674-SO-43, Version 3.0,
SSC-PD-4044

SIRTF Science Center

Downlink Segment

Subsystem Design Specification

AOT Products Subsystem: PREDICTSAT

26 June 2002

California Institute of Technology
SIRTF Science Center

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

THIS PAGE IS INTENTIONALLY LEFT BLANK

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

SIRTF Science Center

Subsystem Design Specification

Prepared by:

Frank Masci

Concurred by:

Bill Reach Dave Shupe

Approved by:

Mehrdad Moshir

Concurred by:

Bill Green

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

1 Revision History

Version	Description	Date
1.0	Initial version	March 6, 2001
2.0	This version includes the option of reading pixel-dependent saturation thresholds computed from "LINCAL's" cubic non-linearity model.	September 25, 2001
3.0	This version updates the input d-masks (with default bit 10) if saturation (and neighboring "pixel-growth" masking) is predicted in any of the input DCEs.	June 26, 2002

2 Table of Contents

1 REVISION HISTORY....IV
2 TABLE OF CONTENTS....V
3 LIST OF FIGURES....VII
4 LIST OF TABLES....VIII
1. INTRODUCTION....9
1.1. Purpose and Scope....9
1.2. Document Organization....9
1.3. Relationship to Other Documents....9
1.4. Change Procedure....10
2. OVERVIEW....10
2.1. PREDICTSAT Requirements....11
2.2. Applicable Documents....11
2.3. Version History....12
2.3.1. Version 1.0....12
2.3.2. Version 2.0....12
2.3.3. Version 3.0....12
2.4. Liens....12
3. INPUT....12
3.1. PREDICTSAT Input....12
3.1.1. PREDICTSAT NAMELIST Input....12
3.1.2. PREDICTSAT Command-Line Input....14
3.1.3. PREDICTSAT FITS Input....14
4. PROCESSING....15
4.1. PREDICTSAT Processing....16

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

4.2 PREDICTSAT Processing Phases 16
4.1.1. PREDICTSAT Initialization....................16
4.1.2. FITS Image List Input 17
4.1.3. LINCAL, Saturation Thresholds and Mask Inputs....................18
4.1.4. Exposure-Time Ordering....................18
4.1.5. Saturation Prediction 18
4.1.6. “Growth” Masking 19
4.1.7. Updating D-Masks....................19
4.1.8. S-Mask FITS-Image Output....................19
4.1.9. Updating of Input FITS Headers 20
4.1.10. Termination 20
5. ALGORITHM....................20
5.1. Algorithm Specifics....................20
5.2. Assumptions and Requirements....................21
6. OUTPUT 21
6.1. PREDICTSAT Output....................22
6.1.1 PREDICTSAT FITS Output 22
6.1.2 PREDICTSAT Log-File Output....................22
7. TESTING....................23
8. USAGE EXAMPLES 23
9. GLOSSARY 24

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

List of Figures

Figure 1. PREDICTSAT data and processing flow 17

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

List of Tables

Table 1. Namelist File 12

Table 2. Command Line Options....14

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

1. Introduction

1.1. Purpose and Scope

The Subsystem Design Specification is a document that describes the basic requirements, assumptions, definitions, software-design details and necessary interfaces for each subsystem. The document will be used to trace the incremental development of each subsystem and also to allow trace-back of levied requirements; this document should have sufficient detail to allow future modification or maintenance of the software by developers other than the original developers. This document is an evolving document as changes may occur in the course of science instrument hardware design and maturity of operational procedures. This document is not intended to repeat sections or chapters from other Project documents; when appropriate, references to proper sections of primary reference documents will be made.

1.2. Document Organization

This document is organized along the major themes of Requirements; Assumptions; Operational Concept; Functional Descriptions; Functional Dependencies; Input; Output; Other S/S Interfaces; Algorithm Descriptions (when applicable); and Major Liens.

