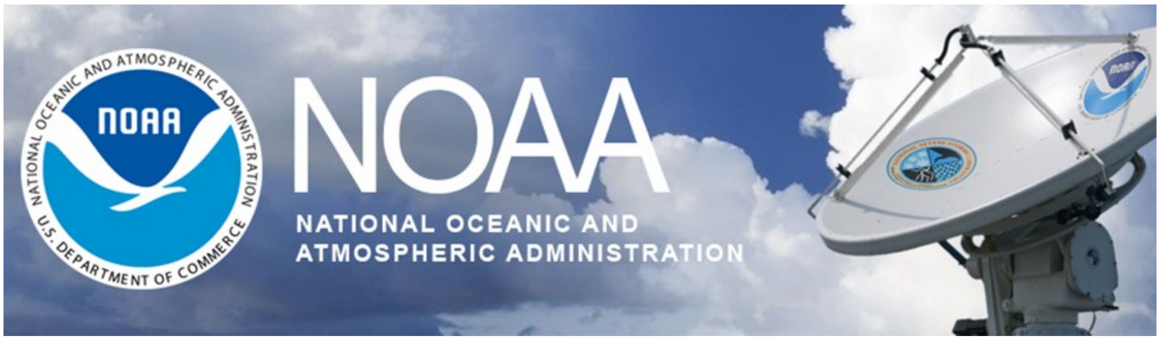
Office of Satellite and Product Operations
Environmental Satellite and Product Services



VIIRS Surface Reflectance (SR)
System Maintenance Manual (SMM)

Version 1.1
November 02, 2023

Prepared by: ERT Inc.

U.S. Department of Commerce
National Oceanic and Atmospheric Administration
National Environmental Satellite, Data, and Information Service
Office of Satellite Products and Operations

Approval Page

Environmental Satellite and Products Services

The Regional Advanced Baseline Imager and Visible Infrared Imaging Radiometer Suite Emissions (RAVE)

GROUP: ESPC Date Hanjun Ding (OSPO), SR Product Area Lead	GROUP: OMS Date Nageswarrao Gopanapalli, Project Manager
GROUP: Date	

Changes/Revisions Record

This document is changed as required to reflect system, operational, or organizational changes. Modifications made to this document are recorded in the Changes/Revisions Record below. This record will be maintained throughout the life of the document.

DOCUMENT TITLE: VIIRS SR System Maintenance Manual			
DOCUMENT CHANGE HISTORY			
Revision No.	**Date**	**Revision Originator Project Group**	**CCR Approval # and Date**
1.0	July 2023	CCAP initial version	N/A
1.1	11/02/2023	NCCF Implement Updates	NCCFPVPM-5

LIST OF CHANGES

Significant alterations made to this document are annotated in the List of Changes table.

DOCUMENT TITLE: GAASP System Maintenance Manual					
LIST OF CHANGE-AFFECTED PAGES/SECTIONS/APPENDICES					
Version Number	**Date**	**Changed By**	**Page**	**Section**	**Description of Change(s)**
1.1	11/02/2023	H. Ding Y. Zhu		ALL	NCCF Implementation updates

TABLE OF CONTENTS

Page

LIST OF TABLES AND FIGURES 7
LIST OF ACRONYMS 8
EXECUTIVE SUMMARY 10
1. INTRODUCTION 12
1.1 Product Overview 12
1.2 Algorithm Overview 13
1.3 Interfaces Overview 14
2. HARDWARE 14
2.1 Hardware Description 14
2.2 Operating System 14
2.3 System Requirements 14
2.3.1 Storage Requirements 14
2.3.2 Computer Resource Requirements .... **Error! Bookmark not defined.**
2.3.3 Communication Needs .... **Error! Bookmark not defined.**
3. SOFTWARE 17
3.1 Software Description 17
3.2 Directory Description 18
3.3 Source Code Description 25
4. NORMAL OPERATIONS 27
4.1 System Control 27
4.1.1 System Control Files 27
4.1.2 Processing Controls 27
4.2 Installation 30
4.2.1 Installation items 30
4.2.2 Compilation Procedures 31
4.2.3 Installation Procedures 31
4.3 Configuration Procedures 32
4.3.1 Production Rules 32
4.4 Operations Procedures 38
4.4.1 Normal Operations 38

4.4.2 Data Preparation ... 38
4.5 Distribution ... 39
4.5.1 Data Transfer / Communications ... 39
4.5.2 Distribution Restrictions ... 39
4.5.3 Product Retention Requirements ... 39
4.5.4 External Product Tools ... 39
5. MONITORING AND MAINTENANCE ... 40
5.1 Job Monitoring ... 40
5.2 Data Signal Monitoring ... 40
5.3 Product Monitoring ... 40
5.3.1 Unit Test Plans ... **Error! Bookmark not defined.**
5.3.2 Internal Product Tools ... 40
5.3.3 Performance Statistics ... 40
5.3.4 Product Monitoring ... 41
5.3.5 Product Criticality ... **Error! Bookmark not defined.**
5.4 Maintenance ... 41
5.4.1 Monitoring ... 41
5.4.2 Science Maintenance ... 41
5.4.3 Library Maintenance ... 42
5.4.4 Special Maintenance Procedures ... 42
5.4.5 Maintenance Utilities ... 42
5.5 Program Backup Procedures ... 42
6. TROUBLESHOOTING ... 43
6.1 Problem Diagnosis and Recovery ... 43
6.1.1 Quality Control Output ... 43
6.1.2 Error Correction ... 48
6.1.3 Problem Diagnosis and Recovery Procedures ... 48
6.1.4 Data Recovery Procedures ... 49
6.1.5 Program Recovery Procedures ... 49
6.2 Application Shutdown and Restart ... 49
6.2.1 Application Shutdown Procedures ... 49
6.2.2 Application Restart Procedures ... 49
6.3 System Shutdown and Restart ... 50

6.3.1 System Shutdown Procedures 50
6.3.2 System Restart Procedures 50
6.3.3 System Reboot Procedures 50
7. APPENDIX 51
7.1 Data Flow 51
7.2 Input Data Files 51
7.3 Ancillary Data Files 52
7.4 Look Up Tables 55
7.5 Intermediate Data Set Description 55
7.6 Output Data Set Description 55
7.7 Archive Data Description 56
8. LIST OF REFERENCES 56

LIST OF TABLES AND FIGURES

Page

Table 0-1 – Product Team Information 10
Table 2-1 – VIIRS Surface Reflectance Storage Requirements (Granule Files) 14
Table 2-2 – Input Data Sizing Per Provider Per Day 15
Table 2-3 – Output Products Sizing Per User Per Day 15
Table 6-1 - QF1 Surface Reflectance 43
Table 6-2 - QF2 Surface Reflectance 44
Table 6-3 - QF3 Surface Reflectance 44
Table 6-4 - QF4 Surface Reflectance 45
Table 6-5 - QF5 Surface Reflectance 46
Table 6-6 - QF6 Surface Reflectance 46
Table 6-7 - QF7 Surface Reflectance 47
Table 7-1 - Satellite Data Input Files 51
Table 7-2 - Dynamic Ancillary Data Files 53
Table 7-3 – Static Ancillary Data Files 54
Table 7-4 - VIIRS Surface Reflectance Output File 55

LIST OF ACRONYMS

AMSR2	Advanced Microwave Scanning Radiometer 2
AOD	Aerosol Optical Depth
AOT	Aerosol Optical Thickness
ATBD	Algorithm Theoretical Basis Document
AWS	Amazon Web Services
BRDF	Bidirectional Reflectance Distribution Function
CLASS	NOAA Comprehensive Large Array-data Stewardship System
COTS	Commercial Off-The-Shelf
ECM	Enterprise Cloud Mask
EDR	Environmental Data Record
ESPC	Earth System Prediction Capability
EUM	External User’s Manual
GAASP	GCOM-W1 AMSR-2 Algorithm Software Processor
GCOM-W1	Global Climate Observation Mission – Water 1
GPDS	GCOM Production and Distribution System
GFS	Global Forecast System
GITCO	VIIRS Geolocation data (Imager)
GMTCO	VIIRS Geolocation data (Moderate Resolution)
GSFC	Goddard Space Flight Center
HDF5	Hierarchical Data Format version 5
I-Band	Imaging Resolution bands of VIIRS
IDPS	Interface Data Processing Segment
IP	Intermediate Product
JPSS	Joint Polar Satellite System
LQF	Land Quality Flag
M-Band	Moderate Resolution bands of VIIRS
MODIS	Moderate-resolution Imaging Spectroradiometer
NCEI	National Centers for Environmental Information
NCCF	NESDIS Common Cloud Framework
NDE	NPOESS Data Exploitation
NDVI	Normalized Difference Vegetation Index
NESDIS	National Environmental Satellite, Data, and Information Service
netCDF4	network Common Data Format version 4
NOAA	National Oceanic and Atmospheric Administration
NPOESS	National Polar-orbiting Operation Environmental Satellite System
NPP	Suomi National Polar-orbiting Partnership
NSOF	NOAA Satellite Operations Facility
OMPS	Ozone Mapping Profiling Suite
OSD	Office of Systems Development
OSGS	Office of Satellite Ground Services

OSPO	Office of Satellite and Product Operations
PAL	Product Area Lead
PCF	Process Control File
PDA	Product Distribution Area
PSF	Production Status File
QC	Quality Control
QF	Quality Flag
QI	Quality Information
RAD	Requirements Allocation Document
SDR	Sensor Data Record
SMM	System Maintenance Manual
STAR	center for SaTellite Application and Research
SR	Surface Reflectance
V&V	Verification and Validation
VCM	VIIRS Cloud Mask
VIIRS	Visible Infrared Imaging Radiometer Suite

EXECUTIVE SUMMARY

The Surface Reflectance NetCDF4 product files will contain surface reflectance data for each M-band and I-band used as input to the algorithm package. For the M-bands, the resolution will be 750 meters. For the I-bands, the resolution will be 375 meters. The amount of data within the I-Band product files will be roughly four times larger than the amount of data within the M-Band product files as the I-Bands have a higher swath resolution.

This algorithm has been modified for the NCCF (NESDIS Common Cloud Framework) environment through assistance from the OSGS (Office of Satellite Ground Services). The NCCF production environment is delivered to the Office of Satellite and Product Operations (OSPO) where it is run operationally.

