VDM+: The Enhanced Vortex Message Dataset: Structure, Intensity, and Environmental Parameters from Atlantic Tropical Cyclones

Jonathan L. Vigh

NCAR Technical Notes
NCAR/TN 517+STR

National Center for
Atmospheric Research
P. O. Box 3000
Boulder, Colorado
80307-3000
www.ucar.edu

NCAR | National Center for
UCAR | Atmospheric Research

National Science Foundation
NCAR IS SPONSORED BY THE NSF

NCAR TECHNICAL NOTES

http://library.ucar.edu/research/publish-technote

The Technical Notes series provides an outlet for a variety of NCAR Manuscripts that contribute in specialized ways to the body of scientific knowledge but that are not yet at a point of a formal journal, monograph or book publication. Reports in this series are issued by the NCAR scientific divisions, serviced by OpenSky and operated through the NCAR Library. Designation symbols for the series include:

EDD – Engineering, Design, or Development Reports
Equipment descriptions, test results, instrumentation, and operating and maintenance manuals.

IA – Instructional Aids
Instruction manuals, bibliographies, film supplements, and other research or instructional aids.

PPR – Program Progress Reports
Field program reports, interim and working reports, survey reports, and plans for experiments.

PROC – Proceedings
Documentation or symposia, colloquia, conferences, workshops, and lectures. (Distribution maybe limited to attendees).

STR – Scientific and Technical Reports
Data compilations, theoretical and numerical investigations, and experimental results.

The National Center for Atmospheric Research (NCAR) is operated by the nonprofit University Corporation for Atmospheric Research (UCAR) under the sponsorship of the National Science Foundation. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

National Center for Atmospheric Research
P. O. Box 3000
Boulder, Colorado 80307-3000

NCAR/TN-517+STR
NCAR Technical Note

2015-07

VDM+: The Enhanced Vortex Message Dataset:

Structure, Intensity, and Environmental Parameters from Atlantic Tropical Cyclones

Jonathan L. Vigh
Research Applications Laboratory/Joint Numerical Testbed Program,
National Center for Atmospheric Research, Boulder, Colorado

NCAR Research Applications Laboratory
Joint Numerical Testbed

NATIONAL CENTER FOR ATMOSPHERIC RESEARCH
P. O. Box 3000
BOULDER, COLORADO 80307-3000
ISSN Print Edition 2153-2397
ISSN Electronic Edition 2153-2400

VDM+: The Enhanced Vortex Data Message Dataset

Structure, Intensity, and Environmental Parameters from Atlantic Tropical Cyclones

Version: 1.001
Release Date: 01 August 2015
Dataset creator: Jonathan Vigh, Ph.D.

Author: Jonathan L. Vigh

Documentation date: 30 Jun 2015

Table of Contents

Introduction

The Vortex Data Message Dataset (VDM+) consists of tropical cyclone data that have been collected, translated, processed, and aggregated from the following four sources:

- an archive of raw Vortex Data Messages (VDMs),
- the Best Track (BT) Data Set,
- the Extended Best Track (EBT) Data Set, and
- the Statistical Hurricane Intensity Prediction Scheme (SHIPS) Development Data Set.

The current version of the dataset includes all tropical cyclones in the North Atlantic basin that occurred during the years 1989-2012.

Scope and Purpose of this Document

This documentation serves four purposes: a) to describe the various tiers of data products provided by the VDM+ dataset, b) to provide updated information on the data sources used in the compilation of the VDM+ dataset as well as provenance of the VDMs, c) to document the structure and indexing of the aggregated data set file, and d) to provide a full manifest of all dataset parameters along with detailed information on how each parameter was translated from the raw data.

Important References

Detailed information on the data sources and processing of the four data sources are provided by:

Vigh, J. L., J. A. Knaff, and W. H. Schubert, 2012: A climatology of hurricane eye formation. *Mon. Wea. Rev.*, **140**, 1405-1426, doi:10.1175/MWR-D-11-00108.1. (article pdf, supplement pdf).

Vigh, J. L., 2010a: Formation of the hurricane eye. Ph.D. dissertation, Dept. of Atmospheric Science, Colorado State University, Fort Collins, Colorado, 80523, 538 pp. [Available in the official graduate school format (best for printing) or a single-spaced format with hyperlinks (best for viewing onscreen)].

Vigh et al (2012) and the accompanying supplement serve as the primary scientific reference for the VDM+ dataset. The supplement provides an introductory overview of the anatomy of VDMs and discusses important caveats regarding the aircraft data sources and their limitations. **It is recommended that any data set user read the first five sections of the supplement first before continuing to the rest of this document.** Vigh (2010) provides an earlier reference detailing how the VDM dataset was applied to examine the structural and intensity changes that occur before, during, and after the formation of the primary hurricane eye.

Limitations of this documentation

The next sections describe the various tiers of the data set and the structure of the associated data files, but the reader should be aware that this document does not provide an exhaustive documentation of all the assumptions and processing techniques used to translate the raw VDM data to the processed structure and intensity parameters. For complete details on how each parameter was parsed and translated, the author recommends the user refer to the source code itself, as this serves as the best description of the minutiae of data processing (to obtain the source code, please contact the author at jvigh@ucar.edu). Nevertheless, this documentation attempts to provide a "good-enough" overview of these details so as to serve as a practicable reference for most dataset users, casual and otherwise.

