National Aeronautics and Space Administration
Goddard Earth Science Data Information and Services Center (GES DISC)

README Document for NASA MEaSUREs Precipitation Fundamental Climate Data Record

Last Revised June 30, 2022

Goddard Earth Sciences Data and Information Services Center (GES DISC)
http://disc.gsfc.nasa.gov
NASA Goddard Space Flight Center
Code 610.2
Greenbelt, MD 20771 USA

Prepared By:

David Silberstein
GES DISC
GSFC Code 610.2

Chris Kidd
NASA/GSFC Code 612, Greenbelt, MD 20771, and University of Maryland, College Park, MD 20740

June 15, 2022

Reviewed By:

David Silberstein
GES DISC
GSFC Code 610.2

June 30, 2022

Goddard Space Flight Center
Greenbelt, Maryland

Revision History

Revision Date	Changes	Author
Initial Version 06/15/22		David Silberstein

Table of Contents

This document provides basic information for using NASA's **M**aking **Ea**rth **S**cience Data Records for **U**se in **R**esearch **E**nvironment**s** (MEaSUREs) Precipitation Fundamental Climate Data Record products.

The NASA MEaSUREs Precipitation Fundamental Climate Data Record products have been produced for the purpose of establishing a record which exploits the observations made by passive microwave imaging radiometers between 1978 and 2020 to generate an ensemble of precipitation retrievals. The satellite/sensors used to create these products are defined in the following table:

Satellite	**Sensor**	**Start (month/year)**	**End (month/year)**	**Length of record (months)**
Nimbus-7	SMMR	01/1979	08/1987	104
DMSP-F08	SSM/I	07/1987	12/1991	54
DMSP-F10	SSM/I	12/1990	11/1997	84
DMSP-F11	SSM/I	12/1991	05/2000	102
DMSP-F13	SSM/I	05/1995	11/2009	175
DMSP-F14	SSM/I	05/1997	08/2008	136
TRMM	TMI	12/1997	04/2015	209
DMSP-F15	SSM/I	02/2000	08/2006	79
AQUA	AMSR-E	06/2002	10/2011	113
DMSP-F16	SSMIS	11/2005	*12/2020*	*192*
DMSP-F17	SSMIS	03/2008	*12/2020*	*154*
DMSP-F18	SSMIS	03/2010	*12/2020*	*130*
GCOM-W1	AMSR-2	07/2012	*12/2020*	*102*
GPM	GMI	03/2014	*12/2020*	*82*
DMSP-F19	SSMIS	12/2014	02/2016	15

1.1 Dataset/Mission Instrument Description

A much deeper dive into the algorithms used in this effort can be found in the Algorithm Theoretical Basis Document (ATBD) available through the documentation link on the particular product's landing page.

1.2 Data Disclaimer

The version of this ensemble described here is the initial version, (V01E) and while every effort has been made to produce an error-free product, a number of unknown issues may present in the product. Where such issues are found they will be noted and rectified in the implementation of future releases.

1.2.1 Contact Information

Chris Kidd - NASA/GSFC Code 612, Greenbelt, MD 20771, and University of Maryland, College Park, MD 20740

2.0 Data Organization

The data consist of orbital swath data (level 2) with one file(granule) per orbit. The swath has dimensions of nscan x npixel with nscan representing the along track direction and npixel representing the cross track direction.

2.1 File Naming Convention

File names are of the form:

TRMM_TMI_FCDR2021_L2_V1_yyyymmdd-Shhmmss-Ehhmmss.orbnum.V01E.nc

Where:

- yyyymmdd = year, month, day of orbit
- Shhmmss = GMT hour, minute, second of orbital starting time
- Ehhmmss = GMT hour, minute, second of orbital ending time
- orbnum = orbital number

Filename example: TRMM_TMI_FCDR2021_L2_V1_20150101-S054034-E071215.097566.V01E.nc

2.2 File Format and Structure

Data set files are in netCDF-4 format. NetCDF is a set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data that was developed by UCAR/Unidata (http://doi.org/10.5065/D6H70CW6) https://www.unidata.ucar.edu/software/netcdf/.