The material contained in this document represent the current understanding of the capabilities of the major SIRTF systems. Areas that require further analysis are noted by TBD (To Be Determined) or TBR (To Be Resolved). TBD indicates missing data that are not yet available. TBR indicates preliminary data that are not firmly established and are subject to change.

1.3. Relationship to Other Documents

The requirements on the operation of SIRTF flow down from the Science Requirements Document (674-SN-100) and the Facility Requirements Document (674-FE-100). The Science Operations System is governed by the SOS Requirements Document (674-SO-100). The current document is also cognizant of the requirements that appear in the Observatory Performance and Interface Control Document (674-SEIT-100) as well as the Flight Ground Interface Control Document (674-FE-101). This document is also affected by the FOS/SOS Interface Control Document (674-FE-102) that governs interfaces between the Flight Operations System and the Science Operations System. Related Software Interface Specifications (SIS) will be as indicated in Section 2.2 of this document.

1.4. Change Procedure

This document is a level 4 document according to the SIRTF Project Documentation Plan (674-FE-103). Changes to this document after approval require the approval of the SOS Change Board (TBD). The process for change control is described in the SOS Configuration Management Plan.

2. Overview

PREDICTSAT reads data from a set of standard FITS images corresponding to a single sky pointing but each differing in exposure time to predict which images contain saturated pixels based on knowledge of exposure times, pixel intensities in the first time-ordered frame and a given (user-specified) saturation threshold. Although generic, this module is specifically designed to predict for saturation in IRAC's High Dynamic Range (HDR) mode by setting a saturation flag so that saturated pixels are rejected by later processing steps.

IRAC's HDR mode corresponds to the situation where an observer, in addition to a selected full-frame exposure time, receives extra frames of the same pointing taken with shorter exposure times. The reason for this is to increase the dynamic range of the resulting data where objects which are saturated in the long exposure can be replaced by data in the shorter exposures. The main relevance of PREDICTSAT to this mode of observation is due to the complex behavior of a saturating pixel in IRAC's arrays. Because IRAC uses Fowler sampling, a pixel which starts to saturate can actually have its DN decrease as the exposure time is increased leading to an erroneous DN value. The DN value can actually be driven to zero, without an observer knowing that it was "really" saturated. One would therefore like to predict in advance from a shorter exposure whether a pixel is in indeed saturated in subsequent longer frames even though it won't appear to be when directly measured.

PREDICTSAT reads in a FITS image list, either a single threshold saturation value or optionally, an array of (pixel dependent) saturation thresholds from output generated by the LINCAL module and optionally, d- and p-mask images. If saturation is predicted in any of the longer exposures in the image list, an 8-bit s-mask (saturation-mask) image will be generated which flags saturated pixels with a relevant bit-word. A bit is also set to indicate "saturation growth" by masking a 3×3 box around each saturated pixel. Furthermore, the corresponding input image in which saturated pixels are predicted has its FITS header updated with a relevant keyword to indicate than an s-mask was produced. The s-masks are written to directories where the input HDR images reside. PREDICTSAT is written in standard C.

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

2.1. PREDICTSAT Requirements

PREDICTSAT is initiated by a startup script under the control of the pipeline executive and does its required functions for a given DCE image or pre-processed DCE image; this involves performing the following tasks.

A.) Retrieve the command line parameters passed by the start up script and use them to run the program.

B.) Read in as input a list of standard FITS images, saturation thresholds and mask images.

C.) Produce as primary output s-mask images if saturated pixels are predicted. Update input D-Mask images if specified.

D.) Provide exit codes to the pipeline executive and also provides logon and logoff messages identifying the version number and write any error messages to the standard output devices.

E.) Produce a processing summary.

2.2. Applicable Documents

The following documents are relevant to the PREDICTSAT program of the AOT PRODUCTS Subsystems.

