# Office of Satellite and Product Operations
# Environmental Satellite Processing Center


# NVPS Green Vegetation Fraction System Maintenance Manual

## Version 4.3

## June 5, 2024

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

# Authors

John Lindeman (OCS)

Updated by Jeffrey Augenbaum (OMS, ERT)

# Approval Page

# Environmental Satellite Processing Center
# NVPS Green Vegetation Fraction
# System Maintenance Manual

| | |
|---|---|
| GROUP: OMS Date: 06/18/2024<br><br>Jeffrey Augenbaum, Lead Programmer | GROUP: OSPO Date: 06/25/2024<br><br>Hanjun Ding, Product Area Lead |
| GROUP: OMS Date: 07/08/2024<br><br>Clay Davenport, Products Manager | |

# Changes/Revisions Record

This system maintenance manual 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.

| Version Number | Date | Description of Change/Revision | Section/Pages Affected | Changes Made by Name/Title/ Organization |
|---|---|---|---|---|
| 4.1 | 01/2023 | Original version | All | Lindeman |
| 4.2 | 05/2024 | Added links to NCCF SMM; Added links to Product monitoring, visualization and Alarms; Added data flow diagrams; Added links to NCCF SMM | 2.1, 2.2; 5.3; 7.1; 1.3, 2.1, 2.2, 4.2.2, 4.2.3, 4.4.1, 5.2-5.3, 6.1-6.3 | Augenbaum |
| 4.3 | 06/05/2024 | Transfer contents to latest template, perform minor editing and formatting | All | Hannah Willett, Technical Writer, ERT Inc. |
| 4.3 | 06/08/2024 | Quality Assurance | All | Clint Sherwood, Quality Assurance Manager, ERT Inc. |

# Preface

This document comprises the National Oceanic and Atmospheric Administration (NOAA) National Environmental Satellite, Data, and Information Service (NESDIS), Office of Satellite and Product Operations (OSPO), publication of this NVPS Green Vegetation Fraction (GVF) System Maintenance Manual. This document reflects current operations for the DOC/NOAA/NESDIS Environmental Satellite Processing Center (ESPC) (NOAA5045) information technology systems. This document describes the established ESPC procedures for GVF system maintenance in accordance with Federal, DOC, NOAA, NESDIS and OSPO requirements.

Future updates and revisions to this document will be produced and controlled by DOC/NOAA/NESDIS for ESPC information technology systems.

The published version of this document can be found at the OSPO SharePoint Products Library.

# Table of Contents

# List of Tables



# List of Figures

# Executive Summary

This is a System Maintenance Manual (SMM) document describing the NDE Vegetation Product System (NVPS) Green Vegetation Fraction (GVF) software package, which generates a consistent set of global and regional gridded vegetation products from Visible Infrared Imaging Radiometer Suite (VIIRS) observations for initializing environmental models and monitoring land use and land cover change.

The NVPS GVF package was designed to run within the NESDIS Cloud Common Framework (NCCF) production environment.

The product development team consists of members from Office of Common Services (OCS), National Environmental Satellite, Data, and Information Service (NESDIS), Office of Satellite and Product Operations (OSPO), and National Weather Service (NWS). The team member's name, organization, role, and contact information can be seen in Table 0-1.

**Table 0-1 - Product Team Members**

| Team Member | Organization | Role | Contact Information |
|---|---|---|---|
| Walter Wolf | OCS | Product Portfolio Management Lead | walter.wolf@noaa.gov |
| Kelly Neely | OCS | ASSISTT Project Manager | kelly.neely@noaa.gov |
| Shanna Sampson | OCS | ASSISTT Solutions Architect | shanna.sampson@noaa.gov |
| Priyanka Roy | OCS | ASSISTT | Priyanka.roy@noaa.gov |
| Gian Villamil-Otero | OCS | ASSISTT | gian.villamil-otero@noaa.gov |
| Yunhui Zhao | OCS | ASSISTT CM Lead & Special Projects | yunhui.zhao@noaa.gov |
| Yunyue (Bob) Yu | OCS | ASSISTT | |
| Zhangyan Jiang | OCS | ASSISTT | |
| Shukming (Eva) Wu | OCS | ASSISTT | |
| Ming Fang | OCS | ASSISTT | |
| Tracey Dorian | OCS | ASSISTT | |
| Letitia Soulliard | OCS | ASSISTT | |
| Hua Xie | OCS | ASSISTT Science QA | hua.xie@noaa.gov |
| Mingming Yao | OCS | ASSISTT | mingming.yao@noaa.gov |

| Team Member | Organization | Role | Contact Information |
|---|---|---|---|
| Edrees Wardak | OCS | ASSISTT | |
| Edward Borders | OCS | ASSISTT | edward.borders@noaa.gov |
| Jonathan Hansford | OCS | ASSISTT | |
| Ramaswamy Tiruchirapalli | OCS | ASSISTT | ramaswamy.tiruchirapalli@noaa.gov |
| John Lindeman | OCS | ASSISTT Documentation | john.lindeman@noaa.gov |

PAL: Hanjun Ding

Email: Hanjun.Ding@noaa.gov

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

Kelly Neely

Email: Kelly.Neely@noaa.gov

Jonathan Hansford

Email: Jonathan.hansford@noaa.gov

Tracey Dorian

Email: Tracey.Dorian@noaa.gov

Priyanka Roy

Email: Priyanka.Roy@noaa.gov

Ramaswamy Tiruchirapalli

Email: Ramaswamy.Tiruchirapalli@noaa.gov

Edward Borders

Email: Edward.Borders@noaa.gov

The low-level code within the NVPS GVF processing system is written in Fortran 90 and C++. This low-level code performs all data processing, scientific computation, reading/writing, reformatting, and opening/closing of files. All high-level code within the NVPS GVF processing system is written in Python. The high-level code performs tasks such as file management, system management, making system calls, and error trapping from the lower-level processing. The driver script will manage the NVPS GVF software and call any necessary unit scripts. The system is comprised of only one unit that handles all of the NVPS GVF processing. This unit will, therefore, produce all expected output product files.