3.0 Data Contents

For each file, the data is organized into 15 groups, one group per algorithm. The algorithms in the file are denoted by three character abbreviations for the algorithm names:

AD1 – “Adler 1”
BA0 – “Bristol Algorithm 0”
BA1 – “Bristol Algorithm 1”
BA3 – “Bristol Algorithm 3”
FE1 – “Ferraro 1”
FE2 – “Ferraro 2”
FE3 – “Ferraro 3”
FE4 – “Ferraro 4”
FR1 – “Ferriday 1”
FR2 – “Ferriday 2”
IO1 – “Iowa Algorithm 1”
NR1 – “Naval Research 1”
NR2 – “Naval Research 2”
PR1 – “Prabhakara 1”
SC2 – “Schlussel 2”

Each algorithm is described in great detail in the ATBD. Each algorithm group contains:

rain_rate
quality score
algorithm flag
processing flag

At the base level of the file:

time expressed in separate variables as year, month, dayofmonth, hour, minute and second as well as time expressed as a string called scan_datetime in YYYY-MM-DDTHH:MM:SS.SSZ format

along with

latitude
longitude
geophysical_flag

3.1 Data Set Attributes (File Metadata)

In addition to SDS arrays containing variables and dimension scales, global metadata is also stored in the files. Some metadata are required by standard conventions, some are present to meet data provenance requirements and others as a convenience to users of these products. A summary of global attributes present in all files is as follows:

Convention = CF-1.6
DOI = 10.5067/MEASURES/TRMM/TMI/DATA201
DOIauthority = http://dx.doi.org
EndDirection = D

EndLatitude = -35.10298

EquatorCrossingDate = 2015-01-01

EquatorCrossingLongitude = 71.02108

EquatorCrossingTime = 06:03:30

Format = NetCDF-4

GranuleID = TRMM_TMI_FCDR2021_L2_V1_20150101-S054034-E071215.097566.V01E.nc

LongName = NASA MEASURES Precipitation Ensemble based on TMI TRMM NASA PPS L1C V05 Tbs

OrbitNumber = 97566

ProcessingLevel = Level 2

ProductionDateTime = 2021-05-26T18:15:59.000Z

RangeBeginningDate = 2015-01-01

RangeBeginningTime = 05:40:35.000000

RangeEndingDate = 2015-01-01

RangeEndingTime = 07:12:15.000000

SatelliteName = TRMM

ShortName = PRECIP_TMI_TRMM

Source = TMI

StartDirection = A

StartLatitude = -35.10303
VersionID = 1

3.2 Geolocation Fields

Global Attribute	Description	Type
latitude	Orbital latitude	32-bit floating-point
longitude	Orbital longitude	32-bit floating-point
scan_datetime	YYYY-MM-DDTHH:MM:SS.SSZ formatted time	string

3.3 Dimensions

Variable	Description	Dimensions
scan_datetime	YYYY-MM-DDTHH:MM:SS.SSZ formatted time	Nscan x 23 (character string length)
latitude	Orbital latitude	Nscan x npixel
longitude	Orbital longitude	Nscan x npixel

4.0 Products/Parameters

4.1 Data Fields

Data Field Name	Long_Name/Description	Type	Dimensions	Undefined Value	Units
(3 character algorithm abbreviation)_ rain_rate	Rain retrieval for (algorithm name) algorithm	32-bit floating-point	Nscan x npixel	-9999.9	mm/ hour
(3 character algorithm abbreviation)_ quality_score	Quality score for (algorithm name) based upon correlation and bias-ratio vs. ERA-5 reanalysis	8-bit integer	Nscan x npixel	255	
(3 character algorithm abbreviation)_ algorithm_flag	Algorithm Flag for (algorithm name) rain retrieval	8-bit integer	Nscan x npixel		
(3 character algorithm abbreviation)_ processing_flag	Processing Flag for (algorithm name) rain retrieval	8-bit integer	Nscan x npixel		

4.2 Quality Control

For each algorithm group, there is a quality score provided for each data value. The basis of this quality score is defined in great detail in the ATBD available through the documentation link on the particular product's landing page.

5.0 Options for Reading the Data

5.1 Command Line Utilities

5.1.1 ncdump

The ncdump tool can be used as a simple browser for HDF data files, to display the dimension names and sizes; variable names, types, and shapes; attribute names and values; and optionally, the values of data for all variables or selected variables in a netCDF file. The most common use of ncdump is with the –h option, in which only the header information is displayed.

ncdump [-c|-h] [-v ...] [[-b|-f] [c|f]] [-l len] [-n name] [-d n[,n]] filename
Options/Arguments:
[-c] Coordinate variable data and header information
[-h] Header information only, no data
[-v var1[,...]] Data for variable(s) <var1>,... only data
[-f [c|f]] Full annotations for C or Fortran indices in data
[-l len] Line length maximum in data section (default 80)
[-n name] Name for netCDF (default derived from file name)
[-d n[,n]] Approximate floating-point values with less precision filename File name of input netCDF file
(https://www.unidata.ucar.edu/software/netcdf/workshops/2011/utilities/Ncdump.html)

