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

README Document for the Nimbus-4 Backscatter Ultraviolet Spectrometer (BUV) Level-1 Radiance and Housekeeping Data in Telemetry Units Product

BUVN4L1PDB

Last Revised 06/22/2017

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:

James E. Johnson | 10/15/2015
Name | Date
GES DISC
GSFC Code 610.2

Reviewed By:

Name | mm/dd/yyyy
Name | Date
GSFC Code xxx

Name | mm/dd/yyyy
Name | Date
GSFC Code xxx

Goddard Space Flight Center
Greenbelt, Maryland

Revision History

Revision Date	*Changes*	*Author*
10/15/2015	Original	James E. Johnson
06/22/2017	Updated URLs in sections 5 and 6	James E. Johnson

Table of Contents

1. Introduction

This document provides basic information on using the Nimbus-4 Backscatter Ultraviolet Spectrometer (BUV) Level-1 Radiance and Housekeeping Data in Telemetry Units, or Primary Data Base (PDB) product.

1.1 Data Product Description

This product contains the BUV data in its first reduced form (telemetry units), with the conversion of the analog signals converted to digital form, deletion of data that did not meet certain quality checks, ordering of data into a logical sequence, and adding of satellite location information.. Each file typically contains one orbit of data. The PDB data are available from April 10, 1970 (day of year 100) through May 6, 1977 (day of year 126).

This product was previously available from the NASA National Space Science Data Center (NSSDC) under the name Backscatter Ultraviolet Spectrometer (BUV) Housekeeping and Radiance Data Reduced but still in Telemetry Units, also known as the Primary Data Base or PDB product, with the identifier ESAC-00024 (old id 70-025A-05E).

1.1.1 The Backscatter Ultraviolet Spectrometer

The Backscatter Ultraviolet Spectrometer (BUV) was designed to monitor the vertical distribution and total column amount of atmospheric ozone on a global scale by measuring the intensity of ultraviolet radiation backscattered by the atmosphere in the 250 to 340 nm spectral region. The primary instrumentation consisted of a double monochromator containing all reflective optics and a photomultiplier detector. The double monochromator was composed of two Ebert-Fastie-type monochromators in tandem. Each monochromator had a 52 x 52-mm grating with 2400 lines per mm. Light from a 0.05-sr solid angle (subtending approximately a 222 km^2 area on the earth's surface from a satellite height of approximately 1100 km) entered the nadir-pointing instrument through a depolarizing filter. A motor-driven cam step rotated the gratings to monitor the intensity of 12 ozone absorption wavelengths. The detector was a photomultiplier tube. For background readings, a filter photometer measured the reflected UV radiation in an ozone-free absorption band at 380 nm.

The Nimbus-4 BUV mission was succeeded by the SBUV instrument flown later on the Nimbus-7 satellite, and subsequently the SBUV/2 instruments on a series of NOAA Polar orbiting Operational Environmental satellites. The BUV experiment was successful and returned data from April 10, 1970 through May 6, 1977. The instrument operated mostly continuously until July 1972 when the Nimbus-4 spacecraft solar array partially failed. After this time data collection had to be curtailed, particularly in the later years.

The original principal investigator for the BUV experiment was Dr.Donald F. Heath.

1.1.2 Nimbus-4 Overview

The Nimbus-4 satellite was successfully launched on April 8, 1970. The spacecraft included nine experiments: (1) an Image Dissector Camera System (IDCS) for providing daytime cloud cover pictures, both in real-time and recorded modes (2) a Temperature-Humidity Infrared Radiometer (THIR) for measuring daytime and nighttime surface and cloudtop temperatures, as well as the water vapor content of the upper atmosphere, (3) an Backscatter Ultraviolet Spectrometer (BUV) for measuring the emission spectra of the earth/atmosphere system, (4) a Satellite Infrared Spectrometer (SIRS) for determining the vertical profiles of temperature and water vapor in the atmosphere, (5) a Monitor of Ultraviolet Solar Energy (MUSE) for detecting solar UV radiation, (6) a Backscatter Ultraviolet (BUV) detector for monitoring the vertical distribution and total amount of atmospheric ozone on a global scale, (7) a Filter Wedge Spectrometer (FWS) for accurate measurement of IR radiance as a function of wavelength from the earth/atmosphere system, (8) a Selective Chopper Radiometer (SCR) for determining the temperatures of six successive 10-km layers in the atmosphere from absorption measurements in the 15-micrometer CO_2 band, and (9) an Interrogation, Recording, and Location System (IRLS) for locating, interrogating, recording, and retransmitting meteorological and geophysical data from remote collection stations.

The orbit of the satellite can be characterized by the following:

- circular orbit at 1100 km
- inclination of 80 degrees
- period of an orbit is about 107 minutes
- orbits cross the equator at 26 degrees of longitude separation
- sun-synchronous

1.2 Algorithm Background

The Nimbus-4 BUV data were generated from the spacecraft telemetry, attitude and orbital data. The data were originally processed on IBM 360 computers using a 32-bit architecture. The intensity of UV radiation at 12 wavelengths between 250 and 340 nm are measured by the monochromator and photomultiplier detector every 32 seconds, along with geolocation, ancillary and housekeeping data. This product is the first step in the ozone processing system. Further information on the BUV data processing can be found in the Nimbus-4 Users' Guide Section 7.