The NESDIS' Policy on Access and Distribution of Environmental Data and Products is provided at:

http://www.ospo.noaa.gov/Organization/About/access.html.

Users need to fill out the Data Access Request Form located on this site and submit to the PAL with a copy to nesdis.data.access@noaa.gov. This address provides the OSPO Data Access Team a copy of the correspondence. Once the request is approved by the OSPO management the data will be delivered by the Product Distribution and Access (PDA) system.

The output products are intended for operational and scientific users. Table 0-2 provides information about the algorithms and products.

**Table 0-2 – NVPS Green Vegetation Fraction Products**

| Product Category | Algorithm | Products |
|---|---|---|
| **NDE Vegetation Products System (NVPS)** | Green Vegetation Fraction (GVF) subsystem | • NetCDF, Geotiff, and text output files containing all the derived variables of the VI product |

# 1. Introduction

## 1.1. Product Overview

Current numerical weather prediction models and land surface monitoring systems require real time, large-scale land surface information for modeling initialization and monitoring land cover change. Daily global observations of the Visible Infrared Imaging Radiometer Suite (VIIRS) onboard Joint Polar-orbiting Satellite System (JPSS) are an excellent data source for such information. Thus, the NOAA JPSS Land Team has developed a NDE Vegetation Production System (NVPS) to produce Vegetation Indices (VI) and Green Vegetation Fraction (GVF). The retrieval algorithm uses VIIRS red (I1), near-infrared (I2) and blue (M3) bands centered at 0.640 μm, 0.865 μm and 0.490 μm, respectively, to calculate the Enhanced Vegetation Index (EVI) and derive GVF from EVI. The initial input to the GVF algorithm is reflectances that have already been gridded within the Vegetation Index (VI) algorithm. This document describes the details of the GVF algorithm that is used for VIIRS GVF retrieval. To meet the data needs of NCEP and other potential users, GVF is produced as a daily rolling weekly composite at 4-km resolution (global scale) and 1-km resolution (regional scale). Details of these products are presented in sections 2 and 3 of this document.

GVF produces rolling weekly (meaning, output is produced every day covering seven days) final products, at 4km resolution for the entire world and 1km resolution for a region encompassing all of North America. GVF also produces intermediate output files for its own use in later executions. Six GVF output files are produced daily – two NetCDF files, two tif image files, and two text files. The file pairs are generated for the global and regional scales.

## 1.2. Algorithm Overview

The NVPS will produce daily rolling weekly Green Vegetation Fraction. The GVF output files include a 0.009° (1-km) GVF regional file, and a 0.036° (4-km) GVF global file, both in NetCDF4 format. The daily rolling weekly production scheme means that the GVF products are derived from VIIRS input data from the past 7 days, but the output is generated every day. Since both VI and GVF use daily gridded surface reflectance as intermediate data, the GVF algorithm starts from reading in the daily gridded surface reflectance.

The GVF system generates daily rolling weekly Green Vegetation Fraction through the following steps:

**Step 1**: A gridded daily surface reflectance map including surface reflectances in bands I1, I2, and M3 produced by the VI unit is read in to the GVF unit, and the gridded reflectances are screened according to a cloud detection algorithm and a solar zenith angle threshold. At the end of a 7-day period, the daily surface reflectance maps of the 7 days are composited to produce a weekly surface reflectance map using the MVA-SAVI compositing algorithm, which selects, at each GVF grid point (pixel), the observation with maximum view-angle adjusted SAVI value in the 7-day period. The 7-day compositing is conducted daily using data in the previous 7 days as input data, which is called daily rolling weekly compositing. Quality flag information of composited pixels is saved.

EVI is then calculated from the daily rolling weekly composited VIIRS surface reflectance data in bands I1, I2 and M3.

**Step 2**: High frequency noise in EVI is reduced by applying a 15-week digital smoothing filter on EVI.

**Step 3**: First take the average of the previous 7 days worth of smoothed EVI data and then calculate GVF at each grid with the fine resolution (0.003 degree). GVF is calculated by comparing the smoothed EVI against the global maximum and minimum EVI values assuming a linear relationship between EVI and GVF.

**Step 4**: GVF is aggregated to 0.009 degree and 0.036 degree resolutions for output maps. Potential gaps on the output maps at high latitudes are filled using monthly VIIRS GVF climatology.

The GVF product is smoothed and therefore no quality flags are provided in the output file. For detailed information about the GVF algorithm, see the GVF Algorithm Theoretical Basis Document (https://www.ospo.noaa.gov/products/documents/GVF_ATBD_V4.1.pdf).

## 1.3. Interfaces Overview

Before reviewing this System Maintenance Manual (SMM), please request the live **master NCCF SMM** (refer to ***System Maintenance Manual - NESDIS Common Cloud Framework (NCCF)***) from the OSPO PALs in Table 1-2. The NCCF system overview is described in the **master NCCF SMM: NCCF Description and Overview (or Document Object: 4,5).**

# 2. Hardware

## 2.1. Hardware Description

The hardware is described in the **master NCCF SMM: NCCF Description and Overview (or Document Object: 4,5): Infrastructure**.