All of the VIIRS Surface Reflectance requirements are available through the VIIRS Surface Reflectance Requirements Allocation Document (RAD). These requirements identify the users and their needs with respect to file content, format, latency, and quality. This document is available upon request from the Surface Reflectance Product Area Lead (PAL), current as of this document in the table below.

Table 0-1 – Product Team Information

Team Member	Organization	Role	Contact Information
Hanjun Ding	OSPO	PAL (Product Area Lead)	hanjun.ding@noaa.gov
Walter Wolf	OSGS	Product Portfolio Management Lead	walter.wolf@noaa.gov
Shanna Sampson	OSGS	ASSISTT Solutions Architect	shanna.sampson@noaa.gov
Priyanka Roy	OSGS	Quality Assurance Lead, Deputy Project Manager	Priyanka.roy@noaa.gov
Tracey Dorian	OSGS	QA Deputy	tracey.dorian@noaa.gov
Edrees Wardak	OSGS	ASSISTT Quality Assurance	edrees.wardak@noaa.gov
Yunhui Zhao	OSGS	ASSISTT CM Lead & Special Projects	yunhui.zhao@noaa.gov
Yunyue Yu	STAR	Science Team	yunyue.yu@noaa.gov
Heshun Wang	STAR	Science Team	heshun.wang@noaa.gov
Eric Vermote	NASA	Science Team	eric.f.vermote@nasa.gov
Mike Wilson	OSGS	ASSISTT GitLab Merge Lead	michael.wilson@noaa.gov

Douglas March	OSGS	ASSISTT Integration Team	douglas.march@noaa.gov
Letitia Soulliard	OSGS	ASSISTT Standalone Integration Team Lead	letitia.soulliard@noaa.gov
Hua Xie	OSGS	ASSISTT Science QA Lead	hua.xie@noaa.gov
John Lindeman	OSGS	ASSISTT Technical Writer	john.lindeman@noaa.gov
Brandon Laufer	OSGS	Python Developer	brandon.laufer@noaa.gov
Eric Buzan	OSGS	ASSISTT R2O & Processing	eric.buzan@noaa.gov
Ramaswamy Tiruchirapalli	OSGS	ASSISTT Framework Integration Team	Ramaswamy.Tiruchirapalli@noaa.gov
Edward Borders	OSGS	ASSISTT Python Developer Team Lead	Edward.borders@noaa.gov
Yufeng Zhu	OSPO	OSPO Team Member	Yufeng.zhu@noaa.gov

Feedback, comments, criticisms, and/or suggestions are welcome and should be sent to:

Name	Email
Hanjun Ding	Hanjun.ding@noaa.gov
Ramaswamy Tiruchirapalli	ramaswamy.tiruchirapalli@noaa.gov
Brandon Laufer	brandon.laufer@noaa.gov
Eric Buzan	eric.buzan@noaa.gov
Ed Borders	edward.borders@noaa.gov
Tracey Dorian	tracey.dorian@noaa.gov
Eddy Wardak	edrees.wardak@noaa.gov

Priyanka Roy	priyanka.roy@noaa.gov

All of the software is written in FORTRAN 90, C, and Python, and is described in more detail in section 3. The main inputs to the VIIRS Surface Reflectance algorithm are the VIIRS SDR (Sensor Data Record) and Geolocation files in HDF-5 format, and the Enterprise Cloud Mask, Cloud Height, Aerosol Optical Depth, and Global Forecast System (GFS) data in NetCDF-4 format. The surface reflectance file is output in NetCDF format. More information about the inputs and outputs can be found in section 7.

The VIIRS Surface Reflectance product is based on heritage from the Moderate-resolution Imaging Spectroradiometer (MODIS) Surface Reflectance product. The details of the Surface Reflectance algorithm are described in the Surface Reflectance Algorithm Theoretical Basis Document (ATBD). This document can be made available by either the GCOM-W1 AMSR-2 Algorithm Software Processor (GAASP) Product Area Lead (PAL) or STAR Algorithm Lead.

The Surface Reflectance output with every VIIRS granule occurs roughly every 85 seconds. This product contains the surface reflectance for each input M-band and I-band. For the M-bands, the resolution is presented at 750 meters, while for the I-bands, the resolution is 375 meters (corresponding to the swath resolution of these bands).

Generation of the Surface Reflectance output depends on the availability of the VIIRS Sensor Data Records (SDRs) and Geolocation data as well as cloud, aerosol, and model products.

1. INTRODUCTION

1.1 Product Overview

The VIIRS Surface Reflectance product is generated from a combination of the VIIRS SDR and geolocations data, and Global Forecast System (GFS), aerosol, cloud mask, and cloud height products. The output products include surface reflectance for channels I1, I2, I3, M1, M2, M3, M4, M5, M7, M8, M10, and M11. Output also includes seven bytes of quality flags (QF1, QF2, QF3, QF4, QF5, QF6, QF7), which are each bitmasks representing various properties of both inputs and outputs. Statistical quality information is also included in the output file. All products are output files in netCDF-4. For information on the science algorithms, see the VIIRS Surface Reflectance Algorithm Theoretical Basic Document (ATBD). This document can be obtained from the Product Area Lead (PAL) identified in the Product Team table in the Executive Summary section.

The Surface Reflectance NetCDF4 product files will contain surface reflectance data for each M-band and I-band used as input to the algorithm package. For the M-bands, the resolution will be 750 meters. For the I-bands, the resolution will be 375 meters. The amount of data within the I-Band product files will be roughly four times larger than the amount of data within the M-Band product files as the I-Bands have a higher swath resolution.

1.2 Algorithm Overview

The Surface Reflectance EDR (Environmental Data Record) algorithm is designed to contain four main subroutines: Extract inputs, Quality Flags, Surface Reflectance Retrieval and Write Surface Reflectance EDR. The Surface Reflectance Retrieval subroutine is the main subroutine since it performs the lambertian approximation (atmospheric correction), the adjacency adjustment, and the bidirectional reflectance distribution function (BRDF) coupling adjustment.

The Surface Reflectance Retrieval routine corrects for the effects of gaseous absorption, molecular and aerosol scattering, thin cirrus contamination, glare from surrounding surface pixels (adjacency adjustment), and the coupling of the atmosphere and the surface bidirectional reflectance as a function of the viewing and solar geometries, elevation of the target and spectral band. The atmospheric adjustment (within the 'Surface Reflectance Retrieval' routine) includes updating the correction coefficients with 'in-view' total column water vapor, ozone, and aerosol optical thickness data input fields. The aerosol information required for surface reflectance retrieval comes from the VIIRS Aerosol Optical Thickness (AOT) EDR and the Aerosol Model Information IP, complemented by total column water vapor, total column ozone, and surface pressure from National Centers for Environmental Prediction (NCEP) feeds. Backups for these inputs include total column ozone from the Ozone Mapping Profiling Suite (OMPS). The atmospherically corrected surface reflectance values derived using the Lambertian approximation are subsequently adjusted for adjacency and bi-directional reflectance distribution function (BRDF) effects. The BRDF-coupling adjustment is presently designed after the MODIS approach with a slight modification making the isotropic shape parameter a function of the normalized difference vegetation index, NDVI, an approximation due to operational constraints, and the developmental maturity of the MODIS BRDF-coupling adjustment routine. The surface reflectance values after each adjustment are included as data layers in the surface reflectance EDR along with the Land Quality Flags. The atmospheric inputs are available elsewhere, and the solar and viewing geometries are kept as part of the SDR.

Thin cirrus effects are removed by implementing an empirically based correction using VIIRS band M9 (1.38 µm). The quality control (QC) flags generated from the Build SDR module and the cloud and aerosol quality flag inputs are fused into a single Land Quality Flag (LQF) structure that applies to the Surface Reflectance EDR, the Surface Albedo EDR, the Vegetation Index EDR, the Surface Type EDR. The LQF output is appended to the Surface Reflectance EDR. Then the heart of the surface reflectance retrieval process begins, by converting the cirrus effects adjusted satellite reflectance values into surface reflectance values assuming the surface is Lambertian. The earth's surface is generally not Lambertian, and as a result a further correction is applied. The conversion of the at-satellite-reflectance values to surface reflectance requires (i) the use of a set of conversion equations that also account for first order atmospheric multiple scattering effects, and (ii) inputs from pre- generated look up tables (LUTs) and analytic equations for gaseous or molecular effect.

1.3 Interfaces Overview

Please reference below link (Document Object: 4 and 5):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7lsXg/edit?pli=1#heading=h.uqntmud8purc

2. HARDWARE

2.1 Hardware Description

Please reference to the NCCF system description in below link (Document Object: 4 and 5):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7lsXg/edit

2.2 Operating System

The operating system description can be found in below link (Document Object: 64 and 65):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7lsXg/edit#heading=h.74yw713gbud9

2.3 System Requirements

2.3.1 Storage Requirements

Table 2-1 contains a list of storage requirements for the VIIRS Surface Reflectance system as needed during processing, including all input, intermediate, output, and ancillary data.

Table 2-1 – VIIRS Surface Reflectance Storage Requirements (Granule Files)

Storage Item	Size
Data Product Files (public output)	209 MB / granule (approx 212 GB / day)
Intermediate Files	517 MB / granule (approx 524 GB / day)
Incoming IDPS Data	293 MB / granule

	(approx 300 GB / day)
Incoming NCCF Data	383 MB / granule (approx 388 GB / day)
Total (approx 1014 granules / day)	1.4 GB / granule (approx 1.42 TB / day)

Unpacking and installing the delivery tar files will require about 11 GB of storage space.

Table 2-2 shows typical data volumes to be transferred. Table 2-3 shows the data volumes of the output products. These tables should be used as a guide for determining the disk space and bandwidth.