Data Product Tiers

The VDM+ data set consists of several "tiers" of data products, as follows:

- **Tier 0**: the raw source data in its original format: raw text VDMs, the individual Automated Tropical Cyclone Forecast (ATCF) b-deck files that comprise the BT dataset, the EBT data set, and the SHIPS development data set.
    - NOTE: The ATCF system uses a comma-delimited file format to represent various types of data. Files that contain the forecast aids (e.g., model guidance, etc.) are referred to as *a-decks*. Files that contain information about the TC's analyzed center location, pressure, intensity, and wind radii are referred to as *b-decks*. The following section contains more information about the contents of the b-decks.)
- **Tier 1**: individual data files (one per storm) in the Network Common Data Format (NetCDF) that contain the processed and translated structure and intensity parameters from the VDMs, the full complement of parameters from the BT b-deck files, information for the storm position, intensity, size, wind radii, and distance from land data from the EBT (merged with BT information, as described below), and the model-analyzed environmental and satellite-based parameters from the SHIPS developmental data set. The Tier 1 files also include the raw lines from the VDMs, intermediate parsed parameters, and several additional parameters that have been calculated from combinations of parameters in the source data sets. Altogether, there are 355 data parameters (including coordinate data).
- **Tier 2**: an aggregated NetCDF file that contains all of the Tier 1 data for all available storms during the period. Two aggregated files are provided: one that includes only storms in the Atlantic basin, and a combined file which includes all storms in the Atlantic, Eastern Pacific, and Central Pacific basins. The Tier 2 product constitutes the full dataset and as such, is the product recommended to users.

The current VDM+ data set version (v1.001) provides the Tier 1 and Tier 2 data in fully self-describing NetCDF data files. Identical descriptive long parameter names are used in both data files. The naming convention will be described later.

Tier 0 Data: Description of the Raw Data Sources

NHC's Master Storm Table

The NHC master storm table is utilized to provide a consistent mapping between storm numbers, storm names, and the unique storm identification tags. This is necessary because storm numbers are sometimes relabeled during post-season anaylsis. The storm table is updated in real-time and is available on the NHC web server at:
ftp://ftp.nhc.noaa.gov/atcf/index/storm_list.txt

To aid in the processing of these data to the Tier 1 data files, this ASCII-formatted file has been converted to a separate, self-contained NetCDF file. The VDM+ v1.001 dataset uses the storm table that was downloaded from NHC's website on 01 November 2013.

Best Track Database/HURDAT2

The VDM+ dataset includes the full parameter set of the National Hurricane Center (NHC)'s Hurricane Database (HURDAT2), otherwise known as the "Best Track" database. Since HURDAT2 is actually built up from the underlying ATCF b-deck files (Franklin 2013, personal communication), the VDM+ dataset uses these b-deck files as the definitive source of Best Track information, rather than the compiled HURDAT2 dataset. **It should be noted however that not all parameters in the b-deck files undergo the rigorous post-season subjective analysis by NHC hurricane specialists.**

The b-deck parameters that are best-tracked, and thus included in HURDAT2, include (ATCF b-deck parameter name given in parantheses):

- Storm status (or type; TY)
- Position of the storm center (LAT/LON)
- Maximum sustained surface wind speed (VMAX, otherwise known as 'Intensity')
- Minimum central pressure (MSLP)
- Wind radii in each quadrant for the 34-, 50-, and 64-kt thresholds (beginning in 2004; given by RAD, WINDCODE, and RAD1, RAD2, RAD3, and RAD4)

The b-deck parameters that are ***not*** best-tracked, and hence are *not* included in HURDAT2, include (ATCF b-deck parameter name given in parantheses):

- Radius of maximum winds (MRD)
- Maximum gusts (GUST)
- Eye diameter (EYE)
- Radius of the last closed isobar (RADP)
- Pressure of the last closed isobar (RRP)
- Maximum seas (MAXSEAS)
- Radial extent of seas of the given threshold (given by SEAS, SEASCODE, SEAS1, SEAS2, SEAS3, and SEAS4)

The b-deck files are freely available for download at ftp://ftp.nhc.noaa.gov/atcf/archive/. The HURDAT2 database itself may be downloaded from: http://www.nhc.noaa.gov/data/#hurdat. Documentation of HURDAT2 is available at: http://www.nhc.noaa.gov/data/hurdat/hurdat2-format-atlantic.pdf. The abr-deck file format is described by: http://www.nrlmry.navy.mil/atcf_web/docs/database/new/abrdeck.html. The VDM+ v1.001 dataset uses b-deck files downloaded from NHC's website on 01 November 2013.

Extended Best Track Dataset

Since the Best Track database does not include wind radii values prior to 2004 when NHC began best-tracking these, the Extended Best Track (EBT) dataset provides these from the values that were operationally analyzed in real-time. The EBT dataset also includes many of the non-best-tracked parameters for storms that are currently missing in the b-deck data files, and adds an additional parameter that provides the distance to the nearest land. The EBT dataset extends back to 1988 and is maintained by NOAA's National Environment Satellite, Data, and Information Service (NOAA/NESDIS) Center for Statellite Applications and Research (StAR).

The VDM+ dataset merges the BT and EBT data into one set of combined Best Track data by using the BT parameter values whenever they exist. If the BT parameter value is missing, then the EBT value is used as the merged value. If data are present in both datasets but differ in value, the BT value is now used as the definitive merged value. (Note that the current approach differs to what was previously done in the earlier version of the VDM dataset described in the supplement to Vigh et al 2012.). This merging process is done for the following b-deck parameters: LAT, LON, VMAX, MSLP, MRD, EYE, RADP, RRP, and the 34-kt, 50-kt, and 64-kt wind radii. Additionally, the EBT distance-to-land parameter is also included in the combined best track parameters of the VDM+ dataset.