## 2.2. Operating System

The NCCF operating system description can be found in the **master NCCF SMM: Operating System (or Document Object: 64).**

## 2.3. System Requirements

The system requirements and timing information for each product are listed below:

**System Requirements:**

| | |
|---|---|
| CPU | Configurable; 4 Recommended |
| Memory | 2 GB per CPU utilized; 8 GB Recommended |

**Timing Information:**

| CPU Time (seconds) | Elapsed Time (h:mm:ss) |
|---|---|
| 26,973 | 2:56:53 |

Note: timing information is acquired from a run using four parallel processes.

### 2.3.1. Storage Requirements

The amount of required storage will depend on the number of files being processed. The individual NetCDF output files are usually about 13MB in size for the global coverage files and 63MB for regional files. The tif global files are about 6MB, and the corresponding regional files roughly 35MB. The text and log files are less than 1MB in size.

Typical input file sizes for the VI-SR files (see the production rules included with this delivery) range from 20MB to 500MB. GVF-EVI file sizes vary between 1MB and 50MB.

The delivery tar gzip files are as follows: 57MB for CODE, 183GB for DATA, and several MB for DOCS.

### 2.3.2. Computer Resource Requirements

The required libraries and utilities will be included in the Docker container.

### 2.3.3. Communication Needs

The target system for the NVPS GVF package is the NESDIS Common Cloud Framework (NCCF). There are no special bandwidth or communication issues associated with NVPS GVF. The communication needs of the processing system must be sufficient to meet the processing requirements described throughout this document.

# 3. Software

This section describes the system-level software elements that are invoked by the OSPO production system. Next, this section describes the source code and system files delivered to OSPO. These files are organized into subdirectories. The contents of each subdirectory are identified and their purpose is explained.

## 3.1. Software Description

The NVPS GVF software mainly consists of C++ program files, and yaml and Python scripts. These files and scripts are listed below, along with the subdirectory in which they are contained. A description of the directory structure can be found in section **Error! Reference source not found.**.

**aggGVF/**
aggGVF_main.cpp
error.cpp
gvfclimat.cpp

GVF.cpp
nc4GVF.cpp
NVPSUtil.cpp

**bin2tiff/**
error.cpp
GVFtiff.cpp
GVFtiff_main.cpp

**calcGVF/**
calcGVF_main.cpp
error.cpp
EVI.cpp
GVF.cpp
LandMask.cpp
NVPSUtil.cpp

**common/**
AllGranule.cpp
error.cpp
EVI.cpp
GeoLoc.cpp
GVF.cpp
GVFUtil.cpp
LandMask.cpp
NVPSUtil.cpp
SRTile_GVF.cpp
SRTile_VIv2r1.cpp
SurfRefl.cpp
WaterMask.cpp

**TileLandMask/**
error.cpp
gen_TileLandMask_main.cpp
LandMask.cpp
NVPSUtil.cpp
WaterMask.cpp

**TSsmooth/**
error.cpp
EVI.cpp
filter.cpp
GVFUtil.cpp
LandMask.cpp

NVPSUtil.cpp
TSsmooth_main.cpp
WaterMask.cpp

**WeeklyComposite/**
error.cpp
EVI.cpp
LandMask.cpp
NVPSUtil.cpp
SRTile.cpp
WeeklyComposite_main.cpp

**config_files/GVF/**
application_info.yaml
docker_info.yaml
GVF_PCF_constants.pcf

## 3.2. Directory Description

The NVPS GVF CCAP delivery consists of 3 gzip'd tar files:

- NVPS_GVF_CODE_202212.tar.gz
- NVPS_GVF_DOCS_202212.tar.gz
- NVPS_GVF_DATA_202212.tar.gz

Once unpacked, there will be four subdirectories immediately following the parent directory:

- wrappers - contains the science code
- nvps - contains ancillary data for GVF algorithm and a sample test case
- GVF_STATIC_ATA - contains Python scripts and configuration files to launch and run the docker
- DOCS - contains the documentation for this CCAP

Using $base to denote the parent directory, the subdirectories are organized as thus:

$base/

wrappers/

config_files/GVF/

dockerfiles/GVF/

scripts/

ccap_utils/

common_utils/

nvps/

build/

build_scripts/

/VI_CODE_NDE/

compile/

GVFEXEDIR/

run/

code/

aggGVF/

bin2diff/

calcGVF/

common/

TileLandMask/

TSsmooth/

WeeklyComposite/

GVF_STATIC_DATA/

gvfclimat/

watermask/

Here is the Docker Information:

1. Packages/Libraries installed in the docker image
    a. python 3.9.16
        i. PyYAML==6.0
2. Image information
    a. Base Image: centos:7
    b. Image Size: 595 MB
    c. Repository Name: 754153872510.dkr.ecr.us-east-1.amazonaws.com/assistt-centos7-python3-nvps-gvf
    d. Tag Name: v1.0

## 3.3. Source Code Description

The source code, written in C++, carries out the NVPS GVF algorithm steps as described in section **Error! Reference source not found.**.

# 4. Normal Operations

## 4.1. System Control

Several yaml and Python scripts are responsible for the execution of NVPS VI. The scripts begin the process by setting up the parameters, deciding what gets processed, and running the algorithm.

### 4.1.1. System Control Files

The wrapper scripts and configuration files are described below:

*docker_info.yaml* – 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.