Table 2-2 – Input Data Sizing Per Provider Per Day

Product	Number of Files/Day	Size/Day
VIIRS SDR Imagery Band 1 (SVI01)	1014 files / day	28.4 GB / day
VIIRS SDR Imagery Band 2 (SVI02)	1014 files / day	28.4 GB / day
VIIRS SDR Imagery Band 3 (SVI03)	1014 files / day	28.4 GB / day
VIIRS SDR Moderate Band 1 (SVM01)	1014 files / day	10 GB / day
VIIRS SDR Moderate Band 2 (SVM02)	1014 files / day	10 GB / day
VIIRS SDR Moderate Band 3 (SVM03)	1014 files / day	10 GB / day
VIIRS SDR Moderate Band 4 (SVM04)	1014 files / day	10 GB / day
VIIRS SDR Moderate Band 5 (SVM05)	1014 files / day	10 GB / day
VIIRS SDR Moderate Band 7 (SVM07)	1014 files / day	10 GB / day
VIIRS SDR Moderate Band 8 (SVM08)	1014 files / day	10 GB / day
VIIRS SDR Moderate Band 10 (SVM10)	1014 files / day	10 GB / day
VIIRS SDR Moderate Band 11 (SVM11)	1014 files / day	10 GB / day
VIIRS SDR Imagery Geolocation (GITCO)	1014 files / day	114 GB / day
VIIRS SDR Moderate Geolocation (GMTCO)	1014 files / day	32 GB / day

Table 2-3 – Output Products Sizing Per User Per Day

Product	Number of Files/Day	Size/Day
Surface Reflectance	1014	212 GB / day

2.3.2 Computer Resource Requirements

VIIRS Surface Reflectance requires the following standard libraries (included with the Docker image):

- Relevant libraries and tools available on the docker image:
    - Python 3.9
        - Autologging==1.3.2
        - h5py==3.7.0
        - netCDF4==1.6.1
        - numpy==1.23.4
        - PyYAML==6.0
    - HDF5 libraries
    - HDF5 tools: h5dump
    - netCDF4
- Docker Image information
    - Base Image: centos:7
    - Image Size: 901 Mb
    - Repository Name: 754153872510.dkr.ecr.us-east-1.amazonaws.com/assistt-centos7-python3-surfref
    - Tag Name: v1.0

System requirements for running VIIRS Surface Reflectance include about 1 GB of memory when run on 1 CPU. Total run time might take up to about 2 minutes.

2.3.3 Communication Needs

The target processing system for the SR algorithm package is the NESDIS Common Cloud Framework (NCCF) environment. The communication needs of the system must be sufficient and meet all processing requirements specified throughout section 2.3 of this document.

3. SOFTWARE

This section describes the system-level software elements of the VIIRS Surface Reflectance system as well as the source code and system files. These files are organized into subdirectories. The contents of each subdirectory are identified and their purpose is explained. This includes the various source codes for each section.

3.1 Software Description

There are four primary parts to the VIIRS Surface Reflectance algorithm: (1) The Preprocessor, (2) The main SR (Surface Reflectance) algorithm, (3) The Global Attributes Converter, and (4) The Postprocessor.

The preprocessor performs these primary functions:

- It converts the Enterprise Cloud Mask (in NetCDF-4 format) into an equivalent VIIRS Cloud Mask (in HDF-5 format). This includes remapping bits to simulate the VIIRS Cloud Mask Flags with the closest possible matches from the Enterprise Cloud Mask.
- It copies specific fields from the Aerosol Optical Depth Product (e.g. AOD550, etc.) to an equivalent AOD product in HDF-5 format.
- It extracts total column ozone, total precipitable water, and surface pressure from the GFS product and creates HDF-5 files for each of those three datasets.

The output of the preprocessor includes these five files, which retain the same name regardless of granule:

- NDE_IICMO_populated.h5 (cloud mask)
- NDE_AOD_populated.h5 (aerosol optical depth)
- NDE_ozone_populated.h5 (GFS total column ozone)
- NDE_pres_populated.h5 (GFS surface pressure)
- NDE_tpw_populated.h5 (GFS total precipitable water)

On completion of the preprocessor, the main Surface Reflectance algorithm can be run. This code generates a surface reflectance for each input M-Band (1-5, 7-8, 10-11) and I-Band (1-3). The output of this file is as follows:

- VNP09.Ayyyyjjj.hdf5

The global attribute converter can be run once the main algorithm is complete. It ingests the VNP09 hdf5 file and converts the global attributes from IDPS conventions to NCCF conventions. Output from this algorithm takes the form:

- NDE_equivalent_values_file.cfg

After the equivalent values file has been generated, the postprocessor can execute. It uses the VNP09 file, equivalent values file, and a static attributes file to generate the final product in NetCDF-4 form for distribution to the PDA. The postprocessor also copies latitude and longitude for the 375 meter (GITCO) and 750 meter (GMTCO) resolutions to the netCDF file and adds variable attributes that are specific to the NCCF environment.

3.2 Directory Description

The directory tree provided with the unpacked delivery package will consist of the subdirectories CODE, DATA, and DOCS. The CODE and DATA subdirectories contain the following:

DATA

```
├── input/
│   ├── j01/
│   │   ├── GITCO_j01_d20200817_t2317203_e2318448_b14239_c20200817233209708558_noac_ops.h5
│   │   ├── GMTCO_j01_d20200817_t2317203_e2318448_b14239_c20200817233209765218_noac_ops.h5
│   │   ├── JRR-AOD_v2r3_j01_s202008172317203_e202008172318448_c202008172343240.nc
│   │   ├── JRR-CloudHeight_v2r3_j01_s202008172317203_e202008172318448_c202008172340120.nc
│   │   ├── JRR-CloudMask_v2r3_j01_s202008172317203_e202008172318448_c202008172340050.nc
│   │   ├── NWP_GFS_v2r3_j01_s202008172317203_e202008172318448_c202008172343040.nc
│   │   ├── SVI01_j01_d20200817_t2317203_e2318448_b14239_c20200817233309432386_noac_ops.h5
│   │   ├── SVI02_j01_d20200817_t2317203_e2318448_b14239_c20200817233309461833_noac_ops.h5
│   │   ├── SVI03_j01_d20200817_t2317203_e2318448_b14239_c20200817233309474737_noac_ops.h5
│   │   ├── SVM01_j01_d20200817_t2317203_e2318448_b14239_c20200817233313298861_noac_ops.h5
│   │   ├── SVM02_j01_d20200817_t2317203_e2318448_b14239_c20200817233313383537_noac_ops.h5
│   │   ├── SVM03_j01_d20200817_t2317203_e2318448_b14239_c20200817233313400198_noac_ops.h5
│   │   ├── SVM04_j01_d20200817_t2317203_e2318448_b14239_c20200817233313409012_noac_ops.h5
│   │   ├── SVM05_j01_d20200817_t2317203_e2318448_b14239_c20200817233313420089_noac_ops.h5
│   │   ├── SVM07_j01_d20200817_t2317203_e2318448_b14239_c20200817233313451001_noac_ops.h5
│   │   ├── SVM08_j01_d20200817_t2317203_e2318448_b14239_c20200817233313462444_noac_ops.h5
│   │   ├── SVM10_j01_d20200817_t2317203_e2318448_b14239_c20200817233313310589_noac_ops.h5
│   │   └── SVM11_j01_d20200817_t2317203_e2318448_b14239_c20200817233313322205_noac_ops.h5
│   ├── j02/
│   │   ├── GITCO_j02_d20170721_t1803389_e1805018_b29700_c20191121101244083262_ADu_ops.h5
│   │   ├── GMTCO_j02_d20170721_t1803389_e1805018_b29700_c20191121101245451614_ADu_ops.h5
│   │   ├── JRR-AOD_v2r1_N21_s201707211803389_e201707211805018_c202009231608262.nc
│   │   ├── JRR-CloudHeight_v2r1_N21_s201707211803389_e201707211805018_c202009231245481.nc
│   │   ├── JRR-CloudMask_v2r1_N21_s201707211803389_e201707211805018_c202009231245481.nc
│   │   ├── NOAA21_VIIRS_750M_2017202_1803_38_NWP_GFS-INTP.nc
│   │   ├── SVI01_j02_d20170721_t1803389_e1805018_b29700_c20191122235911541778_ADu_ops.h5
│   │   ├── SVI02_j02_d20170721_t1803389_e1805018_b29700_c20191122235911670693_ADu_ops.h5
│   │   ├── SVI03_j02_d20170721_t1803389_e1805018_b29700_c20191122235911795827_ADu_ops.h5
│   │   ├── SVM01_j02_d20170721_t1803389_e1805018_b29700_c20191122235912064019_ADu_ops.h5
│   │   ├── SVM02_j02_d20170721_t1803389_e1805018_b29700_c20191122235912109529_ADu_ops.h5
│   │   ├── SVM03_j02_d20170721_t1803389_e1805018_b29700_c20191122235912158056_ADu_ops.h5
│   │   ├── SVM04_j02_d20170721_t1803389_e1805018_b29700_c20191122235912208384_ADu_ops.h5
│   │   ├── SVM05_j02_d20170721_t1803389_e1805018_b29700_c20191122235912258318_ADu_ops.h5
│   │   ├── SVM07_j02_d20170721_t1803389_e1805018_b29700_c20191122235912353114_ADu_ops.h5
│   │   ├── SVM08_j02_d20170721_t1803389_e1805018_b29700_c20191122235912403114_ADu_ops.h5
│   │   ├── SVM10_j02_d20170721_t1803389_e1805018_b29700_c20191122235912492494_ADu_ops.h5
│   │   └── SVM11_j02_d20170721_t1803389_e1805018_b29700_c20191122235912537059_ADu_ops.h5
│   └── npp/
│       ├── GITCO_npp_d20200817_t2317203_e2318448_b14239_c20200817233209708558_noac_ops.h5
│       ├── GMTCO_npp_d20200817_t2317203_e2318448_b14239_c20200817233209765218_noac_ops.h5
│       ├── JRR-AOD_v2r3_npp_s202008172317203_e202008172318448_c202008172343240.nc
```

```
│       ├── JRR-CloudHeight_v2r3_npp_s202008172317203_e202008172318448_c202008172340120.nc
│       ├── JRR-CloudMask_v2r3_npp_s202008172317203_e202008172318448_c202008172340050.nc
│       ├── NWP_GFS_v2r3_npp_s202008172317203_e202008172318448_c202008172343040.nc
│       ├── SVI01_npp_d20200817_t2317203_e2318448_b14239_c20200817233309432386_noac_ops.h5
│       ├── SVI02_npp_d20200817_t2317203_e2318448_b14239_c20200817233309461833_noac_ops.h5
│       ├── SVI03_npp_d20200817_t2317203_e2318448_b14239_c20200817233309474737_noac_ops.h5
│       ├── SVM01_npp_d20200817_t2317203_e2318448_b14239_c20200817233313298861_noac_ops.h5
│       ├── SVM02_npp_d20200817_t2317203_e2318448_b14239_c20200817233313383537_noac_ops.h5
│       ├── SVM03_npp_d20200817_t2317203_e2318448_b14239_c20200817233313400198_noac_ops.h5
│       ├── SVM04_npp_d20200817_t2317203_e2318448_b14239_c20200817233313409012_noac_ops.h5
│       ├── SVM05_npp_d20200817_t2317203_e2318448_b14239_c20200817233313420089_noac_ops.h5
│       ├── SVM07_npp_d20200817_t2317203_e2318448_b14239_c20200817233313451001_noac_ops.h5
│       ├── SVM08_npp_d20200817_t2317203_e2318448_b14239_c20200817233313462444_noac_ops.h5
│       ├── SVM10_npp_d20200817_t2317203_e2318448_b14239_c20200817233313310589_noac_ops.h5
│       └── SVM11_npp_d20200817_t2317203_e2318448_b14239_c20200817233313322205_noac_ops.h5
├── logs/
│   ├── VIIRS_Surface_Reflectance_202306121317198.log
│   ├── VIIRS_Surface_Reflectance_202306121325518.log
│   └── VIIRS_Surface_Reflectance_202306121329188.log
└── output/
        ├── j01/
        │   ├── maininput.cnf
        │   ├── NDE_AOD_populated.h5
        │   ├── NDE_equivalent_values_file.cfg
        │   ├── NDE_IICMO_populated.h5
        │   ├── NDE_ozone_populated.h5
        │   ├── NDE_pres_populated.h5
        │   ├── NDE_static_values.cfg
        │   ├── NDE_tpw_populated.h5
        │   ├── NOAA_populated.pcf
        │   ├── postinput.cnf
        │   ├── preinput.cnf
        │   ├── PSF_Surface_Reflectance.psf
        │   ├── SurfRefl_v1r2_j01_s202008172317203_e202008172318448_c202306121320560.nc
        │   └── VNP09.A2020230.hdf5
        ├── j02/
        │   ├── maininput.cnf
        │   ├── NDE_AOD_populated.h5
        │   ├── NDE_equivalent_values_file.cfg
        │   ├── NDE_IICMO_populated.h5
        │   ├── NDE_ozone_populated.h5
        │   ├── NDE_pres_populated.h5
        │   ├── NDE_static_values.cfg
        │   ├── NDE_tpw_populated.h5
        │   ├── NOAA_populated.pcf
        │   ├── postinput.cnf
        │   ├── preinput.cnf
        │   ├── PSF_Surface_Reflectance.psf
        │   ├── SurfRefl_v1r2_n21_s201707211803389_e201707211805018_c202306121327150.nc
        │   └── VNP09.A2017202.hdf5
        └── npp/
                ├── maininput.cnf
                ├── NDE_AOD_populated.h5
```

```
        ├── NDE_equivalent_values_file.cfg
        ├── NDE_IICMO_populated.h5
        ├── NDE_ozone_populated.h5
        ├── NDE_pres_populated.h5
        ├── NDE_static_values.cfg
        ├── NDE_tpw_populated.h5
        ├── NOAA_populated.pcf
        ├── postinput.cnf
        ├── preinput.cnf
        ├── PSF_Surface_Reflectance.psf
        ├── SurfRefl_v1r2_npp_s202008172317203_e202008172318448_c202306121330370.nc
        └── VNP09.A2020230.hdf5
```