The EBT dataset is freely available for download from: http://rammb.cira.colostate.edu/research/tropical_cyclones/tc_extended_best_track_dataset/. Additional documentation on the EBT dataset are available from: ftp://rammftp.cira.colostate.edu/demaria/ebtrk/ebtrk_readme.txt and in the following reference:

> Demuth, J., M. DeMaria, and J.A. Knaff, 2006: Improvement of advanced microwave sounder unit tropical cyclone intensity and size estimation algorithms. *J. Appl. Meteor.*, **45**, 1573-1581.

To aid in the processing of these data into the Tier 1 data files, the EBT dataset has been first converted from its native ASCII text format into a separate, self-contained Common Network Data Format (NetCDF). Dr. Mark DeMaria has granted permission for the Tropical Cyclone Data Project (TCDP) to provide the NetCDF version of the EBT dataset as a convenience to researchers wishing to use the EBT dataset directly. We ask that users remember that the ASCII text version constitutes the official version of the dataset and that it is possible that errors may have been introduced into the NetCDF version during the conversion process.

As discussed in chapter 6 of Vigh (2010), the size parameters included in the BT data set (in particular, the radius of maximum winds) have not always had a basis in actual data. In most cases, these values are operationally analyzed values. In some cases however, these values appear to be simply placeholders or were chosen for the purpose of initializing the operational numerical models and simply copied over from the CARQ lines of the "a-decks" files which contain the operational forecasts of the models (personal communication, Buck Sampson 2010). In other cases, the radius of maximum winds was held steady for long periods of time with no change. Thus, the BT radius of maximum wind data in this data set may or may not reflect information about the actual storm and should certainly not be considered to be "best track" in the same sense as intensity and track, which undergo post-season evaluation by experts. The VDM+ v1.001 dataset uses v2.01 of the EBT dataset files that were posted on 22 February 2013.

SHIPS Developmental Data Set

VDM+ includes another dataset to provide information about the tropical cyclone's environment and structure. These parameters are derived from model analyses of the Global Forecasting System (GFS) model (prior to 2000, the NCEP/NCAR reanalysis is used) and various satellite data.

The SHIPS developmental data set and additional documentation is freely available for download at: http://rammb.cira.colostate.edu/research/tropical_cyclones/ships/. Additional documentation is available at: http://rammb.cira.colostate.edu/research/tropical_cyclones/ships/docs/SHIPS_predictor_file_2013.doc

Because the ASCII-formatted SHIPS development data set file can be very computationally-intensive to read by interpreted languages such as NCL and Matlab, it has first been converted to a separate, self-contained NetCDF file. Data values in the NetCDF file have been converted from the original scaled units of the ASCII file into the respective physical units for each parameter, according to the conventions described in the SHIPS development data set documentation. Dr. Mark DeMaria has granted permission for TCDP to provide the NetCDF version of the SHIPS development dataset as a convenience to researchers wishing to use the SHIPS dataset directly. We ask that users remember that the ASCII text version constitutes the official version of the dataset and that it is possible that errors may have been introduced into the NetCDF version during the conversion process.

Relevant references for the SHIPS Development Dataset:

- DeMaria, M., and J. Kaplan, 1994: A statistical hurricane intensity prediction scheme (SHIPS) for the Atlantic basin. *Wea. Forecasting*, **9**, 209 220. PDF
- Kaplan, J., and M. DeMaria, 1995: A simple empirical model for predicting the decay of tropical cyclone winds after landfall. *J. Appl. Meteor.*, **34**, 2499 2512. PDF
- DeMaria, M., and J. Kaplan, 1999: An updated statistical hurricane intensity prediction scheme (SHIPS) for the Atlantic and eastern north Pacific basins. *Wea. Forecasting*, **14**, 326-337. PDF
- Kaplan, J., and M. DeMaria, 2001: On the decay of tropical cyclone winds after landfall in the New England area. *J. Appl. Meteor.*, **40**, 280-286. PDF
- Kaplan, J., and M. DeMaria, 2003: Large-scale characteristics of rapidly intensifying tropical cyclones in the North Atlantic basin, *Wea. Forecasting*, **18**, 1093-1108. PDF
- DeMaria, M., M. Mainelli, L.K. Shay, J.A. Knaff and J. Kaplan, 2005: Further Improvements in the Statistical Hurricane Intensity Prediction Scheme (SHIPS). *Wea. Forecasting*, **20**, 531-543. PDF
- DeMaria, M., J.A. Knaff, and J. Kaplan, 2006: On the decay of tropical cyclone winds crossing narrow landmasses, *J. Appl. Meteor.*, **45**, 491-499. PDF
- Jones, T. A., D. J. Cecil, and M. DeMaria, 2006: Passive Microwave-Enhanced Statistical Hurricane Intensity Prediction Scheme. *Wea. Forecasting*, **21**, 613-635. PDF

- DeMaria, M., 2009: A simplified dynamical system for tropical cyclone intensity prediction. *Mon. Wea. Rev.*, **137**, 68-82. PDF
- DeMaria, M. 2010: Tropical Cyclone Intensity Change Predictability Estimates Using a Statistical-Dynamical Model, 29th Conference on Hurricanes and Tropical Meteorology, *Amer. Meteorol. Soc.*, Tucson, AZ. PDF
- Kaplan, J., M. DeMaria, and J.A. Knaff, 2010: A Revised Tropical Cyclone Rapid Intensification Index for the Atlantic and Eastern North Pacific Basins. *Wea. Forecasting*, **25**, 220-241. PDF
- Schumacher, A.S., M. DeMaria, and J. Knaff, 2013: Summary of the New Statistical-Dynamical Intensity Forecast Models for the Indian Ocean and Southern Hemisphere and Resulting Performance. JTWC Project Final Report.