A.) The SOS Requirements Document

B.) The SOS Downlink Requirements Document

C.) The SOS Downlink Software Development Guidelines

D.) The following Software Interface Specifications (SIS)

SOSDL-SIS-PD-3000 (real*4 DCE data input)

SOSDL-SIS-PD-3001 (p- and d-mask inputs)

SOSDL-SIS-CL-3001 (LINCAL model inputs)

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

SOSDL-SIS-CL-3010 (s-mask image outputs)

2.3. Version History

2.3.1. Version 1.0

Initial version created on March 6, 2001.

2.3.2. Version 2.0

Version 2.0 created on September 25, 2001. This version includes the option of reading pixel-dependent saturation thresholds from LINCAL's cubic non-linearity model.

2.3.3. Version 3.0

Version 3.0 created on June 26, 2002. This version updates all input d-masks if saturation is predicted in corresponding input DCEs. Consequently, the same bit is set for neighboring pixels due to "growth-masking".

2.4. Liens

No major liens have been identified.

3. Input

3.1. PREDICTSAT Input

PREDICTSAT takes all of its input from either the command line or namelist file, which is set up by the startup script that is controlled by the pipeline executive or standalone. If the namelist is not specified, then all required inputs are expected from the command line. If both namelist and command-line inputs are specified, then the command-line inputs override the namelist values. Prior to reading namelist and/or command-line parameters, default values for the relevant parameters are assigned.

3.1.1. PREDICTSAT NAMELIST Input

PREDICTSAT reads the NAMELIST file whose name is passed to it by start-up script. The name of the NAMELIST is PRESATIN. The parameters that can be defined in the NAMELIST are listed in Table 1.

Namelist variable	**Description**	**Dim.**	**Type**	**Units**	**Default**
FITS_Image_List_Filename	Required filename containing list of FITS-images.	161	C	DN	Null
FITS_Lincal_Image_Filename	Optional FITS image cube from LINCAL module. Only used if Sat_Threshold = 0	161	C	DN	Null
FITS_Image_PMask_Filename	Optional p-mask FITS-image	161	C	-	Null
FITS_Image_DMask_List_Filename	Optional filename containing list of d-mask FITS-images	161	C	-	Null
Model_Flag	LINCAL non-linearity model flag: 1=Quadratic, 2=Cubic	1	I*1	-	-
PMaskMax	Optional fatal PMask data bits	1	I*2	-	0
DMaskMax	Optional fatal DMask data bits	1	I*2	-	0
DMaskSat	Optional bit to set in d-mask if saturation predicted.	1	I*2	-	1024
Sat_Threshold	Optional saturation threshold value. Must specify if no FITS_Lincal_Image_Filename	1	R*4	DN	0
Log_Filename	Optional output log filename	161	C	-	stdout

Ancillary_File_Path	Pathname where supporting source files are installed.	161	C	-	./ (current directory)

Table 1. Namelist file

The following is an example of the contents of a PRESATIN NAMELIST file that might be used, where the values specified are not necessarily realistic.

```
   &PRESATIN
  Comment = 'Generic namelist file for predictsat, default values.',
  Ancillary_File_Path = '../predictsat_v1',
  FITS_Image_List_Filename = './testing/predictsat_images.list',
  Comment = 'Specify following file if no Sat_Threshold param. below:',
  FITS_Lincal_Image_Filename = './testing/lincal_mod2.fits',
  FITS_Image_DMask_List_Filename = './testing/predictsat_dmask_images.list',
  FITS_Image_PMask_Filename = './testing/pmask_ch1.fits',
  Log_Filename = 'stdout',
  Comment = 'Lincal non-linearity model: 1 = Quadratic, 2 = Cubic’,
  Model_Flag = 1,
  Comment = 'Saturation threshold, must specify if no lincal file above’,
  Sat_Threshold = 10500,
  PMaskMax = 1024,
  DMaskMax = 2,
  DMaskSat = 1024,
 &END
```

3.1.2. PREDICTSAT Command-Line Input

Alternatively, all inputs can be specified via command line, in which case, a namelist file is not needed. Or, inputs can be provided with a hybrid of both namelist and command-line mechanisms, with the latter overriding the former. Table 2 lists the available command-line options associated with their namelist-variable counterparts, as well as other options for specifying the namelist-file name and making the standard output more verbose.