CODE

```
├── VIIRS_Surface_Reflectance/
│   ├── ancillary_data/
│   │   ├── config_files/
│   │   │   ├── config_IDPS_to_NDE_attributes.cfg
│   │   │   ├── NDE_equivalent_values_file.cfg
│   │   │   ├── NDE_static_values.cfg
│   │   │   ├── postinput_template.cnf
│   │   │   ├── preinput_template.cnf
│   │   │   ├── Surface_Reflectance_Constants_Template.cfg
│   │   │   └── Surface_Reflectance_Template.cfg
│   │   ├── LUT/
│   │   │   ├── j01/
│   │   │   │   ├── albedo
│   │   │   │   ├── aot
│   │   │   │   ├── atau
│   │   │   │   ├── band
│   │   │   │   ├── reflec
│   │   │   │   ├── scat_ang_incr
│   │   │   │   ├── scat_ang_val
│   │   │   │   ├── szen
│   │   │   │   ├── trans
│   │   │   │   ├── VIIRS-SR-IncScatAngles-LUT_v1.5.06.02_LP
│   │   │   │   ├── VIIRS-SR-IP-AC-INT_v1.5.06.02_LP
│   │   │   │   ├── VIIRS-SR-ScatAngDims-LUT_v1.5.06.02_LP
│   │   │   │   └── vzen
│   │   │   ├── j02/
│   │   │   │   ├── albedo
│   │   │   │   ├── aot
│   │   │   │   ├── atau
│   │   │   │   ├── band
│   │   │   │   ├── reflec
│   │   │   │   ├── scat_ang_incr
│   │   │   │   ├── scat_ang_val
│   │   │   │   ├── szen
│   │   │   │   ├── trans
│   │   │   │   ├── VIIRS-SR-IncScatAngles-LUT_v1.5.06.02_LP
│   │   │   │   ├── VIIRS-SR-IP-AC-INT_v1.5.06.02_LP
│   │   │   │   ├── VIIRS-SR-ScatAngDims-LUT_v1.5.06.02_LP
│   │   │   │   └── vzen
│   │   │   │
```

```
│   │   │   └── npp/
│   │   │       ├── albedo
│   │   │       ├── aot
│   │   │       ├── atau
│   │   │       ├── band
│   │   │       ├── reflec
│   │   │       ├── scat_ang_incr
│   │   │       ├── scat_ang_val
│   │   │       ├── szen
│   │   │       ├── trans
│   │   │       ├── VIIRS-SR-IncScatAngles-LUT_v1.5.06.02_LP
│   │   │       ├── VIIRS-SR-IP-AC-INT_v1.5.06.02_LP
│   │   │       ├── VIIRS-SR-ScatAngDims-LUT_v1.5.06.02_LP
│   │   │       └── vzen
│   │   └── templates/
│   │       ├── NDE_AOD_template.h5
│   │       ├── NDE_IICMO_template.h5
│   │       ├── NDE_ozone_template.h5
│   │       ├── NDE_pres_template.h5
│   │       ├── NDE_tpw_template.h5
│   │       └── NOAA_template.pcf
│   ├── bin/
│   │   ├── exe/
│   │   │   ├── LSIPS_SRF_Driver.exe
│   │   │   ├── Placeholder.txt
│   │   │   ├── postprocessor
│   │   │   └── preprocessor
│   │   └── scripts/
│   │       ├── meta_data_format_converter_IDPS_to_NDE.pl
│   │       └── Walk_Surface_Reflectance_PCF_Pre-Process_Config_and_Output_check.pl
│   ├── build_alg
│   ├── conda_environment.yml
│   ├── downloadCOTS
│   ├── installOTS
│   ├── packages/
│   │   ├── aim/
│   │   │   ├── __init__.py
│   │   │   ├── _data_models.py
│   │   │   └── _interface.py
│   │   └── surface_reflectance/
│   │       ├── __init__.py
│   │       ├── __main__.py
│   │       └── interface.py
│   ├── setEnv.sh
│   ├── setEnvLib/
│   │   ├── addons/
│   │   │   ├── DOTpre-commit-config.yaml
│   │   │   ├── install_mcidas
│   │   │   ├── install_wgrib2
│   │   │   ├── installGCC
│   │   │   └── run_lcov
│   │   ├── build_scripts/
│   │   │   ├── copyOTS_setup.sh
```

```
│   │   │   ├── failure.sh
│   │   │   ├── installOTS_lib.sh
│   │   │   ├── installOTS_setup.sh
│   │   │   ├── library.sh
│   │   │   ├── setEnv_setup.sh
│   │   │   ├── tests_interface.sh
│   │   │   └── tests_setup.sh
│   │   ├── README-example.md
│   │   ├── README.md
│   │   ├── templates/
│   │   │   ├── build_alg
│   │   │   ├── downloadCOTS
│   │   │   ├── installOTS
│   │   │   ├── setEnv.sh
│   │   │   ├── setEnvStaticRuntime.sh
│   │   │   └── test_alg
│   │   ├── test_alg
│   │   └── test_cases/
│   │       ├── full_gcc
│   │       ├── full_intel
│   │       ├── Makefile
│   │       └── setup_test.sh
│   ├── setEnvStaticRuntime.sh
│   └── source/
│       ├── NOAA_Enterprise_PGE511_V1.0.8/
│       │   ├── Calc_Lamb.f
│       │   ├── Calc_Lamb.o
│       │   ├── FORTRAN_src/
│       │   │   ├── AutoGeneratedViirsProductsGbl.f
│       │   │   ├── autogeneratedviirsproductsgbl.mod
│       │   │   ├── AutoGeneratedViirsProductsGbl.o
│       │   │   ├── fract_const_dp.mod
│       │   │   ├── fract_const_sp.mod
│       │   │   ├── InfTk_FInterface.f
│       │   │   ├── inftk_finterface.mod
│       │   │   ├── InfTk_FInterface.o
│       │   │   ├── int_kind.mod
│       │   │   ├── libFortranSRC.a
│       │   │   ├── Makefile
│       │   │   ├── phys_const_dp.mod
│       │   │   ├── phys_const_sp.mod
│       │   │   ├── ProCmnConst.f
│       │   │   ├── ProCmnConst.o
│       │   │   ├── ProCmnDefs.f
│       │   │   ├── procmndefs.mod
│       │   │   ├── ProCmnDefs.o
│       │   │   ├── ProCmnKinds.f
│       │   │   ├── ProCmnKinds.o
│       │   │   ├── ProCmnPhysConst.f
│       │   │   ├── ProCmnPhysConst.o
│       │   │   ├── ProCmnXmlDimensions.f
│       │   │   ├── procmnxmldimensions.mod
│       │   │   ├── ProCmnXmlDimensions.o
```

```
│        │    │    ├── real_kind.mod
│        │    │    ├── si_prefix_dp.mod
│        │    │    ├── si_prefix_sp.mod
│        │    │    ├── whole_const_dp.mod
│        │    │    ├── whole_const_int32.mod
│        │    │    └── whole_const_sp.mod
│        │    ├── Generate_QC_Flags.cpp
│        │    ├── Generate_QC_Flags.o
│        │    ├── Generate_SurfReflect_IP.cpp
│        │    ├── Generate_SurfReflect_IP.o
│        │    ├── library_items.cpp
│        │    ├── library_items.o
│        │    ├── Local_XGranDriver.h
│        │    ├── LocalDriverUtils.h
│        │    ├── LocalFieldSwathNames.cpp
│        │    ├── LocalFieldSwathNames.h
│        │    ├── LocalFieldSwathNames.o
│        │    ├── LocalProCmnLogger.h
│        │    ├── LocalProCmnViirsTrimTable.h
│        │    ├── LSIPS_driver_lib.cpp
│        │    ├── LSIPS_driver_lib.h
│        │    ├── LSIPS_driver_lib.o
│        │    ├── LSIPS_SRF_Driver.cpp
│        │    ├── LSIPS_SRF_Driver.h
│        │    ├── LSIPS_SRF_Driver.o
│        │    ├── LSIPS_utils_lib.cpp
│        │    ├── LSIPS_utils_lib.o
│        │    ├── Makefile
│        │    ├── output.map
│        │    ├── PGE511_Params.h
│        │    ├── ProEdrViirsSurfReflect.h
│        │    ├── ProFail.h
│        │    ├── rules.make
│        │    ├── SurfReflect.h
│        │    ├── SurfReflectDataStruct.h
│        │    ├── SurfReflectStructStruct.h
│        │    ├── SurfReflPtrStruct.f
│        │    ├── surfreflptrstruct.mod
│        │    ├── SurfReflPtrStruct.o
│        │    └── system_header.h
│        ├── postprocessor/
│        │    ├── generate_uuid.py
│        │    ├── Makefile
│        │    ├── postprocessor.cpp
│        │    └── postprocessor.h
│        └── preprocessor/
│        ├── Makefile
│        ├── preprocessor.cpp
│        └── preprocessor.h
└── wrapper/
         ├── config/
         │    ├── application_config_j01.yaml
         │    ├── application_config_j02.yaml
```

```
│   ├── application_config_npp.yaml
│   ├── docker_config_j01.yaml
│   ├── docker_config_j02.yaml
│   └── docker_config_npp.yaml
└── scripts/
├── ccap_utils/
│   ├── __init__.py
│   ├── application_init.py
│   ├── docker_wrapper.py
│   └── parse_config.py
├── launch_viirs_surface_reflectance.py
├── submodules/
│   └── ccap_utils/
│       ├── ccap_utils/
│       │   ├── __init__.py
│       │   ├── application_init.py
│       │   ├── docker_wrapper.py
│       │   └── parse_config.py
│       ├── dist/
│       │   └── ccaputils-0.0.1-py3-none-any.whl
│       ├── Dockerfile
│       ├── docs/
│       │   ├── make.bat
│       │   ├── Makefile
│       │   ├── requirements.txt
│       │   └── source/
│       │       ├── conf.py
│       │       ├── how_tos/
│       │       │   ├── hide_how_to_contribute.rst
│       │       │   ├── how_to_contribute.rst
│       │       │   └── images/
│       │       │       ├── code_comment.png
│       │       │       ├── issue_description.png
│       │       │       ├── issues_toc.png
│       │       │       ├── merge_request_page.PNG
│       │       │       ├── merge_request_page_ci.PNG
│       │       │       ├── mr_discussion.PNG
│       │       │       ├── new_issue.png
│       │       │       ├── new_issue_submit.PNG
│       │       │       └── new_mr_feature_branch.png
│       │       └── index.rst
│       ├── Jenkinsfile
│       ├── pylintrc
│       ├── README.rst
│       ├── requirements.txt
│       ├── setup.cfg
│       ├── setup.py
│       └── tests/
│           ├── __init__.py
│           ├── conftest.py
│           ├── test_application_init.py
│           ├── test_data/
│           │   └── sample_config.yaml
│           │
```

```
│                    ├── test_docker_wrapper.py
│                    └── test_parse_config.py
└── viirs_surface_reflectance.py
```