The VDM+ v1.001 dataset uses the SHIPS development dataset files that were posted on 10 May 2013.

Raw Vortex Data Messages

VDMs for the Atlantic, Eastern Pacific, and Central Pacific basins for 1989-2012 have been gleaned from several sources including NHC's ftp and http recon archives, the Florida State University archive, and the personal archives of Mark Zimmer and Steve Feuer. Scripts have been used to clean up the 5616 unique VDMs by removing extraneous unprintable characters and then organizing them into text files by storm and year.

Since this data format originates back from the 1950's when reports were transmitted by voice over shortwave radio, a designator representing a letter from the phonetic alphabet (ALPHA, BRAVO, CHARLIE, etc.) is used for each data line of the VDM. A detailed guide to decoding the VDMs is available in Table 5-2 of the National Hurricane Operations Plan (2015, available online at: http://www.ofcm.gov/nhop/15/nhop15.htm). Numerous hand edits were necessary to correct errors so that the data could be read by an automated reader. Most of these errors are typographical errors or variants in formatting or abbreviation. Often, the on-board operator might have put a '0' in place of a 'O' or vice versa, used the wrong units (e.g., 'M' instead of 'MB'), or otherwise made a formatting error in which it was easy to make a correction based on the context. Some errors involved putting data on the wrong lines (such as transposing the information on lines L. and M.). In other cases, it was not possible to determine a plausible correction so no correction has been attempted. In 1998, the VDM format was changed so that the repeated fix information (old item N.) was eliminated. This means that the new item N. corresponds to the old item O., new O. corresponds to the old P., and the new P. corresponds to the old Q. The new VDMs do not have item Q.

Some VDMs may be missing, especially from some of the NOAA missions that were being conducted for research purposes instead of operational reconnaissance. VDMs *may* have been generated from some of these research missions, but they may now exist only on hardcopy at the Aircraft Operations Center (AOC, personal communication, Barry Damiano 2009). Whether apocryphal or not, those NOAA VDMs have not been included here. Also due to a change in data archiving procedures on NHC's servers, VDMs from the INVEST and CYCLONE stages of a storm may not always be included from 2005 onward. Prior to 2005, NHC's recon archive kept the VDMs in directories organized by storm name. Beginning in 2005, VDMs were archived in one large directory by year, making it hard to keep definitively assign INVEST or CYCLONE

VDMs to the proper system. The author has made efforts to include the tropical depression stages since 2005, but some of these VDMs have undoubtedly been left out. Starting in 2007, the unique STORMID has been included in the VDM format, so VDMs from systems should always be included now once the system is operationally designated as a tropical depression.

The automated reader has been designed with certain data consistency checks (such as looking for standard units). In some rare cases, uncorrectable data will not clear these checks; when this occurs, such data items will subsequently be left out of the data set.

The resulting cleaned and hand-edited VDMs, organized by storm and year, are available at: http://verif.rap.ucar.edu/tcdata/vortex/sources/raw_VDMs_v1.000/. Documentation of the hand-edits that have been made to the VDMs is provided at: http://verif.rap.ucar.edu/tcdata/vortex/sources/raw_VDMs_v1.000/NOTES_documentation_of_VDM_edits_for_v1.000.txt.

Tier 1 Data: An Individual Processed Data File for each Tropical Cyclone

File Naming Convention

Each Tier 1 data file is named using the storm identification code (*STORMID*, e.g. AL122008) in NHC's master storm table along with the given *STORMNAME* from the master storm table. The naming convention for the individual Tier 1 data files is: <*STORMID*>_<*STORMNAME*>.nc (e.g., AL092002_HANNA.nc).

Data Flow

The NCAR Command Language (NCL) program 'vortex_create_individual_storm_files.ncl' is used to read the various Tier 0 source data, storm by storm. The data for each storm are processed and translated, and are then written out to an individual standardized NetCDF file – a Tier 1 data file. This program reads the BT and VDM source data directly, while data from the NHC master storm table, the EBT, and the SHIPS data are read from the self-contained NetCDF files that were created prior to the processing.

For each storm, the b-deck and associated BT parameters are merged and matched with the EBT parameters as previously described. This results in 35 parameters for each time point in the BT dataset. Additionally, there are 17 parameters that constitute various storm-level metadata for the storm. While the names of all of these parameters are given the prefix "BT" in the resulting NetCDF file to designate that they came from Best Track sources (e.g. Master Storm Table, the BT database, and EBT dataset), however the user is reminded that only the following parameters are actually considered to have undergone post-season best track analysis: TY, LAT, LON, VMAX, MSLP, and the various wind radii parameters (from 2004 onward).

All of the SHIPS environmental and satellite-based predictors are read from the SHIPS development dataset NetCDF file. The VDM+ dataset includes the predictors calculated from the model analysis time (hour 0). This results in 109 parameters, all of which are given the prefix "SHIPS".

The raw VDM text file for each storm is read and translated to extract the intensity and structure parameters following the data processing stages 1-3 as described in the supplement to Vigh et al (2012). The original raw lines from the VDM contents are stored as 20 parameters, each of which includes the word “raw” in the parameter name (e.g., ‘raw_foxtrot’, ‘raw_golf’, etc). From these data, 150 other parameters are extracted and translated. Each of these parameters are given the prefix “VDM_FIX”. 13 additional parameters are derived or calculated from the VDM and BT data. These are given the prefix “CALC_VDM” or “CALC_BT”. The final result is an individual NetCDF file for each storm that contains 355 parameters.