*application_info.yaml* – Guides the inner script, namely which satellite and time should be processed. It also contains other inputs required by the VI algorithm.

*GVF_inner_wrapper.py* – The main entry point to the algorithm. This script takes a single argument, which should be the full path to the application info YAML file.

*launch.py* – Starts the docker container and runs the nvps_vi script to process the data pointed to by the application_info and docker_info configuration files.

### 4.1.2. Processing Controls

The docker_info and application_info yaml files require a series of arguments to successfully execute. Here's a list of these arguments, and then examples of the yaml files follows:

- Satellite: this parameter is used only in constructing the name of the docker container that is launched. As such, there is no strict requirement that it be set to any particular string, although it is strongly recommended it be set in an informative manner. (GVF_inner_wrapper.py will automatically detect the satellite from the provided input data and will throw an exception if the data cannot be identified.)
- Case ID: This parameter is also used in constructing the name of the docker container that is launched. As such, there is not strict requirement that it be set to any particular string; in the operational environment it is recommended that this string include the HPC job ID.
- Directory_Containing_GVF_Wrapper_Script: The full path to the directory in which GVF_inner_wrapper.py is stored as it appears on the host machine (meaning, outside of the docker container).
- GVF_Inner_Wrapper_Script: The full path to GVF_inner_wrapper.py as it appears on the host machine.
- Application_Info_YAML_File: The full path to the application info YAML file, to be used as the argument to the inner wrapper script.
- container_memory_allotment: How much memory should be allocated to the docker container. This value should be set to the product of two gigabytes (2G) with the number of parallel processes used, which is specified in the application info YAML file. This field can be any string that is valid to pass to docker run's --memory option.

**docker_info.yaml**

```
arguments:
 - &satellite NOAA20
 - &case_id "20200416"
 - &Directory_Containing_GVF_Wrapper_Script
/share/data/assistt/jonathan.hansford/NVPS/NVPS_GVF_Final_CCAP_202212/wrappers/scripts
 - &GVF_Inner_Wrapper_Script
/share/data/assistt/jonathan.hansford/NVPS/NVPS_GVF_Final_CCAP_202212/wrappers/scripts/GVF
_inner_wrapper.py
 - &Application_Info_YAML_File
/share/data/assistt/jonathan.hansford/NVPS/NVPS_GVF_Final_CCAP_202212/wrappers/config_files
/GVF/application_info.yaml
 - &input_dir
/share/data/assistt/jonathan.hansford/NVPS/NVPS_GVF_Final_CCAP_202212/GVF_TEST_DATA/i
nput
 - &output_dir
/share/data/assistt/jonathan.hansford/NVPS/NVPS_GVF_Final_CCAP_202212/GVF_TEST_DATA/
output
 - &log_dir
/share/data/assistt/jonathan.hansford/NVPS/NVPS_GVF_Final_CCAP_202212/GVF_TEST_DATA/l
ogs
 - &GVF_EXE_INSTALLATION_DIR
/share/data/assistt/jonathan.hansford/NVPS/NVPS_GVF_Final_CCAP_202212/nvps/GVF_CODE_N
DE
 - &GVF_STATIC_DATA_DIR
/share/data/assistt/jonathan.hansford/NVPS/NVPS_GVF_Final_CCAP_202212/GVF_STATIC_DAT
A
 - &GVF_PCF_Constants_File
/share/data/assistt/jonathan.hansford/NVPS/NVPS_GVF_Final_CCAP_202212/wrappers/config_files
/GVF/GVF_PCF_constants.pcf
   # GVF consumes ~1.79 GB per parallel process used; allocate 2GB per process to be safe.
   # Note that the number of parallel processes to use is set in application_info.yaml.
 - &container_memory_allotment 8G

docker_info:
  image: 754153872510.dkr.ecr.us-east-1.amazonaws.com/assistt-centos7-python3-nvps-gvf:v1.0
  python_info:
    python: python3
    script: *GVF_Inner_Wrapper_Script
    application_yaml: *Application_Info_YAML_File

  mount_pairs:
    - src: *Directory_Containing_GVF_Wrapper_Script
```

```
    dst: *Directory_Containing_GVF_Wrapper_Script

  - src: *Application_Info_YAML_File
    dst: *Application_Info_YAML_File

  - src: *input_dir
    dst: *input_dir

  - src: *output_dir
    dst: *output_dir

  - src: *log_dir
    dst: *log_dir

  - src: *GVF_EXE_INSTALLATION_DIR
    dst: *GVF_EXE_INSTALLATION_DIR

  - src: *GVF_STATIC_DATA_DIR
    dst: *GVF_STATIC_DATA_DIR

  - src: *GVF_PCF_Constants_File
    dst: *GVF_PCF_Constants_File

 docker_args:
   memory: *container_memory_allotment
   memory-swap: *container_memory_allotment

 container_name:
   sat: *satellite
   proj: GVF
   caseid: *case_id
```