3.3 Source Code Description

The VIIRS surface reflectance algorithm totals approximately 15000 lines of code. This section will identify the main purposes of each of these codes.

The codes are as follows:

SurfRefl/CODE/wrapper/scripts/

- launch_viirs_surface_reflectance.py - This script creates a Docker container which executes a python command needed to complete viirs_surface_reflectance processing.
- viirs_surface_reflectance.py - When run inside of a Docker container, this script will execute the program interface and generate output for the VIIRS Surface Reflectance algorithm.

preprocessor/

- preprocessor.cpp: This is the primary preprocessor code that’s used to convert the NDE NetCDF-4 files into IDPS HDF-5 equivalents.
- preprocessor.h: This includes a list of prototypes to define the functions in the preprocessor.cpp file.

NOAA_Enterprise_PGE511_V1.0.8/

- LSIPS_SRF_Driver.cpp: This is the main driver for the SR primary algorithm.
- Generate_QC_Flags.cpp: This ingests the quality flags from the cloud mask, aerosol, GFS, and VIIRS SDR files.
- Generate_SurfReflect_IP.cpp: This defines the quality flags for the surface reflectance algorithm.
- In addition, several .h files are used to define structures, prototypes, and constants for the cpp codes.
- Calc_Lamb.f: This performs the bulk of the calculations for surface reflectance.
- SurfReflPtrStruct.f: This defines a number of parameters for use in the Calc_Lamb.f routine.
- The remaining files (library_items.cpp, LocalFieldSwathNames.cpp, LSIPS_driver_lib.cpp, LSIPS_utils_lib.cpp) are remnants of the pre-existing ADL routines that are used to define structures and variables common to multiple JPSS algorithms.

postprocessor

- postprocessor.cpp: This is the primary postprocessor code that converts the surface reflectance code from HDF-5 to NetCDF-4, and adds the necessary global attributes to be NDE-compliant. The postprocessor also copies the latitudes and longitudes from the GITCO and GMTCO files.
- postprocessor.h: This includes a list of prototypes to define the functions in the postprocessor.cpp file.

The code is all documented with headers following the SPSRB standards, so if any of the subroutines or functions need to be explored, their functions and descriptions can be found in the source.

4. NORMAL OPERATIONS

4.1 System Control

4.1.1 System Control Files

The VIIRS Surface Reflectance CCAP consists of a single processing unit. Two configuration YAML files containing all the information about the run are required. One contains information used to launch the Docker container, and the other contains information to run the interface.

To execute the processing unit of VIIRS Surface Reflectance processing, the following commands should be executed:

python $home/CODE/wrapper/scripts/launch_viirs_surface_reflectance.py docker_config_<satellite>.yaml

4.1.2 Processing Controls

Two configuration YAML files are necessary for processing. In each unit, one YAML file contains information used to launch the Docker container, and the other contains information to run the processing. The two types of YAML config files presented here are known as 'Docker' and 'Application' YAML files. The parameters for the Application and Docker YAML files are listed below, and then are followed by examples of the YAML files for Surface Reflectance processing.

Docker YAML File Arguments:

This file contains information for setting up the docker run command. For convenience, all the items which may vary between runs or which are system specific are located at the top of the YAML as anchors in arguments. These changing arguments are listed in the table below:

`&bin_dir`	Path to the directory containing the executables
`&scripts`	Path to the python3 wrapper script
`&input_dir`	Path to the input files
`&app_yaml`	Path to the application YAML file
`&ancillary_dir`	Path to the ancillary files

`&packages`	Path to the surface reflectance algorithm python files
`&output_dir`	Path to the output location
`&log_dir`	Path to the log output
`&caseid`	Case ID for the run

Here is an example: **docker_config_j01.yaml**

```
version: 1.0
arguments:
  - &bin_dir /data/data090/brandon.laufer/CCAP/SurfRefl/CODE/VIIRS_Surface_Reflectance/bin
  - &scripts /data/data090/brandon.laufer/CCAP/SurfRefl/CODE/wrapper/scripts
  - &input_dir /data/data090/brandon.laufer/CCAP/SurfRefl/DATA/input/j01
  - &app_yaml /data/data090/brandon.laufer/CCAP/SurfRefl/CODE/wrapper/config/
  - &ancillary_dir
/data/data090/brandon.laufer/CCAP/SurfRefl/CODE/VIIRS_Surface_Reflectance/ancillary_data
  - &packages
/data/data090/brandon.laufer/CCAP/SurfRefl/CODE/VIIRS_Surface_Reflectance/packages
  - &output_dir /data/data090/brandon.laufer/CCAP/SurfRefl/DATA/output/j01
  - &log_dir /data/data090/brandon.laufer/CCAP/SurfRefl/DATA/logs
  - &caseid 20230324

docker_info:
  image: git.star.nesdis.noaa.gov:4567/assistt/docker_image/nrt_python:3.9.15-logging
  python_info:
    script: /home/WORKING_DIR/scripts/viirs_surface_reflectance.py
    application_yaml: /home/WORKING_DIR/config/application_config_j01.yaml

  mount_pairs:
  - src: *bin_dir
    dst: /home/WORKING_DIR/bin
  - src: *scripts
    dst: /home/WORKING_DIR/scripts
  - src: *input_dir
    dst: /home/WORKING_DIR/input
  - src: *app_yaml
    dst: /home/WORKING_DIR/config
  - src: *ancillary_dir
    dst: /home/WORKING_DIR/ancillary
  - src: *packages
    dst: /home/WORKING_DIR/packages
  - src: *output_dir
    dst: /home/WORKING_DIR/output
  - src: *log_dir
```

```
    dst: /home/WORKING_DIR/logs

  docker_args:
    memory: 1Gb
    ulimit:
      stack: -1
  container_name:
    sat: multi
    proj: VIIRS_Surface_Reflectance
    caseid: *caseid
```

Application YAML File Arguments:

The file contains input parameters required by the inner script. The specific variables which will change are maintained in the top of the YAML as the following anchor keys:

`&logging_level`	Sets the logging level for Python's logging module. Possibilities are: `DEBUG, INFO, WARNING, ERROR, CRITICAL`
`&satellite`	Satellite for which the code is being executed. Options are j01, j02, and npp.
`&production_site`	value for the "production_site" metadata
`&production_environment`	value for the "production_environment" metadata

Here is an example: **application_config_j01.yaml**

```
version: 1.0
arguments:
  - &logging_level 'DEBUG'
  - &production_site 'PLAIT'
  - &production_environment 'INT'
  - &satellite 'j01'

application_info:
  outdir: /home/WORKING_DIR/output/
  logdir: /home/WORKING_DIR/logs/
  algorithm_spec:
    - name: logging_level
      value: *logging_level
    - name: production_site
      value: *production_site
```

```
- name: production_environment
  value: *production_environment
- name: working_dir
  value: /home/WORKING_DIR/working/
- name: bin_dir
  value: /home/WORKING_DIR/bin/
- name: ancillary_dir
  value: /home/WORKING_DIR/ancillary/
- name: input_dir
  value: /home/WORKING_DIR/input/
- name: output_dir
  value: /home/WORKING_DIR/output
- name: log_dir
  value: /home/WORKING_DIR/logs/
- name: satellite
  value: *satellite
```

4.2 Installation

Setup of the VIIRS Surface Reflectance code occurs in four stages: (1) unpacking the DAP, (2) compiling the code, (3) installing the compiled code and system files into the NDE production system, and (4) configuring of the NDE system using the production rules supplied in the SR documentation. This section identifies the items to be installed.

