0)
    B8H_Rad_State2_Current =
(93.948,84.423,96.089,84.763,96.898,85.256,97.194,85.395,97.565,87.087,98.395,86.116,97.894,86.73
9,97.360,85.834,98.402,84.873,99.018,85.050,98.837,85.956,98.510,85.452,98.185,85.574,96.513,85.2
76,96.015,86.142,95.205,85.060)
    B8H_Rad_State3_Prelaunch =
(182.440,163.589,186.632,164.440,189.408,165.374,190.111,165.489,190.873,166.669,192.347,165.231,
192.365,165.990,191.352,165.694,193.326,164.103,194.511,164.837,193.731,165.855,193.256,165.002,1
93.115,165.292,190.031,164.790,189.484,166.400,187.450,164.442)
    B8H_Rad_State3_Postlaunch =
(220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,
220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0,220.0)
    B8H_Rad_State3_Current =
(182.440,163.589,186.632,164.440,189.408,165.374,190.111,165.489,190.873,166.669,192.347,165.231,
192.365,165.990,191.352,165.694,193.326,164.103,194.511,164.837,193.731,165.855,193.256,165.002,1
93.115,165.292,190.031,164.790,189.484,166.400,187.450,164.442)
  END_GROUP = LAMP_RADIANCE_HIGH
END_GROUP = LAMP_RADIANCE
GROUP = LAMP_REFERENCE
  Lmp_Rtemp_PreLaunch = (+25.76, +25.76, +25.80, -168.6, +25.09, +25.50, +25.41,
+28.98,+28.98,+24.45,+27.35,+24.45,+23.81,+28.65)
  Lmp_Rtemp_Postlaunch = (+25.0, +25.00, +25.00, -168.6, +25.00, +25.00, +25.00,
+25.00,+25.00,+25.00,+25.00,+25.00,+25.00,+25.00)
  Lmp_Rtemp_Current = (+25.76, +25.76, +25.80, -168.6, +25.09, +25.50, +25.41,
+28.98,+28.98,+24.45,+27.35,+24.45,+23.81,+28.65)
END_GROUP = LAMP_REFERENCE
GROUP = REFLECTIVE_IC_COEFFS
  GROUP = REFLECT_IC_COEFFS_LOW
    B1L_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, 0.0, 0.0, 0.0, 0.0)
    B1L_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, 0.0, 0.0, 0.0, 0.0)
    B1L_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, 0.0, 0.0, 0.0, 0.0)
    B1L_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, 0.0, 0.0, 0.0, 0.0)
    B1L_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, 0.0, 0.0, 0.0, 0.0)
    B1L_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, 0.0, 0.0, 0.0, 0.0)
    B1L_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, 0.0, 0.0, 0.0, 0.0)
    B1L_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, 0.0, 0.0, 0.0, 0.0)
    B1L_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, 0.0, 0.0, 0.0, 0.0)
    B1L_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, 0.0, 0.0, 0.0, 0.0)
```

```
    B1L_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, 0.0, 0.0, 0.0, 0.0)
    B1L_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, 0.0, 0.0, 0.0, 0.0)
    B1L_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, 0.0, 0.0, 0.0, 0.0)
    B1L_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, 0.0, 0.0, 0.0, 0.0)
    B1L_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, 0.0, 0.0, 0.0, 0.0)
    B1L_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, 0.0, 0.0, 0.0, 0.0)
    B2L_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, 0.0, 0.0, 0.0, 0.0)
    B2L_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, 0.0, 0.0, 0.0, 0.0)
    B2L_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, 0.0, 0.0, 0.0, 0.0)
    B2L_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, 0.0, 0.0, 0.0, 0.0)
    B2L_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, 0.0, 0.0, 0.0, 0.0)
    B2L_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, 0.0, 0.0, 0.0, 0.0)
    B2L_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, 0.0, 0.0, 0.0, 0.0)
    B2L_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, 0.0, 0.0, 0.0, 0.0)
    B2L_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, 0.0, 0.0, 0.0, 0.0)
    B2L_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, 0.0, 0.0, 0.0, 0.0)
    B2L_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, 0.0, 0.0, 0.0, 0.0)
    B2L_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, 0.0, 0.0, 0.0, 0.0)
    B2L_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, 0.0, 0.0, 0.0, 0.0)
    B2L_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, 0.0, 0.0, 0.0, 0.0)
    B2L_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, 0.0, 0.0, 0.0, 0.0)
    B2L_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, 0.0, 0.0, 0.0, 0.0)
    B3L_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, 0.0, 0.0, 0.0, 0.0)
    B3L_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, 0.0, 0.0, 0.0, 0.0)
    B3L_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, 0.0, 0.0, 0.0, 0.0)
    B3L_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, 0.0, 0.0, 0.0, 0.0)