1.3 Data Disclaimer

The data should be used with care and one should first read the User’s Guide to the Nimbus-4 Backscatter Ultraviolet Experiment Data Sets, section 3 describing the Primary Data Base, as well as the Nimbus-4 User’s Guide, section 7 describing the BUV experiment. Users should cite this data product in their research.

2. Data Organization

The Nimbus-4 Backscatter Ultraviolet Spectrometer Level-1 Radiance Data (PDB) spans the time period from April 10, 1970 to May 6, 1977. Each file typically contains one orbit of data.

2.1 File Naming Convention

The data product files are named according to the following convention:

<Platform>-<Instrument>_<Level>-<Type>_<Date>_<Orbit>_<Tape>.<Suffix>

where:
- Instrument = name of the instrument (always BUV)
- Platform = name of the platform or satellite (always Nimbus4)
- Level = processing level of data (always L1)
- Type = the data type indentifier (always PDB)
- Date = Data start date and time in UTC in format <YYYY>m<MMDD>t<hhmm> where
    1. YYYY = 4 digit year (1970 or 1971)
    2. MM = 2 digit month (01-12)
    3. DD = 2 digit day of month (01-31)
    4. hh = 2 digit hour of day (01-23)
    5. mm = 2 digit minute of hour (01-59)
- Orbit = orbit number when the data were collected (preceded by the letter ‘o’)
- Tape = tape number (DR primary tape, DS backup tape plus integer number)
- Suffix = the file format (always dat, indicating binary data)

`File name example: Nimbus4-BUV_L1-PDB_1970m0430t090921_o00296_DS6136.TAP`

2.2 File Format and Structure

The data are stored as they were originally written in IBM binary (big-endian) record oriented structured files. The files were written on the original 1600 bpi 9-track tapes using a FORTRAN block format, up to 10 records per block, each record with a size of 1700 bytes (consisting of850 2-byte words). The first record is the header, which is followed by a series of data records containing the radiances measured by the photometer and monochromator at 12 wavelengths. Along with geolocation information. The data records are followed by two different structured trailer records.

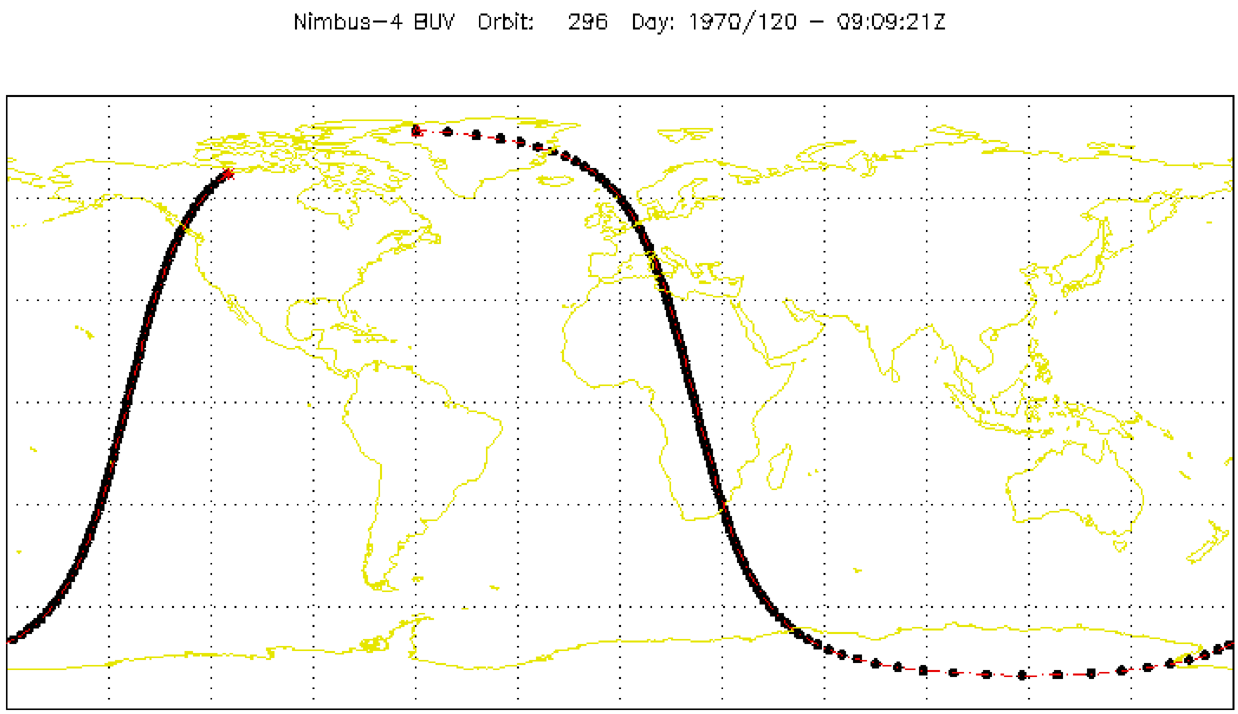
Figure 1: BUV Level-1 PDB File Structure

Records	Object
1	Header Record
2 – N-1	Data Records
N	Trailer Record 2

2.3 Key Science Data Fields

The primary science data fields in this data product are the radiances from the photometer and monochromator stored in their telemetry units.

Figure 2: Typical data coverage for a Nimbus-4 BUV Level-1 PDB orbit file.



3. Data Contents

The granularity of this data collection is one orbit (approximately 107 minutes).