The Difference between Tier 1 and Tier 2 Data Files

Since the Tier 1 and Tier 2 parameters are the same, the actual data and coordinate parameters for Tier 1 data will be documented in the section for Tier 2 Data below. Why then are there two tiers of data? The answer is that the processing and translation of the raw VDMs is computationally intensive due to the multitude of string parsing and if-check branches that must be followed. Because each storm can be treated as a logical unit from a data perspective, it is helpful to process the data set one storm at a time and put these into an individual standardized NetCDF file. Later, these files can then be quickly aggregated into a data set for all storms (Tier 2). Processing by storm also allows easier debugging and data logging and opens up the door to real-time applications.

While the data are the same, the NetCDF file structures of the two data files are somewhat different. The Tier 1 individual storm files have three main dimensions: *nbest* (the # of 6-hourly best track points for that that particular storm), *nship* (the # of data points for which SHIPS development data are available), and *nfix* (the # of valid and unique VDM fixes for that particular storm). Each data parameter array is indexed in the respective dimension. For instance, Hurricane Hugo (*STORMID* = AL111989) has *nbest* = 61, *nship* = 50, and *nfix* = 44. This means that all of the BT-prefixed parameters will contain 61 values each. The SHIPS-prefixed parameters will contain 50 values each. The VDM-prefixed parameter arrays will contain 44 values: one value for each of the 44 unique fix times. In contrast, all of the Tier 2 data arrays have a fixed size, which is large enough to accommodate all data from the longest lived and most sampled storms.

A special note should be made concerning strings. The NetCDF files in this data set are written according to netCDF4 classic specifications. Since strings are not a valid data type, the data have been stored as character arrays. Therefore, each string parameter has its own named dimension that follows the naming convention: ‘<*parameter_name*>_Str_Len’. When reading the NetCDF data file, it is easy to convert these back to strings using ‘chartostring’ in NCL or the equivalent function in whatever language is being used (Note that Matlab is unable to read “string” variables with the standard *ncread* function – instead it is necessary to use *h5read*). The Tier 2 data set also has string length dimensions that are different from the Tier 1 string length dimensions owing to the fact that the Tier 2 data set must accommodate the longest strings for a given parameter for all storms.

Tier 2 Data: Structure and Indexing of the Aggregated Dataset File

It is recommended that users of the VDM+ dataset use the Tier 2 data since these data are packaged in a consistent and easy-to-access format with consistent indexing in both the time and storm dimensions. The entire Tier 2 dataset exists in one large NetCDF file whose file name (e.g., AL_aggregated_structure_and_intensity_dataset_v1.000_20131201.nc) indicates the basin (‘AL’,’EP’, or ‘COMBINED’), the version number of the dataset (e.g. ‘v1.000’), and the date that the NetCDF was aggregated from the individual Tier 1 storm data files (e.g. ‘COMBINED_aggregated_structure_and_intensity_dataset_20101012.nc’).

Reading the Tier 2 Data

Any NetCDF-enabled data analysis program should be able to access the Tier 2 data file. NCAR Command Language (NCL) was used to create the data set from the raw data sources and several NCL modules are provided to users to facilitate reading and opening the NetCDF data file. These modules open up the aggregated data file and automatically read all 355 parameters into memory, along with their coordinate dimensions.

Because NetCDF files cannot represent parameters of type string or Boolean (logical), parameters of this type in the data set have been stored as character and integer arrays, respectively, in the NetCDF data files. The provided reader modules automatically handle the conversion of parameters back to strings and Booleans based on attributes *intended_type* and *intended_FillValue* which are given in the NetCDF data files. The reader module also ensures that the proper _FillValue attribute is used to indicate missing data for all data types. If a user chooses to use the provided reader modules, all parameters will be of the type indicated by the *intended_type* attribute, with a fill value indicated by *intended_FillValue*. If a user chooses to use some other program to read in the data set, he or she will have to handle the conversion back to string and Boolean parameters, the setting of parameter types, and fill values on their own. Because this could be a serious hassle, it is highly recommended that the provided reader modules be used.

Using the reader modules is very easy and can be done by simply including several lines of code before the *begin* statement of a user’s NCL script. The read module gets executed before the begin statement, so all data parameters in the NetCDF file are loaded into memory and are accessible throughout the script as global parameters. The provided script ‘vortex_example_reader.ncl’ illustrates how to do this. A user can simply modify this program after the *begin* statement and begin adding code to analyze the data.

Types of Data Parameters and Their Dimensioning in the Tier 2 Data Set File

There are three main types of parameters: (1) storm description parameters, (2) dimension parameters, and (2) data parameters.

The first parameter type, *storm description parameters*, provide storm-level descriptive data. The names of storm description parameters will always include the prefix “STORM” after their primary category (e.g. “BT_STORM_stormid”). There will always be just one value per storm, allowing a user to easily search for and identify the data belonging to a particular storm. The user may also use the storm description parameters to create a subset of the dataset based on numerous criteria. For instance, all storms that had aircraft data can be obtained by creating an index of the storms for which ‘VDM_datapresent’ = ‘True’. Similarly, one may choose all of the storms from a particular year, all the storms above a maximum intensity, etc. All of the storm description parameters have a main dimension size that is the number of storms. Parameters of type string will have an additional dimension that provides the length of the longest string in the array (more on this is provided in a couple paragraphs).