**application_info.yaml**

```
# This YAML file is intended to be passed as the argument to
# GVF_inner_wrapper.py, the inner wrapper script for the GVF CCAP.
# The value of each parameter in the "arguments" section should be set
# appropriately; some parameters will be different for each execution!

arguments:
  # The date, stated in the format yyyymmdd, for which GVF is being
  # executed. The end date of the final output files will be this
  # date.
- &running_for_date 20200416
```

```
  # The absolute path (as it appears inside the docker container,
  # assuming that GVF_inner_wrapper.py is being executed in a
  # container) of the directory containing all of the input files to
  # be used for this run. (The required and optional VI-SR*.h5,
  # GVF-EVI*.h5, and GVF-ASEVI-P1*.h5 should all be provided in this
  # one directory, and should not be sorted into subdirectories).
- &input_dir
/share/data/assistt/jonathan.hansford/NVPS/NVPS_GVF_Final_CCAP_202212/GVF_TEST_DATA/i
nput

  # After a successful execution of GVF, all output files produced,
  # including both "intermediate" and "final" outputs, will be moved
  # to this directory, which should be specified as an absolute path
  # as it appears inside the docker container (assuming that
  # GVF_inner_wrapper.py is being executed in a container). This
  # directory will be created if it does not already exist, but note
  # that if the algorithm is being executed inside a Docker container,
  # this directory must be created before the container is launched if
  # it is to be mounted directly into the container.
- &output_dir
/share/data/assistt/jonathan.hansford/NVPS/NVPS_GVF_Final_CCAP_202212/GVF_TEST_DATA/
output

  # After execution, the log files generated will be moved to this
  # directory, which should be specified as an absolute path as it
  # appears inside the docker container (assuming that
  # GVF_inner_wrapper.py is being executed in a container). This
  # directory will be created if it does not already exist, but note
  # that if the algorithm is being executed inside a Docker container,
  # this directory must be created before the container is launched if
  # it is to be mounted directly into the container.
- &log_dir
/share/data/assistt/jonathan.hansford/NVPS/NVPS_GVF_Final_CCAP_202212/GVF_TEST_DATA/l
ogs

  # The bulk of the GVF processing will be done in this directory,
  # which should be specified as an absolute path as it appears inside
  # the docker container (assuming that GVF_inner_wrapper.py is being
  # executed in a container). During the course of execution numerous
  # links, temporary files, and output files (multiple gigabytes in
  # total) will be created in this directory. This directory will be
  # created if it does not already exist, but note that if the
  # algorithm is being executed inside a Docker container and this
```

```
  # directory is to be mounted into the container, it must be created
  # before the container is launched.
  # This directory can be a location completely internal to the docker
  # container (meaning, not under the mount point of some bind
  # volume), or a location that is mounted into the container.
  # Note that because of the manner in which the GVF bash scripts use
  # "grep" to read some internally generated parameter files, setting
  # this parameter to certain values may cause the algorithm to fail.
  # In particular, do not set this parameter to a path including the
  # substring "VIIRS_GVF_WORKDIR".
- &VIIRS_GVF_WORKDIR /home/WORKING_DIR

  # The global attribute "production_site" of the netCDF4 output files
  # created will be set to this value.
- &production_site "STAR PPM"

  # The global attribute "production_environment" of the netCDF4
  # output files created will be set to this value.
- &production_environment DEV

  # The absolute path (as it appears inside the docker container,
  # assuming that GVF_inner_wrapper.py is being executed in a
  # container) of the directory containing the GVF executables and
  # bash scripts. (The directory named "GVF_CODE_NDE", containing the
  # subdirectories "GVFEXEDIR" and "run".)
- &GVF_EXE_INSTALLATION_DIR
/share/data/assistt/jonathan.hansford/NVPS/NVPS_GVF_Final_CCAP_202212/nvps/GVF_CODE_N
DE

  # The absolute path (as it appears inside the docker container,
  # assuming that GVF_inner_wrapper.py is being executed in a
  # container) of the directory containing the static data required by
  # the GVF algorithm. (This directory must contain the subdirectories
  # "gvfclimat" and "watermask".)
- &GVF_STATIC_DATA_DIR
/share/data/assistt/jonathan.hansford/NVPS/NVPS_GVF_Final_CCAP_202212/GVF_STATIC_DAT
A

  # The absolute path (as it appears inside the docker container,
  # assuming that GVF_inner_wrapper.py is being executed in a
  # container) of the file "GVF_PCF_constants.pcf".
- &GVF_PCF_Constants_File
/share/data/assistt/jonathan.hansford/NVPS/NVPS_GVF_Final_CCAP_202212/wrappers/config_files
/GVF/GVF_PCF_constants.pcf
```

```
  # Some portions of the GVF algorithm can execute in parallel; this
  # value controls the maximum number of processes that will be
  # executed simultaneously in the parallel sections.
 - &number_of_parallel_processes 4

application_info:
  outdir: *output_dir
  logdir: *log_dir
  algorithm_spec:
  - name: running_for_date
    value: *running_for_date
  - name: VIIRS_GVF_WORKDIR
    value: *VIIRS_GVF_WORKDIR
  - name: GVF_EXE_INSTALLATION_DIR
    value: *GVF_EXE_INSTALLATION_DIR
  - name: GVF_STATIC_DATA_DIR
    value: *GVF_STATIC_DATA_DIR
  - name: GVF_PCF_Constants_File
    value: *GVF_PCF_Constants_File
  - name: number_of_parallel_processes
    value: *number_of_parallel_processes
  - name: input_dir
    value: *input_dir
  - name: production_site
    value: *production_site
  - name: production_environment
    value: *production_environment
```

## 4.2. Installation

### 4.2.1. Installation Items

For more information concerning the installation items created after unpacking the NVPS GVF delivery package, please refer to section **Error! Reference source not found.**.

### 4.2.2. Compilation Procedures

See the **master NCCF SMM: Compilation Procedures and Build Procedures (or Document Object: 76,78)**

### 4.2.3. Installation Procedures

## 4.3. Configuration Procedures

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

See the **master NCCF SMM: Installation Procedures (or Document Object: 77)**

### 4.3.1. Production Rules

Please see the Production Rules document provided with this CCAP delivery.