3.1 Data Records

The Level-1 PDB data files contain a single header record, a series of data records, and a single trailer record, each of size 1700 bytes (850 2-byte words). Data were written with up to 10 records per block on the original tape. See the Nimbus-4 BUV Data Products User's Guide – Section 3 for a description of the Primary Data Base (PDB) data product file format. A sample FORTRAN program to read the data files can be found in the Appendix.

Table 3-1-1: Header Record

Word	Field Name	Units	Type	Comments
1	Logical sequence number	-	I*2	always 1
2	Dummy	-	I*2	set to 0
3-6	Unique number of Input Tape	-	R*8	EBCDIC text
7-14	Day and date of Job Run.	-	2R*8	EBCDIC text
15-18	Job I.D.	-	R*8	EBCDIC text
19-20	Day at start of file	-	R*4	(1-366)
21-22	Time at start of file	s	R*4	(0-86400)
23-24	Latitude at the beginning of the file	degrees	R*4	(-90 to +90)
25-26	Longitude at the beginning of the file	degrees	R*4	westward (0-360)
27-28	Week number of start of file		R*4	
29-32	Program name	-	R*8	EBCDIC text
33-36	Version date	-	R*8	EBCDIC text
37-40	Version number	-	R*8	EBCDIC text
41-42	Orbit number		R*4	
43-46	Date of Job Run	-	R*8	EBCDIC text
47-850	Spares	-	402R*4	= -77.

Table 3-1-2: Data Record

Word	Field Name	Units	Type	Comments
1	Logical sequence number	-	I*2	2 or greater
2	Orbit number	-	I*2	set to 0
3	Flag for missing major frame			0=both; 1=1st; 2=2nd
4	Day of beginning of scan	-	I*2	(1-366)
5-6	Start time of 1st major frame	s	I*4	(0-86400)
7-8	Start time of 2nd major frame	s	I*4	(0-86400)
9	Spare	-	I*2	set to 0
10	Day of end of scan	-	I*2	(1-366)
11-12	Time at end of scan	s	I*4	(0-86400)
13-14	Altitude at beginning of scan	km	R*4	
15-16	Latitude at beginning of scan	degrees	R*4	-90 to +90
17-18	Longitude at beginning of scan	degrees	R*4	westward (0-360)
19-20	Solar zenith angle at beginning of scan	degrees	R*4	
21-22	Azimuth angle at beginning of scan	degrees	R*4	
23-24	Latitude at end of scan	degrees	R*4	-90 to +90
25-26	Longitude at end of scan	degrees	R*4	westward (0-360)
27-28	Solar zenith angle at end of scan	degrees	R*4	
29-30	Azimuth angle at end of scan	degrees	R*4	
31	Day/night code for start of 1st major frame		I*2	0=day; 1=twilight; 2=night
32	Day/night code for start of 2nd major frame		I*2	0=day; 1=twilight; 2=night
33-112	FCN 16200: BUV data for 1st major frame		I*2	See user guide
113-192	FCN 16200: BUV data for 2nd major frame		I*2	See user guide
193	FCN 16012: BUV 10 kHz for 1st major frame		I*2	See user guide
194	FCN 16013: BUV Mode LCH/NDR for 1st major frame		I*2	See user guide

195	FCN 16021: BUV Calib. – INH/ENA for 1st major frame		I*2	See user guide
196	FCN 16022: BUV DPLY DIFF for 1st major frame		I*2	See user guide
197	FCN 16023: BUV STR DIFF for 1st major frame		I*2	See user guide
198	FCN 16024: BUV DIF DPLYD for 1st major frame		I*2	See user guide
199	FCN 16025: BUV DIF STRD for 1st major frame		I*2	See user guide
200	FCN 16030: BUV PWAY Cal. for 1st major frame		I*2	See user guide
201	FCN 16031: Electrical Cal. for 1st major frame		I*2	See user guide
202	FCN 16032: Photo. Cal. for 1st major frame		I*2	See user guide
203	FCN 16033: WC Lamp for 1st major frame		I*2	See user guide
204	FCN 16034: MSH Data for 1st major frame		I*2	See user guide
205	FCN 16035: MSH PCAL. for 1st major frame		I*2	See user guide
206	FCN 16036: MSH DCUR for 1st major frame		I*2	See user guide
207	FCN 16037: PSH Data for 1st major frame		I*2	See user guide
208	FCN 16038: PSH PCAL. for 1st major frame		I*2	See user guide
209	FCN 16039: PSH DCUR for 1st major frame		I*2	See user guide
210-226	Same as words 193-209 but for 2nd major frame		17 I*2	See user guide
227	FCN 16101: +4 VDC for 1st major frame		I*2	See user guide
228	FCN 16102: Thermistor Bias (-6.375V) 1st major frame		I*2	See user guide
229	FCN 16103: Photometer High Voltage 1st major frame		I*2	See user guide
230	FCN 16104: Monochromator High Voltage 1st maj. fr.		I*2	See user guide
231	FCN 16105: Housing Absolute Temperature 1st maj. fr		I*2	See user guide
232	FCN 16106: Photomultiplier Absolute Temp. 1st maj fr		I*2	See user guide
233	FCN 16107: Sensor Mod Elect Temperature 1st maj. fr		I*2	See user guide
234	FCN 16108: Mtr Cur Limiter Temperature 1st maj. fr.		I*2	See user guide
235	FCN 16109: Static Inverter 1 Temperature 1st maj. fr.		I*2	See user guide
236	FCN 16110: Static Inverter 2 Temperature 1st maj. fr.		I*2	See user guide