4.2.1 Installation items

The CCAP consists of 3 gzip'd tar files:

```
VIIRS_Surf_Refl_CCAP_CODE_20230616.tar.gz
VIIRS_Surf_Refl_CCAP_DATA_20230616.tar.gz
VIIRS_Surf_Refl_CCAP_DOCS_20230616.tar.gz
```

To unpack the package, type:

```
tar -xvfz <tarfile_name>,
```

where `tarfile_name` is the name of the tarfile you wish to unpack.

There will be 3 subdirectories created in the current working directory:

- `$home/CODE/`
- `$home/DATA/`
- `$home/DOCS/`

Note it will require about 11 Gb to store the unpacked data.

The executables will be found in: VIIRS_Surface_Reflectance/bin/exe/

LSIPS_SRF_Driver.exe
postprocessor
preprocessor

Refer to Section 3.2 for a detailed description of the directory structure and associated files after all three files are unpacked.

4.2.2 Compilation Procedures

(Please refer to the README document included with this delivery package).

Required libraries:

- HDF-5
- netCDF4

While statically compiled executables are provided, the following details how to compile the executables. (Let $home be the directory in which the tarfiles have been extracted.)

- Set the compiler to gcc-8.3.0 by adding it to your PATH
- Change to the directory $home/CODE/VIIRS_Surface_Reflectance
- Set the required environment variables by using
  source setEnv.sh
- Build the algorithm using the included script with
  ./build_alg clean
  ./build_alg
- This will create three executables in
  $home/CODE/VIIRS_Surface_Reflectance/bin/exe
    - LSIPS_SRF_Driver.exe
    - postprocessor
    - preprocessor

4.2.3 Installation Procedures

Please refer to below link (Document Object: 77):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7IsXg/edit?pli=1#heading=h.uqntmud8purc

4.3 Configuration Procedures

4.3.1 Production Rules

This document contains the production rules required to properly execute the driver script for VIIRS Surface Reflectance. The VIIRS Surface Reflectance generates a single SurfaceReflectance NetCDF file. The rules cover the required input, how to run the driver scripts, and the expected output.

The Surface Reflectance NetCDF4 product files will contain surface reflectance data for each M-band and I-band used as input to the algorithm package. For the M-bands, the resolution will be 750 meters. For the I-bands, the resolution will be 375 meters .The amount of data within the I-Band product files will be roughly four times larger than the amount of data within the M-Band product files as the I-Bands have a higher swath resolution.

Processing Rules

All required inputs must be provided; the list of requirements is in the Primary Input Data section. Most of these inputs are sourced from the same VIIRS granule-set, so the production rule timeout is generally not a concern. The timeout is set to one hour and begins when the first JPSSRR_AOD file is received. This event serves as the trigger. Since this file has similar dependencies to the other files, it is usually one of the last files to arrive. This allows the job to run immediately.

Timing may differ between the Integration and Testing (IT) and Operations OP) departments, as the JRR products are created by a single production rule (JRR v3.2) in IT, while OP still uses JRR v2.3. The production of JPSSRR_NPP_VIIRS_NWP_GFS follows its own set of production rules, which has dependencies on CloudMask and GFS ancillary files, as well as some of the same IDPS-created SDRs required by all of these products.

Supported Satellites:

- JPSS-1
- NOAA-21
- S-NPP

Primary Input Data

These files are the satellite data files required to produce the VIIRS Surface Reflectance product. The sources of satellite data are listed below:

- Operational products of VIIRS RDR/SDR passthroughs (HDF5)
- Active NOAA Enterprise Products (NetCDF-4)

The expected format of the satellite filenames are as follows:

PDA ShortName and Item	**Description**	**Product FileName**
VIIRS-I[1-3]-SDR VIIRS Imagery Band SDR	Sensor Data Records	SVI0[1-3]_<SAT>_d<DATE>_t<TIME>_e<TIME>_b<ORBIT>_c<DATE/TIME>_naoc_ops.h5
Example: SVI01_j01_d20200817_t2317203_e2318448_b14239_c20200817233309432386_noac_ops.h5		
VIIRS-M[1-5/7-8/10-11]-SDR VIIRS Moderate Resolution Band SDR	Sensor Data Records	SVM[01-05/07-08/10-11]_<SAT>_d<DATE>_t<TIME>_e<TIME>_b<ORBIT>_c<DATE/TIME>_naoc_ops.h5
Example: SVM01_j01_d20200817_t2317203_e2318448_b14239_c20200817233313298861_noac_ops.h5		
VIIRS-MOD-GEO-TC Moderate Resolution Band SDR Terrain Correcte	Sensor Data Records	GMTCO_<SAT>_d<DATE>_t<TIME>_e<TIME>_b<ORBIT>_c<DATE/TIME>_naoc_ops.h5

d Geolocation		
Example: GMTCO_j01_d20200817_t2317203_e2318448_b14239_c20200817233209765218_noac_ops.h5		
VIIRS-IMG-GEO-TC Imagery Band SDR Terrain Corrected Geolocation	Sensor Data Records	GITCO_<SAT>_d<DATE>_t<TIME>_e<TIME>_b<ORBIT>_c<DATE/TIME>_naoc_ops.h5
Example: GITCO_j01_d20200817_t2317203_e2318448_b14239_c20200817233209708558_noac_ops.h5		
JRR-AOD Aerosol Optical Depth	NUPS	JRR-AOD_<VERSION>_<SAT>_s<DATE/TIME>_e<DATE/TIME>_c<DATE/TIME>.nc
Example: JRR-AOD_v2r3_j01_s202008172317203_e202008172318448_c202008172343240.nc		
JRR-CloudHeight Cloud Height Product	NUPs	JRR-CloudHeight_<VERSION>_<SAT>_s<DATE/TIME>_e<DATE/TIME>_c<DATE/TIME>.nc
Example: JRR-CloudHeight_v2r3_j01_s202008172317203_e202008172318448_c202008172340120.nc		

JRR-CloudMask Enterprise Cloud Mask	NUPs	JRR-CloudMask_<VERSION>_<SAT>_s<DATE/TIME>_e<DATE/TIME>_c<DATE/TIME>.nc
Example: JRR-CloudMask_v2r3_j01_s202008172317203_e202008172318448_c202008172340050.nc		
NWP_GFS GFS Data	NUPs	NWP_GFS_<VERSION>_<SAT>_s<DATE/TIME>_e<DATE/TIME>_c<DATE/TIME>.nc
Example: NWP_GFS_v2r3_j01_s202008172317203_e202008172318448_c202008172343040.nc		

Where:

<VERSION> - version of file as <v?r?>
<SAT> - the type of satellite used (e.g., j01, j02, npp)
<DATE> - format of <YYYYmmdd>
<TIME> - format of <HHMMsss>
<DATE/TIME> - format of <YYYYmmddHHMMsss>
<YYYY> - Year
<ORBIT> - Orbit number, a five-digit number continuously increasing over time

Dynamic Ancillary Data

There are no dynamic ancillary data needed to generate the VIIRS Surface Reflectance products.

Static Ancillary Data

The static ancillary data needed to generate the VIIRS Surface Reflectance products is located in the following directory:

$home/CODE/VIIRS_Surface_Reflectance/ancillary_data
↳ config_files/

- Contains templates and .cfg files

↳ LUT/<SAT>/

- Look-up Tables for each satellite

↳ templates/

- h5 templates and .pcf template

The templates will be populated after the AIM copies them to the working directory.

Example:

- NDE_twp_template.h5 -> NDE_twp_populated.h5
- NDE_pres_template.h5 -> NDE_pres_populated.h5
- NDE_ozone_template.h5 -> NDE_ozone_populated.h5
- preinput_template.cnf -> preinput.cnf
- postinput_template.cnf -> postinput.cnf
- NOAA_template.pcf -> NOAA_populated.pcf

Other Required Inputs

Two configuration YAML files are needed to guide the python utilities that perform the bulk of the tasks.

docker_config.yaml - This contains the information needed to run the docker container. All needed changes are within arguments at the top of the file. They are:

&bin_dir	Path to the directory containing the executables
&wrapper	Path to the python3 wrapper script
&input_dir	Path to the input files
&app_yaml	Path to the application YAML file
&ancillary_dir	Path to the ancillary files
&packages	Path to the surface reflectance algorithm python files
&output_dir	Path to the output location
&log_dir	Path to the log output
&caseid	Case ID for the run

application_config.yaml - Contains the information needed to guide the script that will run inside the docker container. The needed changes are again within arguments at the top of file and are:

&logging_level	Sets the logging level for Python's logging module. Possibilities are: DEBUG, INFO, WARNING, ERROR, CRITICAL
&satellite	Satellite for which the code is being executed, possibilities are j01, j02, or npp
&production_site	value for the "production_site" metadata
&production_environment	value for the "production_environment" metadata

Execution

To execute the processing unit of VIIRS Surface Reflectance processing, the following commands should be executed:

```
python $home/CODE/VIIRS_Surface_Reflectance/wrapper/launch_
        viirs_surface_reflectance.py $home/CODE/VIIRS_Surface_
        Reflectance/config/docker_config.yaml
```

Log Files

Each run produces multiple logs that can be used to determine if the run was successful or if there were errors. The location of the log files are as follows:

- $home/DATA/logs/

The names of the log file produced by the VIIRS Surface Reflectance processing are as follows:

- VIIRS_Surface_Reflectance_<sat>_<YYYYmmddHHMMSSf>.log
  - This log will contain terminal outputs/errors for the python wrapper/docker container
- PSF_Surface_Reflectance.psf
  - This contains the name of the surface reflectance file on success and an error message on failure

Output Files

The official expected output of the VIIRS Surface Reflectance algorithm processing consists of 1 SurfRefl NetCDF file for each unit.