3.1.3. PREDICTSAT FITS Input

PREDICTSAT uses the FITSIO library routines to read in the FITS-formatted input data file. The routines used are: fits_open_file, fits_read_keys_lng, fits_read_keys_dbl, fits_read_img, and fits_close_file.

Command-line option	Variable
-n	Namelist_Filename
-i	FITS_Image_List_Filename
-il	FITS_Lincal_Image_Filename
-k	Model_Flag
-ip	FITS_Image_PMask_Filename
-id	FITS_Image_DMask_List_Filename
-l	Log_Filename
-a	Ancillary_File_Path
-mp	PMaskMax
-md	DMaskMax
-ms	DMaskSat
-s	Sat_Threshold
-v (verbose switch)	-
-vv (super-verbose switch)	-
-d (debug switch)	-

Table 2. Command-line options

4. Processing

4.1. PREDICTSAT Processing

PREDICTSAT begins processing by writing its name and version number to standard output (verbose mode only), and then it initializes relevant variables with defaults values, and checks that the required namelist parameters and/or command-line parameters were passed to it. If this condition is not true, then it writes a message stating which parameters are missing, recommends a look at this document, and terminates by issuing an appropriate exit code to the pipeline executive; otherwise it proceeds as follows.

If an error occurs during processing, then an error message is written to standard output, a termination-status code is written to the log file, and an exit code to the pipeline executive issued.

After processing, the program name and version number, namelist filename (if used), input, and output filenames, values of other input parameters, date and time, processing time, and a termination-status code are written a log file.

4.2 PREDICTSAT Processing Phases

PREDICTSAT operates in ten phases: initialization, FITS image list data input, lincal image model and (p-,d-) mask inputs, image exposure time-ordering, saturation predictions and flagging, growth masking, d-mask updating, s-mask results output, input FITS header update, and termination. This processing level is depicted in Figure 1.

4.1.1. PREDICTSAT Initialization

PREDICTSAT initializes itself by performing the following tasks.

A.) A message is printed to STDOUT (verbose mode only), which includes the program name and version number.

B.) If specified on the command line, the NAMELIST file is opened and read. If any errors are encountered, a message is printed, and execution aborts.

C.) The remaining command-line inputs are read and checked for correct data range, consistency, etc. If any errors are encountered, a message is printed, and execution aborts.