237	FCN 16111: Arm Gradient for 1st major frame		I*2	See user guide
238	FCN 16112: Housing Gradient for 1st major frame		I*2	See user guide
239-250	Same as words 226-238 but for 2nd major frame		I*2	See user guide
251-266	FCN 14001, MUSE Data for 1st major frame		16 I*2	See user guide
267-282	FCN 14002, MUSE Data for 1st major frame		16 I*2	See user guide
283	FCN 14003: MUSE – 3 volts for 1st major frame		I*2	See user guide
284	FCN 14004: MUSE – 6 volts for 1st major frame		I*2	See user guide
285	FCN 14005: MUSE Aspect Sensor ATA for 1st maj. fr.		I*2	See user guide
286	FCN 14006: MUSE Aspect Sensor EATA for 1st maj. fr.		I*2	See user guide
287	FCN 14007 MUSE Cathode Temps for 1st maj. frame		I*2	See user guide
288	FCN 14008 MUSE Feedback Res. T. for 1st maj. frame		I*2	See user guide
289	FCN 14009 MUSE Elec. Temp. for first major frame		I*2	See user guide
290	FCN 14011 MUSE Pitch Eye 1 in 2° Bit for 1st maj. fr.		I*2	See user guide
291	FCN 14012 MUSE Pitch Eye 2 in 2° Bit for 1st maj. fr.		I*2	See user guide
292	FCN 14013 MUSE Pitch Eye 3 in 2° Bit for 1st maj. fr.		I*2	See user guide
293-295	FCN 14014 MUSE Pitch Eye 4 in 2° Bit for 1st maj. fr.		3 I*2	See user guide
296-298	FCN 14015 MUSE Pitch Eye 5 in 2° Bit for 1st maj. fr.		3 I*2	See user guide
299-301	FCN 14016 MUSE Pitch Eye 6 in 2° Bit for 1st maj. fr.		3 I*2	See user guide
302	FCN 14017 MUSE Pitch Eye 7 in 2° Bit for 1st maj. fr.		I*2	See user guide
303-305	FCN 14021 MUSE Yaw Eye 1 in 2° Bit for 1st maj. fr.		3 I*2	See user guide
306-308	FCN 14022 MUSE Yaw Eye 2 in 2° Bit for 1st maj. fr.		3 I*2	See user guide
309-311	FCN 14023 MUSE Yaw Eye 3 in 2° Bit for 1st maj. fr.		3 I*2	See user guide
312-314	FCN 14024 MUSE Yaw Eye 4 in 2° Bit for 1st maj. fr.		3 I*2	See user guide
315-317	FCN 14025 MUSE Yaw Eye 5 in 2° Bit for 1st maj. fr.		3 I*2	See user guide
318-320	FCN 14026 MUSE Yaw Eye 6 in 2° Bit for 1st maj. fr.		3 I*2	See user guide
321-323	FCN 14027 MUSE Yaw Eye 7 in 2° Bit for 1st maj. fr.		3 I*2	See user guide

324-339	FCN 14030 Reference Ind. In Bit 3 for 1st maj. fr.		16 I*2	See user guide
340-355	FCN 14031 Range Bit 1 in Bit 1 for 1st major frame		16 I*2	See user guide
356-371	FCN 14032 Range Bit 2 in Bit 5 for 1st major frame		16 I*2	See user guide
372-387	FCN 14033 Range Bit 3 in Bit 10 for 1st major frame		16 I*2	See user guide
388-390	FCN 14034 MUSE Power On/Off in Bit 1 for 1st maj. fr.		3 I*2	See user guide
391-393	FCN 14035 MUSE Man/Auto in Bit 10 for 1st maj. fr.		3 I*2	See user guide
394-536	Same as words 251-393 but for 2nd major frame		143 I*2	See user guide
537	FCN 1101 Coarse Pitch Error for 1st major frame		I*2	See user guide
538-553	FCN 1102 Fine Pitch for 1st major frame		16 I*2	See user guide
554-569	FCN 1103 Pitch Tach. Amp. for 1st major frame		16 I*2	See user guide
570	FCN 1201 Coarse Roll for 1st major frame		I*2	See user guide
571-586	FCN 1202 Fine Roll for 1st major frame		16 I*2	See user guide
587-602	FCN 1205 Roll Fwd. Flywheel Speed for 1st maj. frame		16 I*2	See user guide
603-618	FCN 1206 Roll Rear Flywheel Speed for 1st maj. frame		16 I*2	See user guide
619-634	FCN 1303 Yaw Tach. Amp. for 1st major frame		16 I*2	See user guide
635	FCN 1322 Yaw Sun Sensor Amp. for 1st major frame		I*2	See user guide
636-651	FCN 1351 Rup Ind. Rate (Hi. Res.) for 1st major frame		16 I*2	See user guide
652	FCN 1411 Left SAD SSDA		I*2	See user guide
653-655	FCN 1413 Left SAD Phase-Switch in Bit 7 1st maj. fr.		3 I*2	See user guide
656-671	FCN 1417 Left SAD Tach. for 1st major frame		16 I*2	See user guide
672-687	FCN 1431 Left Cosine Pot for 1st major frame		16 I*2	See user guide
688	FCN 2005 Solar Array I for 1st major frame			See user guide
689-840	Same as words 537-688 but for 2nd major frame		152 I*2	See user guide
841	Orbit Number		I*2	
842-850	Spares		9 I*2	= -77.