## 4.4. Operations Procedures

### 4.4.1. Normal Operations

Please refer to the master NCCF SMM: Procedures for Normal Operations (or Document Object 10)

## 4.5. Distribution

### 4.5.1. Data Transfer/Communications

Please refer to the master NCCF SMM: Data Transfer/Communications and Data Preparation (or Document Object 73, 81)

### 4.5.2. Distribution Restrictions

There are no restrictions on NVPS_GVF product distribution.

### 4.5.3. Product Retention Requirements

### 4.5.4. External Product Tools

External software is not provided for viewing/analyzing NVPS GVF.

# 5. Monitoring and Maintenance

## 5.1. Job Monitoring

The operational system and the NVPS GVF algorithm log files produced with the execution can be used to monitor the jobs.

### 5.1.1. Product Monitoring Visualization and Alarms

Product quality is monitored using the NCCF Product Monitoring Tool at https://nccf.espc.nesdis.noaa.gov/mtool/index.html.

Users can use this page to monitor hourly summaries of the NVPS GVF product quality based on parameter thresholds determined by the PAL.

The NCCF NVPS GVF product webpage at https://www.ospo.noaa.gov/products/land/gvf/ can also be used to view cloud composite images of select parameters in near real-time. These images are updated daily.

NCCF PG Product Latency and Product Missing Alarms can be found at https://us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarmsV2

## 5.2. Data Signal Monitoring

## 5.3. Product Monitoring

In addition to the log files, quality flags, variables, and metadata information can be used to evaluate the NVPS GVF products.

### 5.3.1. Unit Test Plans

### 5.3.2. Internal Product Tools

### 5.3.3. Performance Statistics

### 5.3.4. Product Monitoring

### 5.3.5. Product Criticality

## 5.4. Maintenance

### 5.4.1. Monitoring

See the **master NCCF SMM: Maintenance Utilities (or Document Object: 84)**

#### 5.4.1.1. Ingest Monitoring

See the **master NCCF SMM: Data Transfer/Communications and Data Preparation (or Document Object: 73 and 81)**

#### 5.4.1.2. Production Job Monitoring

#### 5.4.1.3. Product Distribution Monitoring

### 5.4.2. Science Maintenance

Product quality monitoring is performed by the OSPO Product Quality Monitoring System and the OCS developers. OCS and OSPO personnel communicate regularly to discuss any potential data quality issues, formulate updates to the code, and schedule updates to the package's science code.

### 5.4.3. Library Maintenance

See the **master NCCF SMM: Library Maintenance (or Document Object: 71)**

### 5.4.4. Special Maintenance Procedures

See the **master NCCF SMM: Special Maintenance Procedures (or Document Object: 72)**

### 5.4.5. Maintenance Utilities

See the **master NCCF SMM: Maintenance Utilities (or Document Object: 84)**

## 5.5. Program Backup Procedures

See the **master NCCF SMM: Data Recovery Procedures and Program Recovery Procedures (or Document Object: 89, 90)**

# 6. Troubleshooting

## 6.1. Program Diagnosis and Recovery

See the **master NCCF SMM: Problem Diagnosis and Recovery Procedures (or Document Object: 82)**

### 6.1.1. Quality Control Output

### 6.1.2. Error Correction

See the **master NCCF SMM: Error Correction – Warnings and Messages for Systems and Error Codes, Menus and Navigation (or Document Object: 43,44,45)**

### 6.1.3. Problem Diagnosis and Recovery Procedures

See the **master NCCF SMM: Problem Diagnosis and Recovery Procedures (or Document Object: 82)**

#### 6.1.3.1. High-Level Errors

#### 6.1.3.2. Low-Level Errors

### 6.1.4. Data Recovery Procedures

See the **master NCCF SMM: Data Recovery Procedures (or Document Object: 89)**

### 6.1.5. Program Recovery Procedures

## 6.2. Application Shutdown and Restart

See the **master NCCF SMM: Program Recovery Procedures (or Document Object: 90)**

### 6.2.1. Application Shutdown Procedures

See the **master NCCF SMM: Application Shutdown Procedures (or Document Object: 94)**

### 6.2.2. Application Restart Procedures

See the **master NCCF SMM: Application Restart Procedures (or Document Object: 92)**

## 6.3. System Shutdown and Restart

See the **master NCCF SMM: Reboot Procedures, Restart Procedures and Shutdown Procedures (or Document Object: 83, 93, 95)**