The NetCDF output file can be found as specified in the docker configuration YAML file.

Each unit produces 1 NetCDF file. The output produced will follow this file name format:

SurfRefl_v1r3_<SAT>_s<DATE/TIME>_e<DATE/TIME>_c<DATE/TIME>.nc

Where:

<SAT> - the type of satellite used (eg. j01, n21, npp)

<DATE/TIME> - format of <YYYYmmddHHMMsss>

Example:

SurfRefl_v1r3_j01_s202008172317203_e202008172318448_c202303131546130.nc

Output File	Average File Size
SurfRefl NetCDF	218.8 MB

4.4 Operations Procedures

4.4.1 Normal Operations

Please refer to below link (Document Object: 10):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7IsXg/edit?pli=1#heading=h.uqntmud8purc

4.4.2 Data Preparation

Please refer to below link (Document Object: 73 and 81):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7IsXg/edit?pli=1#heading=h.uqntmud8purc

and NCCF Data Ingest Pipeline Process document:

https://docs.google.com/document/d/1jOa3IfuEGhRxfZ_UDf9AkGXFiQXJbO1k3WiNS1PxReQ/edit

4.5 Distribution

4.5.1 Data Transfer / Communications

Please refer to below link (Document Object: 73 and 81):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7IsXg/edit?pli=1#heading=h.uqntmud8purc

4.5.2 Distribution Restrictions

Please refer to below link (Document Object: 73 and 81):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7IsXg/edit?pli=1#heading=h.uqntmud8purc

4.5.3 Product Retention Requirements

Please refer to below link (Document Object: 73 and 81):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7IsXg/edit?pli=1#heading=h.uqntmud8purc

4.5.4 External Product Tools

No external product tools are supplied. The Surface Reflectance output file is in netCDF4 format. External users can choose their own tools to display and analyze these output files with any publicly available netCDF tools.

5. MONITORING AND MAINTENANCE

5.1 Job Monitoring

NCCF job Alarms can be found at CloudWatch:

https://us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#home:

and Elastic dashboard:

https://nccf.espc.nesdis.noaa.gov/s/analyst/app/dashboards#/view/bfdb75f9-6402-56bb-8752-1037cc16f681?_g=(filters:!(),refreshInterval:(pause:!f,value:10000),time:(from:now-24h,to:now))

Also, please refer to below link (Document Object: 43, 44, and 45):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7IsXg/edit?pli=1#heading=h.uqntmud8purc

5.2 Data Signal Monitoring

Please refer to below link (Document Object: 73 and 81):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7IsXg/edit?pli=1#heading=h.uqntmud8purc

5.3 Product Monitoring

5.3.1 Internal Product Tools

No internal product tools are supplied.

5.3.2 Performance Statistics

Please refer to below link (Document Object: 43, 44, and 45):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7lsXg/edit?pli=1#heading=h.uqntmud8purc

5.3.3 Product Monitoring

Product quality is monitored using the NCCF Product Monitoring Tool :

https://nccf.espc.nesdis.noaa.gov/mtool/index.html

Users can use this page to monitor hourly summaries of RAVE product quality based on parameter thresholds.

The NCCF Product Visualization Page:
RAVE in the NCCF PROD Public webpage URL is (will be updated):

https://www.ospo.noaa.gov/Products/land/rave/

and the output netCDF data can be found here:

https://www.ospo.noaa.gov/Visualization01/cData/Blended/RAVE/

5.4 Maintenance

5.4.1 Monitoring

Please refer to below link (Document Object: 84):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7lsXg/edit?pli=1#heading=h.uqntmud8purc

5.4.2 Science Maintenance

The science team lead will provide quality monitoring tools (software) and guidance to the OSPO PAL so that regular monitoring of operational products can be performed. The science team and

OSPO personnel will communicate regularly to discuss potential data quality issues and to formulate and schedule updates to Surface Reflectance science code.

The current project plan allows for four deliveries each space about one year apart. The first two deliveries will contain the Day 1 and Day 2 product generation capabilities. Each subsequent delivery will provide an opportunity to fix bugs, update algorithms, and enhance product quality for those capabilities previously delivered. The periods in between deliveries will allow the science team additional time to perform more comprehensive and long validation.

5.4.3 Library Maintenance

Please refer to below link (Document Object: 71):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7IsXg/edit?pli=1#heading=h.uqntmud8purc

5.4.4 Special Maintenance Procedures

Please refer to below link (Document Object: 72):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7IsXg/edit?pli=1#heading=h.uqntmud8purc

5.4.5 Maintenance Utilities

Please refer to below link (Document Object: 84):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7IsXg/edit?pli=1#heading=h.uqntmud8purc

5.5 Program Backup Procedures

Please refer to below link (Document Object: 89 and 90):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7IsXg/edit?pli=1#heading=h.uqntmud8purc

6. TROUBLESHOOTING

6.1 Problem Diagnosis and Recovery

Please refer to below link (Document Object: 82):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7lsXg/edit?pli=1#heading=h.uqntmud8purc

6.1.1 Quality Control Output

There are seven sets of quality bitmask flags in the Surface Reflectance system. These bits do not mean that the surface reflectance algorithm failed. Instead, they just indicate that the quality of the product may be impacted either by an input file or a calculation inside the algorithm. These flags are defined in detail in Table 6-1 through Table 6-7.

Table 6-1 - QF1 Surface Reflectance

Bit #	Meaning
7	(Empty)
6	Sun Glint: 0: no sun glint detected 1: sun glint detected
5	Low Sun Mask 0: high 1: low
4	Day/Night Flag 0: day 1: night
2-3	Cloud Detection and Confidence 00: confident clear 01: probably clear 10: probably cloudy 11: confidence cloudy
0-1	Cloud Mask Quality (currently undefined) 00: poor 01: low 10: medium 11: high

Table 6-2 - QF2 Surface Reflectance

Bit #	Meaning
7	Thin Cirrus Detected – Emissive Test 0: no cloud 1: cloud
6	Thin Cirrus Detected – Reflective Test 0: no cloud 1: cloud
5	Snow/Ice Flag 0: no snow/ice 1: snow or ice
4	Heavy Aerosol Mask 0: no heavy aerosol 1: heavy aerosol
3	Cloud Shadow Mask 0: no cloud shadow 1: shadow
0-2	Land/Water Background 001: deep ocean 010: shallow water 011: land 100: snow 101: arctic 110: Antarctic and Greenland 111: desert

Table 6-3 - QF3 Surface Reflectance

Bit #	Meaning
7	Bad M10 SDR data 0: no 1: yes
6	Bad M8 SDR data 0: no 1: yes
5	Bad M7 SDR data 0: no 1: yes
4	Bad M5 SDR data 0: no 1: yes
3	Bad M4 SDR data

	0: no 1: yes
2	Bad M3 SDR data 0: no 1: yes
1	Bad M2 SDR data 0: no 1: yes
0	Bad M1 SDR data 0: no 1: yes

Table 6-4 - QF4 Surface Reflectance

Bit #	Meaning
7	Missing Precipitable Water data 0: no 1: yes
6	Invalid Land AM input data 0: valid 1: invalid or over ocean
5	Missing AOT input data 0: no 1: yes
4	Overall Quality of AOT 0: good 1: bad
3	Bad I3 SDR data 0: no 1: yes
2	Bad I2 SDR data 0: no 1: yes
1	Bad I1 SDR data 0: no 1: yes
0	Bad M11 SDR data 0: no 1: yes

Table 6-5 - QF5 Surface Reflectance

Bit #	Meaning
7	Overall Quality of M7 Surface Reflectance Data 0: good 1: bad
6	Overall Quality of M5 Surface Reflectance Data 0: good 1: bad
5	Overall Quality of M4 Surface Reflectance Data 0: good 1: bad
4	Overall Quality of M3 Surface Reflectance Data 0: good 1: bad
3	Overall Quality of M2 Surface Reflectance Data 0: good 1: bad
2	Overall Quality of M1 Surface Reflectance Data 0: good 1: bad
1	Missing Surface Pressure input data 0: no 1: yes
0	Missing total column ozone input data 0: no 1: yes

Table 6-6 - QF6 Surface Reflectance

Bit #	Meaning
7	Unused
6	Unused
5	Overall Quality of I3 Surface Reflectance Data 0: good 1: bad
4	Overall Quality of I2 Surface Reflectance Data 0: good 1: bad
3	Overall Quality of I1 Surface Reflectance Data 0: good 1: bad
2	Overall Quality of M11 Surface Reflectance Data

	0: good 1: bad
1	Overall Quality of M10 Surface Reflectance Data 0: good 1: bad
0	Overall Quality of M8 Surface Reflectance Data 0: good 1: bad

Table 6-7 - QF7 Surface Reflectance

Bit #	Meaning
7	Unused
6	Unused
5	Unused
4	Thin Cirrus Flag 0: no 1: yes
2-3	Aerosol Quantity 00: climatology 01: low 10: average 11: high
1	Adjacent to Cloud (currently undefined) 0: no 1: yes
0	Snow Present 0: no 1: yes
0	Overall Quality of M8 Surface Reflectance Data 0: good 1: bad

Details regarding statistical quality information NetCDF variable (quality_information) are as follows:

PercentCloud
PercentLand
PercentWater
long_name (includes the metadata: total number of retrievals, percentage of optimal retrievals, percentage of bad retrievals)
percent_cloud_shadow
percent_low_sun

percent_missing_for_band_I1
percent_missing_for_band_I2
percent_missing_for_band_I3
percent_missing_for_band_M1
percent_missing_for_band_M10
percent_missing_for_band_M11
percent_missing_for_band_M2
percent_missing_for_band_M3
percent_missing_for_band_M4
percent_missing_for_band_M5
percent_missing_for_band_M7
percent_missing_for_band_M8
percent_poor_retrieval_for_I1
percent_poor_retrieval_for_I2
percent_poor_retrieval_for_I3
percent_poor_retrieval_for_M1
percent_poor_retrieval_for_M10
percent_poor_retrieval_for_M11
percent_poor_retrieval_for_M2
percent_poor_retrieval_for_M3
percent_poor_retrieval_for_M4
percent_poor_retrieval_for_M5
percent_poor_retrieval_for_M7
percent_poor_retrieval_for_M8
percentage_bad_retrievals
percentage_optimal_retrievals
total_number_retrievals