    B3L_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, 0.0, 0.0, 0.0, 0.0)
    B3L_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, 0.0, 0.0, 0.0, 0.0)
    B3L_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, 0.0, 0.0, 0.0, 0.0)
    B3L_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, 0.0, 0.0, 0.0, 0.0)
    B3L_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, 0.0, 0.0, 0.0, 0.0)
    B3L_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, 0.0, 0.0, 0.0, 0.0)
    B3L_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, 0.0, 0.0, 0.0, 0.0)
    B3L_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, 0.0, 0.0, 0.0, 0.0)
    B3L_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, 0.0, 0.0, 0.0, 0.0)
```

```
    B3L_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, 0.0, 0.0, 0.0, 0.0)
    B3L_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, 0.0, 0.0, 0.0, 0.0)
    B3L_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, 0.0, 0.0, 0.0, 0.0)
    B4L_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, 0.0, 0.0, 0.0, 0.0)
    B4L_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, 0.0, 0.0, 0.0, 0.0)
    B4L_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, 0.0, 0.0, 0.0, 0.0)
    B4L_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, 0.0, 0.0, 0.0, 0.0)
    B4L_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, 0.0, 0.0, 0.0, 0.0)
    B4L_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, 0.0, 0.0, 0.0, 0.0)
    B4L_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, 0.0, 0.0, 0.0, 0.0)
    B4L_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, 0.0, 0.0, 0.0, 0.0)
    B4L_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, 0.0, 0.0, 0.0, 0.0)
    B4L_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, 0.0, 0.0, 0.0, 0.0)
    B4L_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, 0.0, 0.0, 0.0, 0.0)
    B4L_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, 0.0, 0.0, 0.0, 0.0)
    B4L_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, 0.0, 0.0, 0.0, 0.0)
    B4L_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, 0.0, 0.0, 0.0, 0.0)
    B4L_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, 0.0, 0.0, 0.0, 0.0)
    B4L_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, 0.0, 0.0, 0.0, 0.0)
    B5L_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, 0.0, 0.0, 0.0, 0.0)
    B5L_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, 0.0, 0.0, 0.0, 0.0)
    B5L_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, 0.0, 0.0, 0.0, 0.0)
    B5L_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, 0.0, 0.0, 0.0, 0.0)
    B5L_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, 0.0, 0.0, 0.0, 0.0)
    B5L_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, 0.0, 0.0, 0.0, 0.0)
    B5L_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, 0.0, 0.0, 0.0, 0.0)
    B5L_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, 0.0, 0.0, 0.0, 0.0)
    B5L_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, 0.0, 0.0, 0.0, 0.0)
    B5L_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, 0.0, 0.0, 0.0, 0.0)
    B5L_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, 0.0, 0.0, 0.0, 0.0)
    B5L_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, 0.0, 0.0, 0.0, 0.0)
    B5L_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, 0.0, 0.0, 0.0, 0.0)
    B5L_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, 0.0, 0.0, 0.0, 0.0)
    B5L_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, 0.0, 0.0, 0.0, 0.0)
    B5L_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, 0.0, 0.0, 0.0, 0.0)
```

```
    B7L_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, 0.0, 0.0, 0.0, 0.0)
    B7L_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, 0.0, 0.0, 0.0, 0.0)
    B7L_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, 0.0, 0.0, 0.0, 0.0)
    B7L_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, 0.0, 0.0, 0.0, 0.0)
    B7L_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, 0.0, 0.0, 0.0, 0.0)
    B7L_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, 0.0, 0.0, 0.0, 0.0)
    B7L_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, 0.0, 0.0, 0.0, 0.0)
    B7L_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, 0.0, 0.0, 0.0, 0.0)
    B7L_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, 0.0, 0.0, 0.0, 0.0)
    B7L_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, 0.0, 0.0, 0.0, 0.0)
    B7L_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, 0.0, 0.0, 0.0, 0.0)