Parameters of the second type, *dimension parameters*, provide information about the coordinate time dimension for each type of data (BT, SHIPS, or VDM). Time coordinate dimensions are provided in two forms: one using a string of the format ‘YYYYMMDDHHMM’ that represents the 4-digit year ‘YYYY’, the 2-digit month ‘MM’, the 2-digit day ‘DD’, the 2-digit hour ‘HH’, and the 2-digit minute ‘MM’ that corresponds to the time of the parameter value. All of the data parameters are dimensioned using these time dimensions. The time coordinate dimension is also represented by another equivalent form ‘timeoffset’ which are the seconds since a base time. Throughout the VDM+ dataset, the same base time is used: “seconds since 1970-01-01 00:00:00 UTC”. This time coordinate allows for easy numerical comparison between various time points to determine which is most recent, and how far apart two time points may be. For the VDM data parameters, another additional dimension is provided that represents the unique fix number (starting from 0 and incrementing by one for each unique fix).

The third parameter type, *data parameters*, provide data values for each time point, whether it be a BT data time point, a SHIPS data time point, or a VDM fix. Such data parameters are indicated using the following prefixes: ‘BT’, ‘SHIPS’, ‘VDM_FIX’, ‘CALC_BT’, or ‘CALC_VDM’.

A full listing of the file dimensions and parameters may be obtained by using the *ncdump* command with the –h option (e.g., ‘ncdump –h <*filename*>’). The main file dimension sizes are: the number of storms (*nstorms)*, the maximum number of 6-hourly or off-synoptic Best Track data points that are permitted to be stored for a given storm (*nmax_BT_points*), the maximum number of 6-hourly SHIPS data points that are allowed to be stored for a given storm, and the maximum number of VDM fixes that are allowed to be stored for a given storm:

```
nstorms = 382

nmax_BT_points = 160
nmax_SHIPS_points = 160
nmax_VDM_fixes = 160
```

Thus, most of the Tier 2 data parameters are two dimensional arrays whose first dimension is [*nstorms*] and whose second dimension is one of the following: [*nmax_BT_points, nmax_SHIPS_points*, *nmax_VDM_fixes*]. While this dimensioning approach necessarily causes wasted space since most storms do not have anywhere close to 160 BT points, SHIPS points, or VDM fixes, the fixed array dimensions considerably simplify data access.

The data parameters that are of type ‘string’ will have an additional dimension size that represents the number of characters in the longest string in the array (including termination character). All of these additional dimensions are listed below. **Please note that these**

dimensions lengths may change as additional years of data are added to the data set. As such, please read the dimensions and use their current values rather than hard-coding the values below into your program (or use the provided example reader which will handle this for you).

BT_STORM_stormid_StrLen = 9
BT_STORM_status_StrLen = 8
BT_STORM_bdate_StrLen = 11
BT_STORM_edate_StrLen = 11
BT_STORM_peak_StrLen = 3
BT_STORM_stormbasin_StrLen = 3
BT_STORM_stormyear_StrLen = 5
BT_STORM_stormnumber_StrLen = 3
BT_STORM_subregion_StrLen = 2
BT_STORM_stormname_StrLen = 11
BT_STORM_cy_StrLen = 3
BT_STORM_all_basins_StrLen = 3
BT_STORM_all_subregions_StrLen = 2
BT_STORM_all_stormnames_StrLen = 26
BT_STORM_all_tys_StrLen = 39
BT_yyyymmddhhmm_StrLen = 13
BT_ty_StrLen = 3
BT_stormname_StrLen = 11
BT_initials_StrLen = 8
VDM_STORM_stormname_StrLen = 10
VDM_FIX_transmission_yyyymmddhhmm_StrLen = 13
VDM_FIX_fix_yyyymmddhhmm_StrLen = 13
VDM_FIX_fix_identifier_StrLen = 100
VDM_FIX_month_StrLen = 3
VDM_FIX_corrected_string_StrLen = 4
VDM_FIX_corrected_number_StrLen = 3
VDM_FIX_raw_mission_StrLen = 77
VDM_FIX_header_StrLen = 7
VDM_FIX_office_StrLen = 5
VDM_FIX_transmission_day_StrLen = 3
VDM_FIX_transmission_time_StrLen = 5
VDM_FIX_mission_corrected_string_StrLen = 4
VDM_FIX_mission_corrected_number_StrLen = 3
VDM_FIX_raw_flight_StrLen = 73
VDM_FIX_aircraft_identification_StrLen = 6
VDM_FIX_wx_mission_identification_StrLen = 6
VDM_FIX_stormname_identification_StrLen = 10
VDM_FIX_ob_number_StrLen = 3
VDM_FIX_basin_identifier_StrLen = 3
VDM_FIX_designated_cyclone_number_StrLen = 3
VDM_FIX_flight_corrected_string_StrLen = 4
VDM_FIX_flight_corrected_number_StrLen = 3
VDM_FIX_stormid_identification_StrLen = 9
VDM_FIX_raw_storm_StrLen = 79
VDM_FIX_raw_alpha_StrLen = 67
VDM_FIX_day_StrLen = 3

```
VDM_FIX_time_StrLen = 5
VDM_FIX_raw_bravo_lat_StrLen = 67
VDM_FIX_raw_bravo_lon_StrLen = 67
VDM_FIX_raw_delta_StrLen = 67
VDM_FIX_raw_echo_StrLen = 67
VDM_FIX_raw_foxtrot_StrLen = 123
VDM_FIX_raw_golf_StrLen = 104
VDM_FIX_raw_hotel_StrLen = 90
VDM_FIX_raw_india_StrLen = 67
VDM_FIX_raw_juliet_StrLen = 67
VDM_FIX_raw_kilo_StrLen = 67
VDM_FIX_raw_lima_StrLen = 108
VDM_FIX_eyewall_completeness_StrLen = 7
VDM_FIX_eyewall_definition_StrLen = 7
VDM_FIX_eyewall_weakness_direction_StrLen = 11
VDM_FIX_raw_mike_StrLen = 109
VDM_FIX_eye_shape_StrLen = 11
VDM_FIX_raw_november_StrLen = 67
VDM_FIX_determination_string_StrLen = 10
VDM_FIX_determination_decoded_StrLen = 45
VDM_FIX_level_string_StrLen = 30
VDM_FIX_level_flight_StrLen = 4
VDM_FIX_level_decoded_StrLen = 20
VDM_FIX_raw_oscar_StrLen = 67
VDM_FIX_navigational_accuracy_string_StrLen = 5
VDM_FIX_meteorological_accuracy_string_StrLen = 25
VDM_FIX_raw_remark_StrLen = 333
VDM_FIX_cleaned_remark_StrLen = 428
VDM_FIX_maximum_flight_level_wind_relative_quadrant_StrLen = 4
VDM_FIX_maximum_flight_level_wind_day_StrLen = 3
VDM_FIX_maximum_flight_level_wind_time_StrLen = 5
VDM_FIX_maximum_flight_level_wind_yyyymmddhhmm_StrLen = 13
VDM_FIX_outbound_maximum_flight_level_wind_relative_quadrant_StrLen = 3
VDM_FIX_outbound_maximum_flight_level_wind_day_StrLen = 3
VDM_FIX_outbound_maximum_flight_level_wind_time_StrLen = 5
VDM_FIX_outbound_maximum_flight_level_wind_yyyymmddhhmm_StrLen = 13
VDM_FIX_secondary_maximum_flight_level_wind_relative_quadrant_StrLen = 4
VDM_FIX_secondary_maximum_flight_level_wind_day_StrLen = 3
VDM_FIX_secondary_maximum_flight_level_wind_time_StrLen = 5
VDM_FIX_secondary_maximum_flight_level_wind_yyyymmddhhmm_StrLen = 13
VDM_FIX_maximum_surface_wind_relative_quadrant_StrLen = 3
VDM_FIX_maximum_surface_wind_day_StrLen = 3
VDM_FIX_maximum_surface_wind_time_StrLen = 5
VDM_FIX_maximum_surface_wind_yyyymmddhhmm_StrLen = 13
VDM_FIX_outbound_maximum_surface_wind_relative_quadrant_StrLen = 3
VDM_FIX_outbound_maximum_surface_wind_day_StrLen = 3
VDM_FIX_outbound_maximum_surface_wind_time_StrLen = 5
VDM_FIX_outbound_maximum_surface_wind_yyyymmddhhmm_StrLen = 13
VDM_FIX_combined_maximum_flight_level_wind_yyyymmddhhmm_StrLen = 13
VDM_FIX_combined_maximum_surface_wind_yyyymmddhhmm_StrLen = 13
VDM_FIX_minimum_sea_level_pressure_extrapolated_level_string_StrLen = 14
```