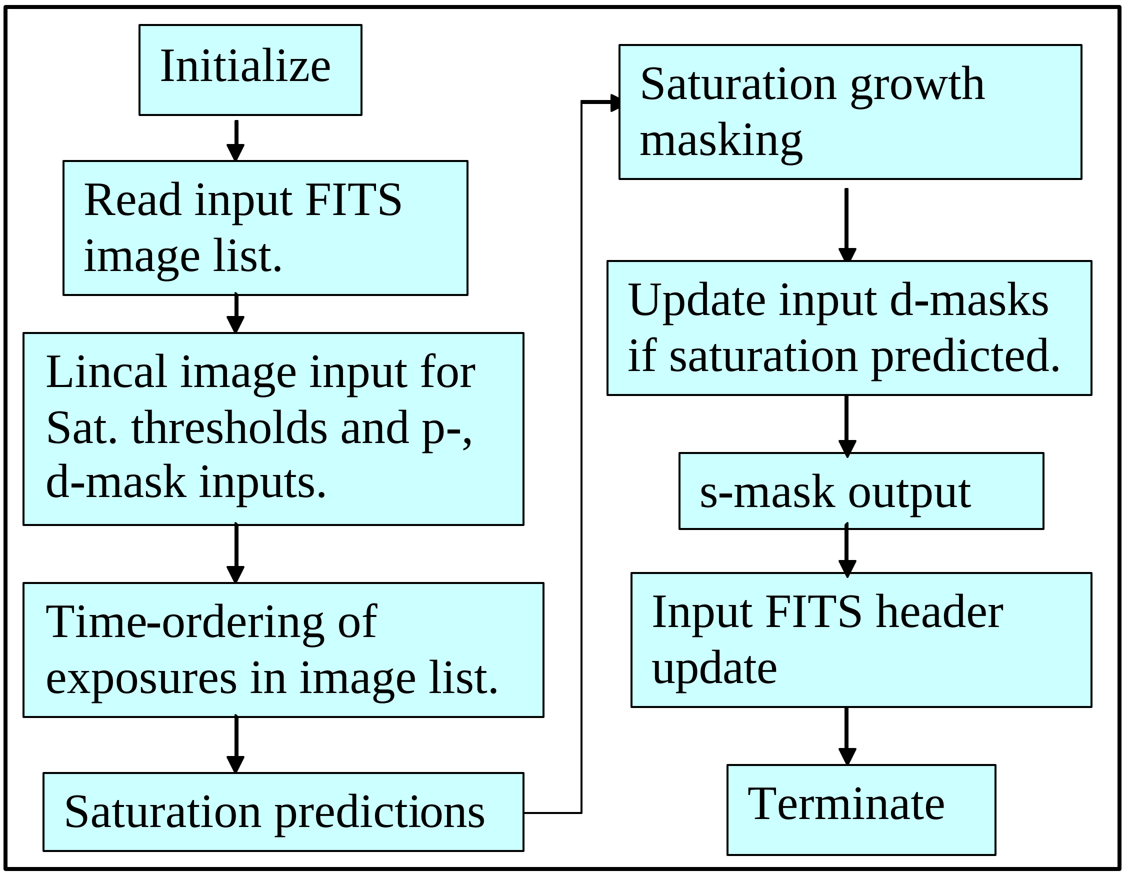

Figure 1. PREDICTSAT data and processing flow

4.1.2. FITS Image List Input

The list of FITS images (namelist parameter: FITS_Image_List_Filename) are read as standard FITS files and stored in memory. All images must be composed of a single data plane and all must

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

correspond to a single sky pointing each differing in length of exposure time. The images need not be listed in order of increasing exposure-time (see below).

4.1.3. LINCAL, Saturation Thresholds and Mask Inputs

The saturation threshold can be read in as a single user-specified parameter (Sat_Threshold) which will apply to all pixels, or it can be read from LINCAL's model estimate for each pixel independently. PREDICTSAT is set up to read output from both LINCAL's "quadratic" and "cubic" non-linearity model. If Model_Flag=1, in which case a quadratic non-linearity model is expected, the *second plane* in LINCAL's output contains the saturation threshold values. If however Model_Flag=2, a cubic non-linearity model is expected and this is read from LINCAL's *fourth plane* image output.

PREDICTSAT also optionally reads in p-mask (pixel) and d-mask (dce) images. There is one d-mask for each input DCE image and these are listed in the same order in a separate file (namelist parameter: FITS_Image_DMask_List_Filename). These are stored in memory and used in the processing stage (see below).

4.1.4. Exposure-Time Ordering

Each image read from the input list are ordered from the shortest exposure time to the longest and stored in memory. The image with the shortest exposure becomes the "base" image from which given a knowledge of its pixel intensities and exposures of subsequent images, is used to predict whether pixels in the longer exposures are saturated.

Because the IRAC arrays use Fowler sampling, the exposure time which is proportional to the "measured" DN in the well is not specified by a single quantity but is the sum of two quantities: the "wait-period" (FITS header keyword AWAITPER) and the "Fowler-number" (FITS header keyword AFOWLNUM). The exposure time is proportional to (AWAITPER + AFOWLNUM) and images are ordered according to this quantity.