    B7L_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, 0.0, 0.0, 0.0, 0.0)
    B7L_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, 0.0, 0.0, 0.0, 0.0)
    B7L_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, 0.0, 0.0, 0.0, 0.0)
    B7L_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, 0.0, 0.0, 0.0, 0.0)
    B7L_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, 0.0, 0.0, 0.0, 0.0)
    B8L_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, 0.0, 0.0, 0.0, 0.0)
    B8L_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, 0.0, 0.0, 0.0, 0.0)
    B8L_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, 0.0, 0.0, 0.0, 0.0)
    B8L_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, 0.0, 0.0, 0.0, 0.0)
    B8L_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, 0.0, 0.0, 0.0, 0.0)
    B8L_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, 0.0, 0.0, 0.0, 0.0)
    B8L_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, 0.0, 0.0, 0.0, 0.0)
    B8L_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, 0.0, 0.0, 0.0, 0.0)
    B8L_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, 0.0, 0.0, 0.0, 0.0)
    B8L_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, 0.0, 0.0, 0.0, 0.0)
    B8L_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, 0.0, 0.0, 0.0, 0.0)
    B8L_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, 0.0, 0.0, 0.0, 0.0)
    B8L_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, 0.0, 0.0, 0.0, 0.0)
    B8L_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, 0.0, 0.0, 0.0, 0.0)
    B8L_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, 0.0, 0.0, 0.0, 0.0)
    B8L_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, 0.0, 0.0, 0.0, 0.0)
    B8L_Coefficients_Detector17 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8L_Coefficients_Detector18 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8L_Coefficients_Detector19 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
```

```
    B8L_Coefficients_Detector20 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8L_Coefficients_Detector21 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8L_Coefficients_Detector22 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8L_Coefficients_Detector23 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8L_Coefficients_Detector24 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8L_Coefficients_Detector25 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8L_Coefficients_Detector26 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8L_Coefficients_Detector27 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8L_Coefficients_Detector28 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8L_Coefficients_Detector29 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8L_Coefficients_Detector30 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8L_Coefficients_Detector31 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8L_Coefficients_Detector32 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 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 = REFLECT_IC_COEFFS_LOW
  GROUP = REFLECT_IC_COEFFS_HIGH
    B1H_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, 0.0, 0.0, 0.0, 0.0)
    B1H_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, 0.0, 0.0, 0.0, 0.0)
    B1H_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, 0.0, 0.0, 0.0, 0.0)
    B1H_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, 0.0, 0.0, 0.0, 0.0)
    B1H_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, 0.0, 0.0, 0.0, 0.0)
    B1H_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, 0.0, 0.0, 0.0, 0.0)
    B1H_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, 0.0, 0.0, 0.0, 0.0)
    B1H_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, 0.0, 0.0, 0.0, 0.0)
    B1H_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, 0.0, 0.0, 0.0, 0.0)
    B1H_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, 0.0, 0.0, 0.0, 0.0)
    B1H_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, 0.0, 0.0, 0.0, 0.0)
    B1H_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, 0.0, 0.0, 0.0, 0.0)
    B1H_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, 0.0, 0.0, 0.0, 0.0)
    B1H_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, 0.0, 0.0, 0.0, 0.0)
    B1H_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, 0.0, 0.0, 0.0, 0.0)
    B1H_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, 0.0, 0.0, 0.0, 0.0)
    B2H_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, 0.0, 0.0, 0.0, 0.0)