Table 3-1-3: Trailer Record

Word	Field Name	Units	Type	Comments
1	Logical sequence number	-	I*2	negative
2	Dummy word	-	I*2	= 0
3-4	Day of last scan	-	R*4	(1-366)
5-6	Time at end of orbit	s	R*4	(0-86400)
7-8	Latitude at end of orbit	degrees	R*4	-90 to +90
9-10	Longitude at end of orbit	degrees	R*4	westward (0-360)
11-12	Number of SSDT records (frames) read		R*4	
13-14	Number of STP records (scans) written		R*4	
15-18	Unique number of the input tape		R*8	EBCDIC text
19-20	Data Record Read Error		R*4	
21-22	Wrong Record Length Encountered		R*4	
23-24	Time Not Available		R*4	
25-26	Frame Synch. Error		R*4	
27-28	BUV Power Off		R*4	
29-30	Bad Time on Record		R*4	
31-32	Data Cycle Neither First nor Second		R*4	
33-34	Backward Time Step		R*4	
35-850	Spares		816 R*4	= -77.

3.2 Metadata

The metadata are contained in a separate XML formatted file having the same name as the data file with .xml appended to it.

Table 3-2: Metadata attributes associated with the data file.

Name	Description
LongName	Long name of the data product.
ShortName	Short name of the data product.
VersionID	Product or collection version.
GranuleID	Granule identifier, i.e. the name of the file.
Format	File format of the data file.
CheckSumType	Type of checksum used.
CheckSumValue	The value of the calculated checksum.
SizeBytesDataGranule	Size of the file or granule in bytes.
InsertDateTime	Date and time when the granule was inserted into the archive. The format for date is YYYY-MM-DD and time is hh-mm-ss.
ProductionDateTime	Date and time the file was produced in format YYYY-MM-DDThh:mm:ss.ssssssZ
RangeBeginningDate	Begin date when the data was collected in YYYY-MM-DD format.
RangeBeginningTime	Begin time of the date when the data was collected in hh-mm-ss format.
RangeEndingDate	End date when the data was collected in YYYY-MM-DD format.
RangeEndingTime	End time of the date when the data was collected in hh-mm-ss format.
GPolygon: PointLatitude	Latitudes of the polygon (rectangle) points that represent the satellite coverage. Each point is identified by its latitude and longitude pair.
GPolygon: PointLongitude	Longitudes of the polygon (rectangle) points that represent the satellite coverage. Each point is identified by its latitude and longitude pair.
PlatformShortName	Short name or acronym of the platform or satellite
InstrumentShortName	Short name or acronym of the instrument
SensorShortName	Short name or acronym of the sensor
OrbitNumber	Number of the orbit in the file
Elapsed_Min_Time	Duration in minutes of data collected during an orbit

4. Reading the Data

The data are written in a binary record-oriented format. Using the record format specification in the section above, users can write software to read the data files. Please note that the data were originally written using a big-endian format, therefore users on little-endian machines will need to swap bytes for the words. Also, the floating point data were written using IBM 360 machines, and must be converted if reading on a machine that understands IEEE floats (integers are not affected).

A sample FORTRAN program is included in the Appendix section which will read the data files. Additionally three FORTRAN functions are included to perform byte swapping, conversion from IBM float to IEEE float, and translation of EBCDIC to ASCII text.

5. Data Services

5.1 GES DISC Search

The GES DISC provides a keyword, spatial, temporal and advanced (event) searches through its unified search and download interface:

https://disc.gsfc.nasa.gov/

5.2 Documentation

The data product landing pages provide information about this data product, as well as links to download the data files and relevant documentation:

https://disc.gsfc.nasa.gov/datacollection/BUVN4L1PDB_001.html

5.3 Direct Download

This data product is available for users to download directly using HTTPS:

https://acdisc.gesdisc.eosdis.nasa.gov/data/Nimbus4_BUV_Level1/BUVN4L1PDB.001/

6. More Information

6.1 Contact Information

Name: GES DISC Help Desk
URL: https://disc.gsfc.nasa.gov/
E-mail: gsfc-help-disc@lists.nasa.gov
Phone: 301-614-5224
Fax: 301-614-5228
Address: Goddard Earth Sciences Data and Information Services Center
Attn: Help Desk
Code 610.2
NASA Goddard Space Flight Center
Greenbelt, MD 20771, USA

6.2 References

D. F. Heath, and A. J. Kruger, "The Nimbus-4 User's Guide - Section 7: The Backscatter Ultraviolet Spectrometer (BUV) Experiment", NASA Goddard Space Flight Center, March 1970, Pages 149-172

Ozone Processing Team, "User's Guide to the Nimbus-4 Backscatter Ultraviolet Experiment Data Sets – Section 3: Primary Data Base", NASA Goddard Space Flight Center, Technical Memorandum 78069, Jan. 1978, Pages 8-23

7. Appendices

Acknowledgements

The Nimbus data recovery task at the GES DISC is funded by NASA's Earth Science Data and Information System program.