VDM_FIX_lightning_direction_StrLen = 10
VDM_FIX_hail_direction_StrLen = 4
VDM_FIX_banding_presentation_StrLen = 5
VDM_FIX_banding_direction_StrLen = 7
VDM_FIX_radar_presentation_StrLen = 10
SHIPS_yyyymmddhhmm_StrLen = 13

Fill Values

Throughout the data set, various fill values (sometimes known as ‘missing value’) are used to indicate missing data. The actual value used to represent the missing value varies according to the parameter type. The fill value for each parameter is indicated in the NetCDF data file using the ‘*_FillValue’* attribute. The following fill values are used:

Parameter Type	Fill Value
integer	-999
float	-9999
string	“” (an empty string obtained by two double quote marks with no space inside)
double	-99999
character	'\0' (the null character, obtained by ‘inttochar(0)’ in NCL)
logical	_Missing

Complete Manifest of the Tier 2 Data Parameters

This section provides a listing of all of the storm description (type 1) and data parameters (type 2) contained in the Tier 2 data set file. Each parameter definition normally consists of seven lines. Here is an example:

```
char BT_STORM_stormid(nstorms, BT_STORM_stormid_StrLen) ;
        BT_STORM_stormid:_FillValue = "" ;
        BT_STORM_stormid:intended_type = "string" ;
        BT_STORM_stormid:intended_FillValue = "" ;
        BT_STORM_stormid:units = "none" ;
        BT_STORM_stormid:description = "ATCF STORM ID used to designate a given storm in
                    the Best Track dataset b-deck file and the Storm Table (e.g. \'AL012013\')" ;
        BT_STORM_stormid:long_name = "BT_STORM_stormid" ;
```

The first line gives the type of the parameter *as it is represented in the NetCDF data file*, followed by the parameter name, with the parameter’s dimensions listed in parentheses. Parameter types may include ‘int’ for “integer”, ‘char’ for “character”, as well as “float” or “double”. The next six lines provide attributes that are associated with the parameter. The second line gives the _FillValue used for that parameter in NetCDF file. The third line gives the intended type, i.e., the type that the parameter will be converted to when read using the provided modules. The fourth line gives the intended fill value. Once the parameters have been read into memory in the provided example read script, the parameter will have the intended type and fill value indicated by these two attributes, not the type and _FillValue that the parameter was represented with in the NetCDF file. So for this example, the parameter 'BT_STORM_stormid' is represented *in the data file* as a character array of size [*nstorms*, *BT_STORM_stormid_StrLen*] with a fill value attribute of "". Once it has been read into memory by the example script, it will be a string array that uses "" as the fill value. The fifth line provides the units of the data quantity. The sixth line provides a detailed description of the variable, including information on which part of the VDM it was translated from (if applicable). The seventh line provides a '*long_name*' attribute which is identical to the parameter name.