    B2H_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, 0.0, 0.0, 0.0, 0.0)
    B2H_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, 0.0, 0.0, 0.0, 0.0)
    B2H_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, 0.0, 0.0, 0.0, 0.0)
    B2H_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, 0.0, 0.0, 0.0, 0.0)
```

```
    B2H_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, 0.0, 0.0, 0.0, 0.0)
    B2H_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, 0.0, 0.0, 0.0, 0.0)
    B2H_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, 0.0, 0.0, 0.0, 0.0)
    B2H_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, 0.0, 0.0, 0.0, 0.0)
    B2H_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, 0.0, 0.0, 0.0, 0.0)
    B2H_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, 0.0, 0.0, 0.0, 0.0)
    B2H_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, 0.0, 0.0, 0.0, 0.0)
    B2H_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, 0.0, 0.0, 0.0, 0.0)
    B2H_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, 0.0, 0.0, 0.0, 0.0)
    B2H_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, 0.0, 0.0, 0.0, 0.0)
    B2H_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, 0.0, 0.0, 0.0, 0.0)
    B3H_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, 0.0, 0.0, 0.0, 0.0)
    B3H_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, 0.0, 0.0, 0.0, 0.0)
    B3H_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, 0.0, 0.0, 0.0, 0.0)
    B3H_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, 0.0, 0.0, 0.0, 0.0)
    B3H_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, 0.0, 0.0, 0.0, 0.0)
    B3H_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, 0.0, 0.0, 0.0, 0.0)
    B3H_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, 0.0, 0.0, 0.0, 0.0)
    B3H_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, 0.0, 0.0, 0.0, 0.0)
    B3H_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, 0.0, 0.0, 0.0, 0.0)
    B3H_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, 0.0, 0.0, 0.0, 0.0)
    B3H_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, 0.0, 0.0, 0.0, 0.0)
    B3H_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, 0.0, 0.0, 0.0, 0.0)
    B3H_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, 0.0, 0.0, 0.0, 0.0)
    B3H_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, 0.0, 0.0, 0.0, 0.0)
    B3H_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, 0.0, 0.0, 0.0, 0.0)
    B3H_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, 0.0, 0.0, 0.0, 0.0)
    B4H_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, 0.0, 0.0, 0.0, 0.0)
    B4H_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, 0.0, 0.0, 0.0, 0.0)
    B4H_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, 0.0, 0.0, 0.0, 0.0)
    B4H_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, 0.0, 0.0, 0.0, 0.0)
    B4H_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, 0.0, 0.0, 0.0, 0.0)
    B4H_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, 0.0, 0.0, 0.0, 0.0)
    B4H_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, 0.0, 0.0, 0.0, 0.0)
    B4H_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, 0.0, 0.0, 0.0, 0.0)
```

```
    B4H_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, 0.0, 0.0, 0.0, 0.0)
    B4H_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, 0.0, 0.0, 0.0, 0.0)
    B4H_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, 0.0, 0.0, 0.0, 0.0)
    B4H_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, 0.0, 0.0, 0.0, 0.0)
    B4H_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, 0.0, 0.0, 0.0, 0.0)
    B4H_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, 0.0, 0.0, 0.0, 0.0)
    B4H_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, 0.0, 0.0, 0.0, 0.0)
    B4H_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, 0.0, 0.0, 0.0, 0.0)
    B5H_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, 0.0, 0.0, 0.0, 0.0)
    B5H_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, 0.0, 0.0, 0.0, 0.0)
    B5H_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, 0.0, 0.0, 0.0, 0.0)
    B5H_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, 0.0, 0.0, 0.0, 0.0)
    B5H_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, 0.0, 0.0, 0.0, 0.0)
    B5H_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, 0.0, 0.0, 0.0, 0.0)
    B5H_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, 0.0, 0.0, 0.0, 0.0)