### 6.3.1. System Shutdown Procedures

### 6.3.2. System Restart Procedures

### 6.3.3. System Reboot Procedures

# 7. Appendix

## 7.1. Data Flow

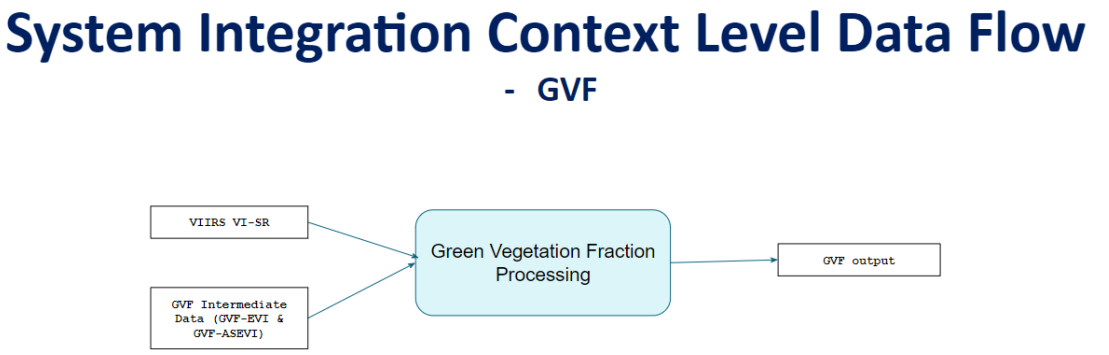


**Figure 7-1 - Context Level Data Flow**

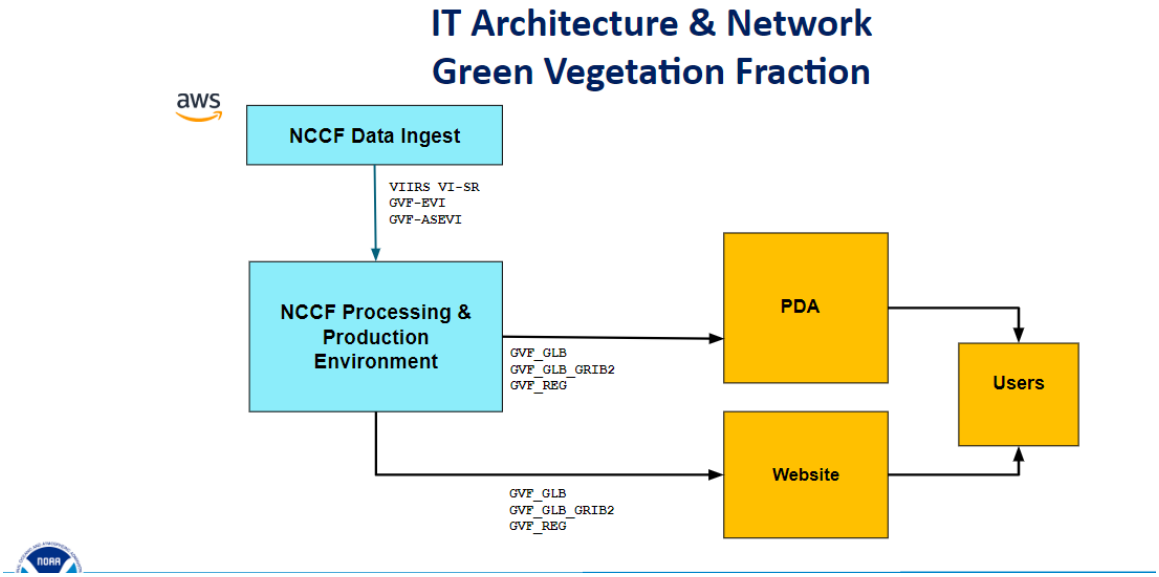


**Figure 7-2 - IT Architecture & Network**

## 7.2. Input Data Files

This version of GVF may process SNPP, NOAA-20, or NOAA-21 data. Only data from a single satellite should be provided as the input for a single execution. A sample input filename is:

VI-SR-J01_s<YYYYMMDD>_e<YYYYMMDD>_h<HR>v<VR>_cYYYYMMDDhhmmsss.h5

where

| | | |
|---|---|---|
| s<YYYYMMDD> | → | the start time in 4-digit year, 2-digit month, 2-digit day |
| e<YYYYMMDD> | → | the end time in 4-digit year, 2-digit month, 2-digit day |
| c<YYYYMMDDhhmmsss> | → | the creation time in 4-digit year, 2-digit month, 2-digit day, 2-digit hour, 2-digit minute, 2-digit second, 1 digit tenths of a second format associated with the file |
| VI-SR | → | intermediate product (IP) from the daily gridded surface reflectance products of VI (Vegetation Index) process |
| J01 | → | Satellite platform |
| h<HR> | → | horizontal range of [00, 19] |
| v<VR> | → | vertical range of [00, 09] |

## 7.3. Ancillary Data Files

Previous NVPS GVF intermediate file data can be used for current GVF processing, though this is optional. Two types of the intermediate data would be required - unsmoothed and smoothed GVF EVI data. The unsmoothed GVF-EVI files are weekly composited files from the previous 14 weeks. The smoothed GVS-ASEVI-P1 files consist of 7 daily rolling IP EVI files after 15 weeks of smoothing.

## 7.4. Look Up Tables

Static ancillary data files are included in the package delivered to operations. Climatological data and the watermask are needed to generate the NVPS GVF products.

## 7.5. Intermediate Data Set Description

Static ancillary data files are included in the package delivered to operations.

## 7.6. Output Data Set Description

The 6 products created on a daily basis from the NVPS GVF product system are in text, NetCDF and tif (image) formats. The filenames are shown in Table 7-1.

**Table 7-1 - NVPS GVF Product Output Files**

| File | Description | Format | Size/file |
|---|---|---|---|
| GVF-WKL-REG _vxry_sid_s[YYYYMMDD1]_e[YYYYMMDD7] _c[YYYYMMDDhhmmsss].nc | This is the weekly regional GVF product | netCDF4 | Typical file size 65 MB. |
| GVF-WKL-GLB _vxry_sid_s[YYYYMMDD1]_e[YYYYMMDD7] _c[YYYYMMDDhhmmsss].nc | This is the weekly global GVF product | netCDF4 | Typical file size 11 MB. |
| GVF-WKL-REG _vxry_sid_s[YYYYMMDD1]_e[YYYYMMDD7] _c[YYYYMMDDhhmmsss].tif | Browse image of the regional GVF product | Geotiff | Typical file size 34 MB |
| GVF-WKL-GLB _vxry_sid_s[YYYYMMDD1]_e[YYYYMMDD7] _c[YYYYMMDDhhmmsss].tif | Browse image of the weekly global GVF product | Geotiff | Typical file size 5 MB |
| GVF-WKL-REG _vxry_sid_s[YYYYMMDD1]_e[YYYYMMDD7] _c[YYYYMMDDhhmmsss]_stat.txt | Statistics file of the weekly regional GVF product for monitoring purposes | text | Typical file size 10 KB |
| GVF-WKL-GLB _vxry_sid_s[YYYYMMDD1]_e[YYYYMMDD7] _c[YYYYMMDDhhmmsss]_stat.txt | Statistics file of the weekly global GVF product for monitoring purposes | text | Typical file size 10 KB |