Acronyms

EOS: Earth Observing System

ESDIS: Earth Science and Data Information System

GES DISC: Goddard Earth Sciences Data and Information Services Center

GSFC: Goddard Space Flight Center

BUV: Backscatter Ultraviolet Spectrometer

PDB: Primary Data Base

L1: Level-1 Data

NASA: National Aeronautics and Space Administration

QA: Quality Assessment

U-Values: Radiance in resolution units of the digitizer

UT: Universal Time

FORTRAN Code

```
C----------------------------------------------------------------------
C ^NAME: READ_BUV_PDB
C
C ^DESCRIPTION:
C     This program opens and reads a Nimbus-4 BUV level-1 PDB data file
C     and prints the contents of the file to the screen. Data files
C     consist of a single header, a number of data, and a final trailer
C     record, each of size 400 bytes (100 4-byte words). See the User's
C     Guide for the Nimbus-4 Backscatter Ultraviolet Experiment - Section 4
C     Radiance Data (or U-Tape) which gives the product file specification.
C
C ^MAJOR VARIABLES:
C     FNAME  - name of input file
C
C ^NOTES:
C     Compile: gfortran -xxx -o READ_BUV_PDB.EXE READ_BUV_PDB.FOR
C
C ^ORGANIZATION: NASA/GSFC, Code 610.2
C
C ^AUTHOR:  James Johnson
C
C ^ADDRESS: james.johnson@nasa.gov
C
C ^CREATED: October 13, 2015
C----------------------------------------------------------------------

      CHARACTER          FNAME*1024    ! Name of input file
      CHARACTER          BLOCK(10000)  ! Blocks are maximum 25 100 4-byte words
      CHARACTER          BUFF(4)       ! Buffer to hold 4-byte word
      INTEGER*4          WORD          ! 4-byte word
      INTEGER*4          BLKSIZ        ! Block size header
      EQUIVALENCE        (BUFF, WORD)

C     Get the name of the input data file to read
      PRINT *, 'Enter the name of the input file:'
      READ (5,'(A)') FNAME

C     Open the specified input file
      OPEN (UNIT=1, FILE=FNAME, STATUS='OLD', ACCESS='DIRECT',
     &      FORM='UNFORMATTED', RECL=1, ERR=99, IOSTAT=IOS)

C     Initialize N (block number) and IOFF (byte offset)
      N=1
      IOFF=0

C     Loop through the file reading all blocks of data
   10 DO

C       Read first 4-byte word or block size
        DO I=1,4
          READ (1, REC=I+IOFF, ERR=91, IOSTAT=IOS) BUFF(I)
        END DO
        IOFF = I+IOFF-1
        BLKSIZ = WORD
```

```
        IF (BLKSIZ .GT. 0) THEN

C       Next read the block of data
          DO I=1,BLKSIZ
            READ (1, REC=I+IOFF, ERR=92, IOSTAT=IOS) BLOCK(I)
          END DO
          IOFF=I+IOFF-1
          CALL RECBLK(BLOCK, BLKSIZ, N)

        END IF

C       Finally read last 4-byte word (should match first block size)
        DO I=1,4
          READ (1, REC=I+IOFF, ERR=93, IOSTAT=IOS) BUFF(I)
        END DO
        IOFF=I+IOFF-1

        IF (BLKSIZ .NE. WORD) THEN
          PRINT '("BLKSIZ: ",I10," != ",I10)', BLKSIZ, WORD
        END IF

C       END OF FILE MARKER
        IF (BLKSIZ .EQ. 0 .AND. WORD .EQ. 0) GOTO 90

        N=N+1

      END DO

C     Close the input file
   90 CLOSE(1)
      GOTO 100
   91 PRINT '("ERROR: READ FIRST WORD, IOSTAT: ",I5)', IOS
      GOTO 100
   92 PRINT '("ERROR: READ BLOCK ",I4,", IOSTAT: ",I5)', N, IOS
      GOTO 100
   93 PRINT '("ERROR: READ LAST WORD, IOSTAT = ",I5)', IOS
      GOTO 100
   99 PRINT '("ERROR: OPENING FILE, IOSTAT: ",I6)', IOS

  100 STOP
      END
```

```
C-----------------------------------------------------------------------
C ^SUBROUTINE: RECBLK
C
C     This subroutine will extract records from the block of data
C-----------------------------------------------------------------------

      SUBROUTINE RECBLK(BLOCK, IBLKSZ, N)

      INTEGER*4          BLOCK(25*100) ! Blocks are maximum 25 100 4-byte words
      INTEGER*4          IREC(100)     ! Record is 100 4-byte words
      LOGICAL            LTR2          ! Second Trailer Record flag

      DATA IRECSZ /400/

      NRECS = IBLKSZ/IRECSZ

      DO I=1,NRECS

        IREC = BLOCK((I-1)*100+1:I*100)

C       Check the logical sequence number in word 1
        IF (.NOT. LTR2) RNUM = R4IBM(I4SWAP(IREC(1)))
        IF (RNUM .EQ. 1) THEN
          CALL PRHREC(IREC, N)
        ELSE IF (RNUM .GT. 1) THEN
          CALL PRDREC(IREC, N)
        ELSE IF (RNUM .LT. 0) THEN
          IF (.NOT. LTR2) THEN
            CALL PRLRC1(IREC, N)
            LTR2 = .TRUE.
          ELSE
            CALL PRLRC2(IREC, N)
          END IF
        ELSE
          PRINT '("UNKNOWN RECORD TYPE: ", I3,X,I11)', ITYPE, N
        END IF

      END DO

      RETURN
      END
```

```
C-----------------------------------------------------------------------
C ^SUBROUTINE: PRHREC