4.1.5. Saturation Prediction

Let us consider a single pixel with intensity in the (first) shortest exposure-time image as $DN(1)$, where the "1" refers to image number one, and let us define the wait-period and Fowler-number keyword parameters of this image as $W(1)$ and $F(1)$ respectively. Furthermore, let us define these same quantities in a subsequent "longer" exposure-time image n as $DN(n)$, $W(n)$ and $F(n)$. The exposure-time is proportional to ($F + W$), which relates to the total measured DN. The "frame time" is proportional to ($2F + W$) which relates time to the "number of real" DN in the well. Given a nominal saturation value DN_{sat} , the same pixel in image n (and hence images n+1, n+2 etc…) will be *saturated* if the following condition will hold in image 1:

$$DN(1) \geq \left[\frac{F(1) + W(1)}{2F(n) + W(n)} \right] DN_{sat}$$

This condition is searched for using every pixel in image 1 and the wait-periods and Fowler-number for each image n. If this condition is satisfied, an s-mask is generated for image n and a relevant bit-word set therein (see section 4.1.7).

4.1.6. "Growth" Masking

Each pixel predicted to saturate in all subsequent images have their surrounding pixels masked by a 3×3 box in the s-mask as well. These are flagged by setting a different bit-word. This is due to effects of charge "growth" (bleed) from the saturated pixel into neighboring pixels.

4.1.7. Updating D-Masks

If saturation (and growth masking) is predicted in any of the input DCEs, the accompanying d-mask is updated to indicate this. The current default is bit #10 (value 1024). The bit setting can be controlled via the namelist/command-line parameter "DMaskSat".

4.1.8. S-Mask FITS-Image Output

If saturation is predicted in any of the longer expsoure images, a corresponding s-mask image is generated. This is an 8-bit/pixel bit-mask image with the following bit-word definitions:

Bit 0 (integer value 1): Pixel is predicted to be saturated

Bit 1 (integer value 2): Pixel masked due to saturation growth

Bit 2 (integer value 4): Pixel masked due to flagging from d-mask (e.g. radhit)

Bit 3 (integer value 8): Pixel masked due to flagging from p-mask

The remaining four bits are currently unassigned. Each s-mask image that is generated is named in the format: “*_smask.fits”, where * refers to the input image filename in which saturation is predicted. The images are output to the directories where the input images reside.

4.1.9. Updating of Input FITS Headers

In addition to producing an s-mask for each image where saturated pixels are predicted, the program also updates the FITS header of the input image to indicate that an s-mask was produced. The following keyword string is appended at the end of the input FITS header:

```
SMASKFIL = ‘*_smask.fits’ / S-Mask Filename
```

Where “*” refers to the input image name.

4.1.10.Termination

Summary output is appended to the log file (the log file is created if previously non-existent), which includes diagnostic reports for the Q/A Subsystem and the appropriate exit code issued to be picked up by the pipeline executive. A detailed list of log file contents is given in Section 6.1.2.

5. Algorithm

5.1. Algorithm Specifics

The simple algorithm employed in this software has been adequately described in the previous section. As a detail, “fatal” pixels defined in the p-mask are handled by replacing them by NaNs in the input images. Accordingly, all pixels that are read as NaNs from the input images are reported in the s-mask by setting different bits for that pixel depending on whether they originated from a flag in the d-mask or p-mask (see above).

5.2. Assumptions and Requirements

A. PREDICTSAT assumes that each image in the input list has a *single plane*, in other words with the standard FITS keyword values: NAXIS = 2 or NAXIS3 = 1.

B. If one chooses to use pixel-dependent saturation thresholds computed by LINCAL via the namelist parameter **FITS_Lincal_Image_Filename,** then the model to which this corresponds to must also be specified. This is specified via the namelist parameter **Model_Flag**. If Model_Flag=1 (quadratic model), PREDICTSAT assumes that the thresholds from LINCAL's image cube are *contained in plane 2*, if Model_Flag=2 (cubic model), *plane 4* is expected to contain the thresholds.