Descriptions of the lettering used in the output filenames are listed in Table 7-2.

**Table 7-2 - NVPS GVF Output Files Standard Name Description**

| Sequence | Description |
|---|---|
| GVF | Green Vegetation Fraction |
| VI | Vegetation Indices (NDVI, EVI) |
| NDVI | Normalized Difference Vegetation Index |
| EVI | Enhanced Vegetation Index |
| DLY | Daily (1-day temporal scale) |
| WKL | Weekly (7-day temporal resolution) |
| BWKL | Biweekly (16-day temporal resolution, in term of conventions) |
| GLB | Global (spatial resolution: 4-km) |
| REG | Regional (spatial resolution:1-km) |
| TOA | Top of Atmosphere |
| TOC | Top of Canopy |
| vxry | Version (e.g., v2r2) |

| Sequence | Description |
|---|---|
| sid | Indicates the observations from JPSS-01 |
| s | start (data observation time) |
| e | end (data observation time) |
| c | current (data processing time) |
| YYYYMMDD | 4-digit year, 2-digit month, and 2-digit day |
| hhmmsss | 2-digit hour, 2-digit minute, 2-digit second, and 1-digit fractional second |
| .nc | netCDF4 file |
| .tif | GeoTiff image file |
| stat.txt | Text file stored statistics analysis results |

Examples of the output filenames are:

GVF-WKL-GLB_v3r0_j01_s20200410_e20200416_c202301041524440.nc

GVF-WKL-REG_v3r0_j01_s20200410_e20200416_c202301041527160.nc

GVF-WKL-GLB_v3r0_j01_s20200410_e20200416_c202301041524440.tif GVF-WKL-REG_v3r0_j01_s20200410_e20200416_c202301041527160.tif

GVF-WKL-GLB_v3r0_j01_s20200410_e20200416_c202301041524440_stat.txt GVF-WKL-REG_v3r0_j01_s20200410_e20200416_c202301041527160_stat.txt

## 7.7. Archive Data Files

The NetCDF output files for NVPS GVF will be archived at NCEI.

## 7.8. References

Vargas, M., Miura, T., Shabanov, N., & Kato, A. (2013). An initial assessment of Suomi NPP VIIRS vegetation index EDR. *Journal of Geophysical Research-Atmospheres*, 118, 12301-12316.

NESDIS/STAR (2021): Vegetation Index (VI) Product Algorithm Theoretical Basis Document (ATBD) v2.1

NESDIS/STAR (2021): Green Vegetation Fraction (GVF) Product Algorithm Theoretical Basis Document (ATBD) v4.1

NESDIS/STAR (2022): NDE Vegetation Products System (NVPS) External Users Manual (EUM) v2.2

NESDIS/STAR (2022): NDE Vegetation Products System (NVPS) System Maintenance Manual (SMM) v2.2

NESDIS/STAR (2022): NVPS GVF Product Delivery memo, Readme file, PCF_PSF doc, and Production Rules doc

NESDIS/STAR (2022): Normalized Vegetation Products System (NVPS) Vegetation Index (VI) System Maintenance Manual

NESDIS/STAR (2022): Normalized Vegetation Products System (NVPS) Vegetation Index (VI) External Users' Manual

NESDIS/STAR (2022): Normalized Vegetation Products System (NVPS) Green Vegetation Fraction (GVF) External Users' Manual

# 8. Acronyms

| Acronym | Definition |
|---|---|
| ASSISTT | Algorithm Scientific Software Integration and System Transition Team |
| ATBD | Algorithm Theoretical Basis Document |
| CCAP | Cloud Containerized Algorithm Package |
| CM | Configuration Management |
| DDS | Data Distribution System |
| ESPC | Environmental Satellite Processing Center |
| EVI | Enhanced Vegetation Index |
| GVF | Green Vegetation Fraction |
| JPSS | Joint Polar Satellite System |
| NCCF | NESDIS Common Cloud Framework |
| NCEI | National Centers for Environmental Information |
| NDE | NPP Data Exploitation |
| NESDIS | National Environmental Satellite, Data, and Information Service |
| NetCDF | Network Common Data Form |
| NOAA | National Oceanic and Atmospheric Administration |
| NWS | National Weather Service |
| OCS | Office of Common Services |
| OSPO | Office of Satellite and Product Operations |
| PAL | Product Area Lead |
| PDA | Product Distribution and Access |
| PG | Product Generation |
| QA | Quality Assurance |
| SMM | System Maintenance Manual |
| S-NPP | Suomi National Polar-orbiting Partnership |
| STAR | Center for Satellite Applications and Research |
| V&V | Verification and Validation |
| VI | Vegetation Index |
| VIIRS | Visible and Infrared Imaging Radiometer Suite |