5.1.2 HDFView

HDFView is a Java based graphical user interface created by the HDF Group which can be used to browse HDF files. The utility allows users to view all objects in an HDF file hierarchy which is represented as a tree structure. Additional information about HDFView can be found at https://support.hdfgroup.org/products/java/hdfview/ and for HDF at https://portal.hdfgroup.org/display/support

5.2 Tools/Programming

The product files can be read and queried using the NetCDF4 library and tools maintained by Unidata (http://www.unidata.ucar.edu/software/netcdf/). Support for reading NetCDF is offered in many programming languages, including Python, Matlab, IDL, C/C++ and Fortran. NetCDF4

files are legal HDF5 files with additional bookkeeping information managed by the NetCDF4 library. It is therefore possible to inspect and copy data out of the NetCDF4 files by using the HDF5 utilities and libraries maintained by the HDF Group (https://www.hdfgroup.org/products/hdf5_tools/index.html) or by using the HDF5 interface in your favorite programming language. However, the two libraries should not be considered fully interchangeable.

Matlab users should note that the Matlab NetCDF4 interface is currently (as of version R2017a) not able to read attributes that are string arrays, and will throw an exception if that is attempted.

5.2.1 Sample Python Code

To use Python to read data into arrays, and determine the shape as well as the min and max values of that array:

```
import xarray as xr
import numpy as np

filename="NIMBUS7_SMMR_FCDR2021_L2_V1_19800101_S083000-E101400.06000.V01E.nc"
f=xr.open_dataset(filename)
lat = f['latitude'][...].values
lon = f['longitude'][...].values
fg=xr.open_dataset(filename,group="FE3")
rr=fg["FE3_rain_rate"].values
np.shape(rr)
np.nanmin(rr)
np.nanmax(rr)
```

These data sets contain numerous flags; sample code to access them follows:

```
gpv=f["geophysical_flag"].values
test=np.where(gpv==1) # searching for bit 0 flag set
loc=list(zip(test[0],test[1]))
for c in loc:
   print (gpv[c],c)
```

To display bits for a given pixel, use bin -

```
bin(gpv[415,14])[2:][::-1]
```

where sequence is bit 0 first

and bit values are as follows for flag on:

0 1 2 3 4   5
1 2 4 8 16 32

Example:

output is 0111

so bits 1, 2 and 3 are on

0 1 2 3
0 2 4 8

and pixel value is 14.

You can also examine the specific bit settings for all pixels:

```
c=np.unpackbits(np.uint8(gpv),bitorder='little')
bit_len=int(np.size(c)/np.size(gpv))
```

In this case, pixel [415,14] is again selected -

```
along_track_pixel = 415
cross_track_pixel = 14


initial_bit = (along_track_pixel*np.shape(gpv)[1]+cross_track_pixel)*bit_len

print(c[initial_bit:initial_bit+bit_len])
```

will display the bits for this pixel

[0 1 1 1 0 0 0 0]

The strings of the variable “scan_datetime” can be easily broken down into arrays of component time variables by using the following code:

```
import xarray as xr
import numpy as np
from datetime import datetime
```

```
filename="TRMM_TMI_FCDR2021_L2_V1_20150101-S054034-E071215.097566.V01E.nc"
f=xr.open_dataset(filename)
sd = f['scan_datetime'][...].values
a=[]
for x in range (0,np.size(sd)):
    a.append(datetime.strptime(str(sd[x],"utf-8"),'%Y-%m-%dT%H:%M:%S.%fZ'))

a[0].date().year
a[0].date().month
a[0].date().day
a[0].time().hour
a[0].time().minute
a[0].time().second
```

6.0 GES DISC Data Services

If you need assistance or wish to report a problem:
Email: gsfc-dl-help-disc@mail.nasa.gov
Voice: 301-614-5224
Fax: 301-614-5268
Address:
Goddard Earth Sciences Data and Information Services Center NASA Goddard Space Flight Center Code 610.2 Greenbelt, MD 20771 USA

6.1 How To Articles

The GESDISC web site contains many informatiative articles under the "How To Section", "FAQ" (frequently asked questions), "News", "Glossary", and "Help" . A sample of these articles includes:

Earthdata Login for Data Access

How to Download Data Files from HTTPS Service with wget

Quick View Data with Panoply

How to Read Data in NetCDF Format with R

How to Read Data in HDF-5 or netCDF Format with GrADS

How to read and plot NetCDF MERRA-2 data in Python

How to Subset Level-2 Data

How to use the Level 3 and 4 Subsetter and Regridder