6.1.2 Error Correction

Please refer to below link (Document Object: 43, 44, and 45):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7lsXg/edit?pli=1#heading=h.uqntmud8purc

6.1.3 Problem Diagnosis and Recovery Procedures

Please refer to below link (Document Object: 82):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7lsXg/edit?pli=1#heading=h.uqntmud8purc

6.1.4 Data Recovery Procedures

Please refer to below link (Document Object: 89):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7lsXg/edit?pli=1#heading=h.uqntmud8purc

6.1.5 Program Recovery Procedures

Please refer to below link (Document Object: 90):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7lsXg/edit?pli=1#heading=h.uqntmud8purc

6.2 Application Shutdown and Restart

6.2.1 Application Shutdown Procedures

Please refer to below link (Document Object: 94):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7lsXg/edit?pli=1#heading=h.uqntmud8purc

6.2.2 Application Restart Procedures

Please refer to below link (Document Object: 92):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7lsXg/edit?pli=1#heading=h.uqntmud8purc

6.3 System Shutdown and Restart

6.3.1 System Shutdown Procedures

Please refer to below link (Document Object: 83, 93, and 95):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7lsXg/edit?pli=1#heading=h.uqntmud8purc

6.3.2 System Restart Procedures

Please refer to below link (Document Object: 83, 93, and 95):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7lsXg/edit?pli=1#heading=h.uqntmud8purc

6.3.3 System Reboot Procedures

Please refer to below link (Document Object: 83, 93, and 95):

https://docs.google.com/document/d/1W2m4Mc0WXDc4AeqIbHDR-BiK1kPCE3ZCoSoBTJ7lsXg/edit?pli=1#heading=h.uqntmud8purc

7. APPENDIX

7.1 Data Flow

See Section 2.3 for the type and amount of dynamical input data into the product system and output files produced on a daily basis. External data is processed at Operations prior to the Surface Reflectance product system.

7.2 Input Data Files

The VIIRS instrument is the sole instrument providing data for processing. The VIIRS Surface Reflectance algorithm requires the VIIRS SDR for each band where surface reflectance is calculated (M1-M5, M7, M8, M10, M11, I1-3), the geolocation for the moderate and imagery bands (GMTCO/GITCO). Table 7-1 lists the naming convention and examples of the VIIRS satellite data input files.

Table 7-1 - Satellite Data Input Files

PDA ShortName and Item	Description	Product FileName
VIIRS-I[1-3]-SDR VIIRS Imagery Band SDR	Sensor Data Records	SVI0[1-3]_<SAT>_d<DATE>_t<TIME>_e<TIME>_b<ORBIT>_c<DATE/TIME>_naoc_ops.h5
Example: SVI01_j01_d20200817_t2317203_e2318448_b14239_c20200817233309432386_noac_ops.h5		
VIIRS-M[1-5/7-8/10-11]-SDR VIIRS Moderate Resolution Band SDR	Sensor Data Records	SVM[01-05/07-08/10-11]_<SAT>_d<DATE>_t<TIME>_e<TIME>_b<ORBIT>_c<DATE/TIME>_naoc_ops.h5
Example: SVM01_j01_d20200817_t2317203_e2318448_b14239_c20200817233313298861_noac_ops.h5		

VIIRS-MOD-GEO-TC Moderate Resolution Band SDR Terrain Corrected Geolocation	Sensor Data Records	GMTCO_<SAT>_d<DATE>_t<TIME>_e<TIME>_b<ORBIT>_c<DATE/TIME>_naoc_ops.h5
Example: GMTCO_j01_d20200817_t2317203_e2318448_b14239_c20200817233209765218_noac_ops.h5		
VIIRS-IMG-GEO-TC Imagery Band SDR Terrain Corrected Geolocation	Sensor Data Records	GITCO_<SAT>_d<DATE>_t<TIME>_e<TIME>_b<ORBIT>_c<DATE/TIME>_naoc_ops.h5
Example: GITCO_j01_d20200817_t2317203_e2318448_b14239_c20200817233209708558_noac_ops.h5		

Where:

<VERSION> - version of file as <v?r?>
<SAT> - the type of satellite used (e.g., j01, j02, npp)
<DATE> - format of <YYYYmmdd>
<TIME> - format of <HHMMsss>
<DATE/TIME> - format of <YYYYmmddHHMMsss>
<ORBIT> - Orbit number, a five-digit number continuously increasing over time

For the above format definitions, YYYY is the 4-digit year, mm is the 2-digit month, dd is the 2-digit day, HH is the 2-digit hour, MM is the 2-digit minute, and sss is the 3 digit second (the third ‘s’ is 10^{ths} of seconds).

7.3 Ancillary Data Files

The main algorithm ingests several dynamic files and static lookup tables. The static files are provided with the delivery. The enterprise cloud mask, cloud height file, aerosol optical depth file, and GFS files at the pixel level are dynamic ancillary data files. Table 7-2 contains the file names and a brief description of the dynamic files, and the static files are presented in Table 7-3.

Table 7-2 - Dynamic Ancillary Data Files

PDA ShortName and Item	Description	Product FileName
JRR-AOD Aerosol Optical Depth	NUPS	JRR-AOD_<VERSION>_<SAT>_s<DATE/TIME>_e<DATE/TIME>_c<DATE/TIME>.nc
Example: JRR-AOD_v2r3_j01_s202008172317203_e202008172318448_c202008172343240.nc		
JRR-CloudHeight Cloud Height Product	NUPs	JRR-CloudHeight_<VERSION>_<SAT>_s<DATE/TIME>_e<DATE/TIME>_c<DATE/TIME>.nc
Example: JRR-CloudHeight_v2r3_j01_s202008172317203_e202008172318448_c202008172340120.nc		
JRR-CloudMask Enterprise Cloud Mask	NUPs	JRR-CloudMask_<VERSION>_<SAT>_s<DATE/TIME>_e<DATE/TIME>_c<DATE/TIME>.nc
Example: JRR-CloudMask_v2r3_j01_s202008172317203_e202008172318448_c202008172340050.nc		
NWP_GFS GFS Data	NUPs	NWP_GFS_<VERSION>_<SAT>_s<DATE/TIME>_e<DATE/TIME>_c<DATE/TIME>.nc
Example: NWP_GFS_v2r3_j01_s202008172317203_e202008172318448_c202008172343040.nc		

Where:

<VERSION> - version of file as <v?r?>
<SAT> - the type of satellite used (e.g., j01, j02, npp)
<DATE> - format of <YYYYmmdd>
<TIME> - format of <HHMMsss>
<DATE/TIME> - format of <YYYYmmddHHMMsss>
<YYYY> - Year

For the above format definitions, YYYY is the 4-digit year, mm is the 2-digit month, dd is the 2-digit day, HH is the 2-digit hour, MM is the 2-digit minute, and sss is the 3 digit second (the third 's' is 10ths of a second).

Table 7-3 – Static Ancillary Data Files

File Name	File Description
aot	Table of 20 aerosol optical thicknesses
vzen	Table of 40 viewing angles
szen	Table of 38 solar zenith angles
reflec	Table of reflectivities, one value each for combinations of: - 4 aerosol models - 20 AOT values - 9 M-bands - 5527 scattering angle combinations
trans	Table of transmittances, one value each for combinations of: - 4 aerosol models - 20 AOT values - 9 M-bands - 15 solar zenith angle bins
albedo	Table of albedos, one value each for combinations of - 4 aerosol models - 20 AOT values - 9 M-bands
VIIRS-SR-IncScatAngles-LUT_v1.5.06.02_LP	Scattering Angle Increment
VIIRS-SR-ScatAngDims-LUT_v1.5.06.02_LP	Location of the first (maximum) scattering angle corresponding to a pair of solar/sensor zenith angles
VIIRS-SR-IP-AC-INT_v1.5.06.02_LP	- Min/Max for Retrieved Surface Reflectance - Min/Max for Aerosol Optical Depth - Minimum value for GFS water vapor, ozone, surface pressure - Aerosol Min/Max Model Values - Rayleigh Optical Depth - Ozone (1 value), water vapor (3 values), and other gas (6 values) transmittance coefficients.

7.4 Look Up Tables

The static ancillary files listed in section 7.3 effectively are lookup tables.

7.5 Intermediate Data Set Description

The working directory will also include some intermediate non-deliverable files, which can be ignored. These include:

- VNP09.A<YYYYmmdd>.hdf5
- NDE_*_populated.h5
- maininput.cnf
- preinput.cnf
- postinput.cnf
- NOAA_populated.pcf

7.6 Output Data Set Description

Table 7-4 lists the single output file from the VIIRS Surface Reflectance algorithm, along with a description and approximate size.

Table 7-4 - VIIRS Surface Reflectance Output File

File	Description	Format	Size/file
SurfRefl_v#r#_<sat>_sYYYYMMDDhhmmsss_eYYYYMMDDhhmmsss_cYYYYMMDDhhmmsss.nc	The SR (Surface Reflectance) file contains surface reflectances for I-bands 1,2,3 and for M-bands 1,2,3,4,5,7,8,10,11. This file also includes five bitmapped quality flags.	NetCDF-4	~209 MB/file 288 files/day

Where:

SurfRefl – Surface Reflectance Products
v#r# – version number, revision number (eg. v1r2)
<SAT> - the type of satellite used (eg. j01, n21, npp)
YYYY – 4-digit Year
MM – 2-digit Month
DD – 2-difit Day of Month
hh – 2-digit Hour
mm – 2-digit Minute
sss – 3-digit Second (3rd digit 10ths of a second)

7.7 Archive Data Description

An archive agreement is currently in progress for the Surface Reflectance product. The SR files will be archived in netCDF-4 format at NCEI and all metadata content will follow the Climate and Forecast metadata convention that NDE adopts for all NPP products.

8. LIST OF REFERENCES

NESDIS/OSPO (2023), VIIRS Surface Reflectance External Users Manual (EUM)

NESDIS/STAR (2019), Surface Reflectance Requirements Allocation Document (RAD)

NESDIS/STAR (2023), VIIRS Surface Reflectance Delivery (CCAP) Documents (Readme file, PCF-PSF document, Production Rules, and the Delivery Memo)

NOAA/CIRA (2018): Surface Reflectance Algorithm Theoretical Basis Document (ATBD)

END OF DOCUMENT