C
C     This subroutine will print the header record to the screen
C-----------------------------------------------------------------------

      SUBROUTINE PRHREC(IREC, N)

      INTEGER*4          IREC(100)     ! Data record
      INTEGER*4          IEBC(4)       ! Temporary array for EBC text
      CHARACTER          EBCTMP(16)    ! Temporary EBCDIC character
      CHARACTER          ASCVAL*16     ! Array for ASCII text
      CHARACTER          EBC2ASC       ! Function to convert EBC to ASCII
      EQUIVALENCE        (IEBC,EBCTMP)

C Words 3-4, 5-8, 9-10, 17-18, 19-20, and 25-26 contain EBCDIC text data.
C All other words are in IBM floating-point format (REAL*4). Any word in the
C record may contain -77., indicating fill data.

      PRINT '("HEADER RECORD")'
      DO 100 I=1,100

        IF ( (I .GE.  3 .AND. I .LE.  4) .OR.
     &       (I .GE.  5 .AND. I .LE.  8) .OR.
     &       (I .GE.  9 .AND. I .LE. 10) .OR.
     &       (I .GE. 17 .AND. I .LE. 18) .OR.
     &       (I .GE. 19 .AND. I .LE. 20) .OR.
     &       (I .GE. 21 .AND. I .LE. 22) .OR.
     &       (I .GE. 25 .AND. I .LE. 26) ) THEN
          IF (I .NE.  3 .AND. I .NE.  5 .AND. I .NE.  9 .AND.
     &        I .NE. 17 .AND. I .NE. 19 .AND.
     &        I .NE. 21 .AND. I .NE. 25) GOTO 100
          IF (I .EQ. 5) THEN
            IEBC = IREC(I:I+3)
            DO 10 J=1,16
              ASCVAL(J:J) = EBC2ASC(ICHAR(EBCTMP(J)))
   10       CONTINUE
            PRINT '("WORD ",I3," = ",A16)', I,ASCVAL                ! Two Real*8
          ELSE
            IEBC = IREC(I:I+1)
            DO 20 J=1,8
              ASCVAL(J:J) = EBC2ASC(ICHAR(EBCTMP(J)))
   20       CONTINUE
            PRINT '("WORD ",I3," = ",A8)', I,ASCVAL                 ! One Real*8
          END IF
        ELSE
          PRINT '("WORD ",I3," = ",G12.6)', I,R4IBM(I4SWAP(IREC(I)))
        END IF

  100 CONTINUE
      PRINT '("")'

      RETURN
      END
```

```
C----------------------------------------------------------------------
C ^SUBROUTINE: PRDREC
C
C     This subroutine will print the data records to the screen
C----------------------------------------------------------------------

      SUBROUTINE PRDREC(IREC, N)

      INTEGER*4          IREC(100)     ! Data record

C All words are in IBM floating-point format (REAL*4). Any word in the
C record may contain -77., indicating fill data.

      PRINT '("DATA RECORD")'
      DO 100 I=1,100

        PRINT '("WORD ",I3," = ",G12.6)', I,R4IBM(I4SWAP(IREC(I)))

  100 CONTINUE
      PRINT '("")'

      RETURN
      END

C----------------------------------------------------------------------
C ^SUBROUTINE: PRLRC1
C
C     This subroutine will print the first trailer record to the screen
C----------------------------------------------------------------------

      SUBROUTINE PRLRC1(IREC, N)

      INTEGER*4          IREC(100)     ! Data record
      INTEGER*4          IEBC(4)       ! Temporary array for EBC text
      CHARACTER          EBCTMP(16)    ! Temporary EBCDIC character
      CHARACTER          ASCVAL*16     ! Array for ASCII text
      CHARACTER          EBC2ASC       ! Function to convert EBC to ASCII
      EQUIVALENCE        (IEBC,EBCTMP)

C Words 9-10 contain EBCDIC text data. All other words are in IBM
C floating-point format (REAL*4). Any word in the record may contain -77.,
C indicating fill data.

      PRINT '("1ST TRAILER RECORD")'
      DO 100 I=1,100

        IF ( (I .GE.  9 .AND. I .LE. 10) ) THEN
          IF (I .NE.  9) GOTO 100
          IEBC = IREC(I:I+1)
          DO 10 J=1,8
            ASCVAL(J:J) = EBC2ASC(ICHAR(EBCTMP(J)))
   10     CONTINUE
          PRINT '("WORD ",I3," = ",A8)', I,ASCVAL                 ! One Real*8
        ELSE
          PRINT '("WORD ",I3," = ",G12.6)', I,R4IBM(I4SWAP(IREC(I)))
        END IF

  100 CONTINUE
      PRINT '("")'

      RETURN
      END
```

```
C-----------------------------------------------------------------------
C ^SUBROUTINE: PRLRC2
C
C     This subroutine will print the second trailer record to the screen
C-----------------------------------------------------------------------

      SUBROUTINE PRLRC2(IREC, N)

      INTEGER*4          IREC(100)     ! Data record

C All words are in IBM floating-point format (REAL*4). Any word in the
C record may contain -77., indicating fill data.

      PRINT '("2ND TRAILER RECORD")'
      DO 100 I=1,100

        PRINT '("WORD ",I3," = ",G12.6)', I,R4IBM(I4SWAP(IREC(I)))

  100 CONTINUE
      PRINT '("")'

      RETURN
      END

C-----------------------------------------------------------------------
C ^SUBROUTINE: IBLKID
C