    B5H_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, 0.0, 0.0, 0.0, 0.0)
    B5H_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, 0.0, 0.0, 0.0, 0.0)
    B5H_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, 0.0, 0.0, 0.0, 0.0)
    B5H_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, 0.0, 0.0, 0.0, 0.0)
    B5H_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, 0.0, 0.0, 0.0, 0.0)
    B5H_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, 0.0, 0.0, 0.0, 0.0)
    B5H_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, 0.0, 0.0, 0.0, 0.0)
    B5H_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, 0.0, 0.0, 0.0, 0.0)
    B5H_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, 0.0, 0.0, 0.0, 0.0)
    B7H_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, 0.0, 0.0, 0.0, 0.0)
    B7H_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, 0.0, 0.0, 0.0, 0.0)
    B7H_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, 0.0, 0.0, 0.0, 0.0)
    B7H_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, 0.0, 0.0, 0.0, 0.0)
    B7H_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, 0.0, 0.0, 0.0, 0.0)
    B7H_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, 0.0, 0.0, 0.0, 0.0)
    B7H_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, 0.0, 0.0, 0.0, 0.0)
    B7H_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, 0.0, 0.0, 0.0, 0.0)
    B7H_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, 0.0, 0.0, 0.0, 0.0)
    B7H_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, 0.0, 0.0, 0.0, 0.0)
    B7H_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, 0.0, 0.0, 0.0, 0.0)
```

```
    B7H_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, 0.0, 0.0, 0.0, 0.0)
    B7H_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, 0.0, 0.0, 0.0, 0.0)
    B7H_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, 0.0, 0.0, 0.0, 0.0)
    B7H_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, 0.0, 0.0, 0.0, 0.0)
    B7H_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, 0.0, 0.0, 0.0, 0.0)
    B8H_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, 0.0, 0.0, 0.0, 0.0)
    B8H_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, 0.0, 0.0, 0.0, 0.0)
    B8H_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, 0.0, 0.0, 0.0, 0.0)
    B8H_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, 0.0, 0.0, 0.0, 0.0)
    B8H_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, 0.0, 0.0, 0.0, 0.0)
    B8H_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, 0.0, 0.0, 0.0, 0.0)
    B8H_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, 0.0, 0.0, 0.0, 0.0)
    B8H_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, 0.0, 0.0, 0.0, 0.0)
    B8H_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, 0.0, 0.0, 0.0, 0.0)
    B8H_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, 0.0, 0.0, 0.0, 0.0)
    B8H_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, 0.0, 0.0, 0.0, 0.0)
    B8H_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, 0.0, 0.0, 0.0, 0.0)
    B8H_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, 0.0, 0.0, 0.0, 0.0)
    B8H_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, 0.0, 0.0, 0.0, 0.0)
    B8H_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, 0.0, 0.0, 0.0, 0.0)
    B8H_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, 0.0, 0.0, 0.0, 0.0)
    B8H_Coefficients_Detector17 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8H_Coefficients_Detector18 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8H_Coefficients_Detector19 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8H_Coefficients_Detector20 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8H_Coefficients_Detector21 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8H_Coefficients_Detector22 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8H_Coefficients_Detector23 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8H_Coefficients_Detector24 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8H_Coefficients_Detector25 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8H_Coefficients_Detector26 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8H_Coefficients_Detector27 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8H_Coefficients_Detector28 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8H_Coefficients_Detector29 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8H_Coefficients_Detector30 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
```

```