C. Only one of the following saturation threshold namelist parameters must by specified: **FITS_Lincal_Image_Filename**, or the single value **Sat_Threshold** (see Table 1). If both are specified, then the Sat_Threshold value is used. Otherwise, if one wants to use the pixel dependent values from LINCAL, then the Sat_Threshold parameter must either be set to zero, or, excluded from the namelist or command-line.

D. Each FITS image in the input list (namelist parameter: FITS_Image_List_Filename) is listed one per line and does not have to be in any specific order. Time-ordering is done internally.

E. The list of d-mask images (namelist parameter: FITS_Image_DMask_List_Filename) must directly correspond, in the same order to the images listed in the input list (from D).

F. PREDICTSAT requires that the following keywords be present in the FITS-headers of the input images: the wait-period keyword, AWAITPER and the Fowler number keyword, AFOWLNUM. If these are not present, the program will abort indicating that these are missing.

G. If saturation is predicted in any input frame, the corresponding s-mask is written to the same directory where the input frame resides, named " *_smask.fits".

6. Output

6.1. PREDICTSAT Output

PREDICTSAT is capable of generating the following output:

A.) Standard-output processing and status messages.

B.) An 8-bit FITS bit-mask image for each input image if saturated pixels are predicted therein with filename *_smask.fits, where * is the input FITS image name.

C.) Input D-Masks are updated with a relevant bit if saturation (and neighboring growth-masking) is predicted in any input DCE.

D.) A new keyword string (keyword: SMASKFIL) in the FITS header of the input image if saturated pixels are predicted therein.

E.) A log file containing processing statistics, status messages and ancillary information.

All PREDICTSAT disk output is written to the pathnames that are specified with the output filenames in the command-line or namelist inputs.

6.1.1 PREDICTSAT FITS Output

PREDICTSAT uses the FITSIO library routines to create FITS-formatted output data files. The routines used are: fits_read_key_lng, fits_insert_key_lng, fits_create_file, fits_open_file, fits_copy_hdu, fits_flush_file, fits_write_key, fits_update_key, fits_write_date, fits_write_key_str, fits_write_key_fixflt, fits_write_img, fits_get_hdrspace, fits_read_record, fits_write_record, and fits_close_file.

6.1.2 PREDICTSAT Log-File Output

The information stored in the log file at the output of this program includes: program name and version number, values of all namelist and/or command-line inputs, a message indicating the type of calculation performed, status code, processing time, date and time, and a message indicating program termination.

7. Testing

PREDICTSAT has been successfully unit-tested as a stand-alone program for a variety of different input cases. The tests were designed to check PREDICTSAT robustness and capability of generating corrected results.

Here is a summary of the unit tests that were conducted:

1. Tested PREDICTSAT on a simulated list of IRAC images taken in HDR-mode (see Section 2) with relevant FITS headers updated to form a time-ordered list of exposures.

2. Executed PREDICTSAT with inputs read from and output written to directories different from where the program was run. Both namelist and command-line input mechanisms were exercised.

3. Executed PREDICTSAT with input p- and d-mask images defined with different fatal bit-words, (e.g. cosmic radiation hits, hot/dead pixels) to test that they integrate properly with the output s-masks produced.

4. Executed PREDICTSAT with different single saturation thresholds and also LINCAL saturation image planes to test that the algorithm flags saturated pixels appropriately.

5. Executed PREDICTSAT for all combinations of input parameters, in order to test that they function properly.

6. Executed PREDICTSAT on non-square images.

8. Usage Examples

Using a namelist file with verbose (-v) output re-directed to a file “out.log”:

```
PREDICTSAT –n predictsat.nl -v | & tee out.log
```

Without using a namelist file:

```
PREDICTSAT –i image_list.txt –il lincal.fits –k 2 -a
../ancpath  –ip  pmask.fits –id dmask_list.txt –mp 1024 –md
255 –ms 1024 -v
```

9. Glossary

DCE	Data Collection Event
DN	Data Number
IOC	In-Orbit Checkout
SDS	Subsystem Design Specification
SIS	Software Interface Specification
TBD	To Be Determined
TBR	To Be Resolved