C     This subroutine will decode the block identifier (word 1)
C-----------------------------------------------------------------------

      SUBROUTINE IBLKID(IWORD, IBLKNO, ILASTB, ILASTF, IRECID)
                                                     !  bits
          IBLKNO = ISHFT(IWORD,-20)                  !  1-12 Block Number
          ILASTB = IAND(IWORD,'1000000000000000'B)   !    17 Last Block of File
          ILASTF = IAND(IWORD,'0100000000000000'B)   !    18 Last File of Tape
          IRECID = ISHFT(ISHFT(IWORD,18),-26)        ! 19-24 Record Identifier

      RETURN
      END

C-----------------------------------------------------------------------
C ^FUNCTION: I4SWAP
C
C     This subroutine will swap the bytes of a data element
C-----------------------------------------------------------------------

      FUNCTION I4SWAP(INPUT)

      INTEGER*4          IWORD          ! Input 4-byte word
      INTEGER*4          IDROW          ! Byte-swapped 4-byte word
      CHARACTER          DATBUF(4)      ! Input data buffer
      CHARACTER          SWPBUF(4)      ! Output swapped buffer
      EQUIVALENCE        (IWORD, DATBUF)
      EQUIVALENCE        (IDROW, SWPBUF)

      IWORD = INPUT
      DO 10 K=1,4
        SWPBUF(K) = DATBUF(4-K+1)
   10 CONTINUE
      I4SWAP = IDROW

      RETURN
      END
```

```
C-----------------------------------------------------------------------
C ^FUNCTION: R4IBM
C
C     This function will convert an input word to an IBM float
C-----------------------------------------------------------------------

      FUNCTION R4IBM(IWORD)

      INTEGER*4      IDROW              ! reverse the bits of input word
      REAL*8         A /16.0/           ! base number
      INTEGER*4      B /64/             ! exponent offset
      REAL*8         C /0.0/            ! fraction offset
      INTEGER*1      S                  ! sign flag
      INTEGER*2      E                  ! binary exponent
      REAL*8         F                  ! binary fraction
      REAL*8         M                  ! mantissa
      REAL*8         V                  ! float value
      INTEGER*4      I                  ! counter

      S = ISHFT(IWORD, -31)

      E = 0
      DO 10 I=0,6
        E = E + IAND(ISHFT(IWORD,-24),ISHFT(1,I))
   10 END DO

      IDROW = 0
      DO 11 I=0,31
        IF (IAND(IWORD,ISHFT(1,I)) .NE. 0) THEN
          IDROW = IOR(IDROW,(ISHFT(1,31-I)))
        END IF
   11 END DO

      F = 0.0
      DO 12 I=0,31
        IF (ISHFT(IAND(ISHFT(IDROW,-8),ISHFT(1,I)),1) .NE. 0) THEN
          F = F + 1 / FLOAT(ISHFT(IAND(ISHFT(IDROW,-8),ISHFT(1,I)),1))
        END IF
   12 END DO

      M = C + F                         ! calculate the mantissa
      V = (-1)**S * M * A**(E - B)      ! calculate the float value
      IF (ABS(V) .LT. 2.0**(-149)) THEN
        V = (-1)**S * 0.0               ! avoid underflow
      END IF

      R4IBM = V
      RETURN
      END
```

```
C-----------------------------------------------------------------------
C ^FUNCTION: EBC2ASC
C
C     This function will convert an EBCDIC character to ASCII
C-----------------------------------------------------------------------

      CHARACTER FUNCTION EBC2ASC(N)

      INTEGER            EBCTAB(256)

C  EBCDIC to ASCII table
      DATA EBCTAB/
     & 000,001,002,003,156,009,134,127,151,141,142,011,012,013,014,015, ! 0
     & 016,017,018,019,157,133,008,135,024,025,146,143,028,029,030,031, ! 1
     & 128,129,130,131,132,010,023,027,136,137,138,139,140,005,006,007, ! 2
     & 144,145,022,147,148,149,150,004,152,153,154,155,020,021,158,026, ! 3
     & 032,160,161,162,163,164,165,166,167,168,213,046,060,040,043,124, ! 4
     & 038,169,170,171,172,173,174,175,176,177,033,036,042,041,059,094, ! 5
     & 045,047,178,179,180,181,182,183,184,185,229,044,037,095,062,063, ! 6
     & 186,187,188,189,190,191,192,193,194,096,058,035,064,039,061,034, ! 7
     & 195,097,098,099,100,101,102,103,104,105,196,197,198,199,200,201, ! 8
     & 202,106,107,108,109,110,111,112,113,114,203,204,205,206,207,208, ! 9
     & 209,126,115,116,117,118,119,120,121,122,210,211,212,091,214,215, ! A
     & 216,217,218,219,220,221,222,223,224,225,226,227,228,093,230,231, ! B
     & 123,065,066,067,068,069,070,071,072,073,232,233,234,235,236,237, ! C
     & 125,074,075,076,077,078,079,080,081,082,238,239,240,241,242,243, ! D
     & 092,159,083,084,085,086,087,088,089,090,244,245,246,247,248,249, ! E
     & 048,049,050,051,052,053,054,055,056,057,250,251,252,253,254,255/ ! F
C       0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F

      EBC2ASC = CHAR(EBCTAB(N+1))

      RETURN
      END
```