    B8H_Coefficients_Detector31 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
    B8H_Coefficients_Detector32 = ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 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 = REFLECT_IC_COEFFS_HIGH
END_GROUP = REFLECTIVE_IC_COEFFS
GROUP = B6_VIEW_COEFFS
  B6_View_Coefficients_Detector1 =
(0.2055,0.1770,0.0220,0.0220,0.0066,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.708862512,+1.0,+0.986347162)
  B6_View_Coefficients_Detector2 =
(0.2120,0.1775,0.0256,0.0256,0.0077,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.711234307,+1.0,+1.000250279)
  B6_View_Coefficients_Detector3 =
(0.1990,0.1780,0.0252,0.0252,0.0075,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.709157515,+1.0,+0.988932392)
  B6_View_Coefficients_Detector4 =
(0.2070,0.1805,0.0253,0.0253,0.0076,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.711378831,+1.0,+0.998771185)
  B6_View_Coefficients_Detector5 =
(0.1875,0.1810,0.0242,0.0242,0.0072,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.710184278,+1.0,+0.98391741)
  B6_View_Coefficients_Detector6 =
(0.2075,0.1815,0.0242,0.0242,0.0072,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.710184278,+1.0,+0.995997411)
  B6_View_Coefficients_Detector7 =
(0.1955,0.1785,0.0232,0.0232,0.0069,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.710531517,+1.0,+0.98538612)
  B6_View_Coefficients_Detector8 =
(0.2230,0.1815,0.0226,0.0226,0.0068,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.706144426,+1.0,+0.997189057)
END_GROUP = B6_VIEW_COEFFS
GROUP = B6_TEMP_MODEL_COEFFS
  B6_Temp_Model_Parm = (+1.0178,+0.0,+0.0,+0.0,+0.0,+0.0)
END_GROUP = B6_TEMP_MODEL_COEFFS
GROUP = THERMISTOR_COEFFS
  Black_Body_Isolated_Temp = (16.778000,0.092912,0.00011322,0,0,0)
  Black_Body_Control_Temp = (51.724000,-0.16368,0.000071646,0,0,0)
  Cold_FP_Control_Temp = (110.350500,-0.10204,0,0,0,0)
  Cold_FP_Monitor_Temp = (109.718500,-0.10177,0,0,0,0)
  Cal_Shutter_Flag_Temp = (37.23,-0.16878,3.8161E-05,0.0,0.0,0.0)
  Backup_Shutter_Flag_Temp = (37.230000,-0.16878,0.000038161,0,0,0)
  Baffle_Heater_Temp = (-2.999300,0.093187,0.00026150,0,0,0)
  Silicon_FP_Array_Temp = (5.139200,0.086259,0.00020767,0,0,0)
  Primary_Mirror_Temp = (121.499000,-1.95685,0.0202707,-1.2745E-04,4.0681E-07,-5.2512E-10)
  Secondary_Mirror_Temp = (121.499000,-1.95685,0.0202707,-1.2745E-04,4.0681E-07,-5.2512E-10)
  Scan_Line_Corrector_Temp = (109.650000,-2.3891,0.029481,-1.9470E-04,6.2209E-07,-7.5546E-10)
  Baffle3_Tube_Temp = (121.499000,-1.95685,0.0202707,-1.2745E-04,4.0681E-07,-5.2512E-10)
  Baffle2_Support_Temp = (121.499000,-1.95685,0.0202707,-1.2745E-04,4.0681E-07,-5.2512E-10)
  Cal_Lamp_Housing_Temp = (121.499000,-1.95685,0.0202707,-1.2745E-04,4.0681E-07,-5.2512E-10)
  Cal_Shutter_Hub_Temp = (121.499000,-1.95685,0.0202707,-1.2745E-04,4.0681E-07,-5.2512E-10)
  Ambient_Preamp_HighCh_Temp = (121.499000,-1.95685,0.0202707,-1.2745E-04,4.0681E-07,-5.2512E-10)
  Ambient_Preamp_LowCh_Temp = (121.499000,-1.95685,0.0202707,-1.2745E-04,4.0681E-07,-5.2512E-10)
  Postamp_Temp_B4 = (121.499000,-1.95685,0.0202707,-1.2745E-04,4.0681E-07,-5.2512E-10)
  Cold_Preamp_B7_Temp = (121.499000,-1.95685,0.0202707,-1.2745E-04,4.0681E-07,-5.2512E-10)
  Pan_Band_Postamp_Temp = (121.499000,-1.95685,0.0202707,-1.2745E-04,4.0681E-07,-5.2512E-10)
  Telescope_Housing_Temp = (121.499000,-1.95685,0.0202707,-1.2745E-04,4.0681E-07,-5.2512E-10)
  Primary_Mirror_Mask_Temp = (121.499000,-1.95685,0.0202707,-1.2745E-04,4.0681E-07,-5.2512E-10)
  Secondary_Mirror_Mask_Temp = (121.499000,-1.95685,0.0202707,-1.2745E-04,4.0681E-07,-5.2512E-10)
  Telescope_Baseplate_Temp = (121.499000,-1.95685,0.0202707,-1.2745E-04,4.0681E-07,-5.2512E-10)
  Mem_Heat_Sink_Power_Supply1_Temp = (121.499000,-1.95685,0.0202707,-1.2745E-04,4.0681E-07,-
5.2512E-10)
  Mem_Heat_Sink_Power_Supply2_Temp = (121.499000,-1.95685,0.0202707,-1.2745E-04,4.0681E-07,-
5.2512E-10)
  Mux1_Power_Supply_Temp = (109.484000,-2.42279,0.0286100,-1.9000E-04,6.1400E-07,-7.7500E-10)
  Mux1_Electronics_Temp = (109.484000,-2.42279,0.0286100,-1.9000E-04,6.1400E-07,-7.7500E-10)
END_GROUP = THERMISTOR_COEFFS
GROUP = LAMP_CURRENTS
  Tec_Lamp_i1 = (95.449,-0.041194)
  Tec_Lamp_i2 = (95.449,-0.041194)
END_GROUP = LAMP_CURRENTS
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.

A useful ODL document is the Jet Propulsion Laboratory (JPL), California Institute of Technology's Planetary Data System Standards Reference, Version 3.2, Chapter 12. Object Description Language Specification and Usage, July 24, 1995.
This document is online at http://pds.jpl.nasa.gov .