In the interest of brevity, the '_FillValue', `intended_type', 'intended_FillValue', and 'long_name' attributes are omitted in the following manifest. A user can obtain the full manifest with these attributes by using 'ncdump –h <*filename*>'. Also, in the output of ncdump, certain characters are protected (e.g. end-of-line and the single quotation mark), and every line is followed by a semicolon. In the following manifest, these protected characters and semicolons have been removed to make the descriptions more readable.

Several Boolean parameters are used to indicate whether data are present for the main basic data types. These are:

VDM_datapresent
SHIPS_datapresent
EBT_datapresent

As an example, if 'VDM_datapresent' = 'False', the user can expect that all 'VDM_' parameters will contain missing values.

BT Storm Description Parameters

```
        char BT_STORM_stormid(nstorms, BT_STORM_stormid_StrLen)
                BT_STORM_stormid:units = "none"
                BT_STORM_stormid:description = "ATCF STORM ID used to designate a given
storm in the Best Track dataset b-deck file and the Storm Table (e.g. 'AL012013')"

        char BT_STORM_status(nstorms, BT_STORM_status_StrLen)
                BT_STORM_status:units = "none"
                BT_STORM_status:description = "ATCF storm status for a given storm, from the
operational or archive Storm Table (e.g. 'WARNING', 'METWATCH', or 'ARCHIVE')"

        char BT_STORM_bdate(nstorms, BT_STORM_bdate_StrLen)
                BT_STORM_bdate:units = "UTC"
                BT_STORM_bdate:description = "Beginning time/date of the record for this
storm in YYYYMMDDHH format (string), from the operational or archive Storm Table (e.g.
'2012102918')"

        char BT_STORM_edate(nstorms, BT_STORM_edate_StrLen)
```

BT_STORM_edate:units = "UTC"
BT_STORM_edate:description = "Ending time/date of the record for this storm in YYYYMMDDHH format (string), from the operational or archive Storm Table (e.g. '2012103112'). If '9999999999', then the storm is still ongoing or its ATCF file has not been archived yet."

char BT_STORM_peak(nstorms, BT_STORM_peak_StrLen)
BT_STORM_peak:units = ""
BT_STORM_peak:description = "2-letter ATCF code representing the peak development during the storm's lifetime, from the operational or archive Storm Table. 'DB' = disturbance, 'TD' = tropical depression, 'TS' = tropical storm, 'TY' = typhoon, 'ST' = super typhoon, 'TC' = tropical cyclone, 'HU' = hurricane, 'SD' = subtropical depression, 'SS' = subtropical storm, 'EX' = extratropical systems, 'IN' = inland, 'DS' = dissipating, 'LO' = low, 'WV' = tropical wave, 'ET' = extrapolated, 'XX' = unknown."

char BT_STORM_stormbasin(nstorms, BT_STORM_stormbasin_StrLen)
BT_STORM_stormbasin:units = "none"
BT_STORM_stormbasin:description = "2-letter ATCF identifier for the basin that the storm is located in, from the operational or archive Storm Table. 'WP' = West Pacific, 'IO' = Indian Ocean, 'SH' = Southern Hemisphere, 'CP' = Central Pacific, 'EP' = Eastern Pacific, 'AL' = North Atlantic, 'SL' = South Atlantic."

char BT_STORM_stormyear(nstorms, BT_STORM_stormyear_StrLen)
BT_STORM_stormyear:units = "UTC"
BT_STORM_stormyear:description = "4-digit year (string) in which the storm first was designated, from the operational or archive Storm Table (string)"

char BT_STORM_stormnumber(nstorms, BT_STORM_stormnumber_StrLen)
BT_STORM_stormnumber:units = "none"
BT_STORM_stormnumber:description = "2-digit ATCF storm number (string) in the extended best track dataset, from the operational or archive Storm Table (string)"

char BT_STORM_subregion(nstorms, BT_STORM_subregion_StrLen)
BT_STORM_subregion:units = "none"
BT_STORM_subregion:description = "1-letter ATCF subbasin code (string) to designate the subbasin that the storm is located in, from the operational or archive Storm Table. 'A' = Arabian Sea, 'B' = Bay of Bengal, 'C' = Central Pacific, 'E' = Eastern Pacific, 'L' = Atlantic, 'P' = South Pacific (135E - 120W), 'Q' = South Atlantic, 'S' = South Indian Ocean (20E to 135E), and 'W' = West Pacific."

char BT_STORM_stormname(nstorms, BT_STORM_stormname_StrLen)
BT_STORM_stormname:units = "none"
BT_STORM_stormname:description = "Literal stormname, 'NONAME', or 'INVEST', from the operational or archive Storm Table. Prior to 1999, 'TCcyx' was used, where 'cy' = annual cyclone number 01 through 99, 'x' = subregion code where 'A' = Arabian Sea, 'B' = Bay of Bengal, 'C' = Central Pacific, 'E' = Eastern Pacific, 'L' = Atlantic, 'P' = South Pacific

(135E - 120W), 'Q' = South Atlantic, 'S' = South Indian Ocean (20E to 135E), and 'W' = West Pacific."

int BT_STORM_npts(nstorms)
BT_STORM_npts:description = "Number of data points that exist for this storm in the Best Track b-deck file (including off-synoptic time points)"
BT_STORM_npts:units = "count"

char BT_STORM_cy(nstorms, BT_STORM_cy_StrLen)
BT_STORM_cy:units = "none"
BT_STORM_cy:description = "Annual cyclone number (string) 01 through 99, from the Best Track b-deck file"

char BT_STORM_all_basins(nstorms, BT_STORM_all_basins_StrLen)
BT_STORM_all_basins:units = "none"
BT_STORM_all_basins:description = "All basin codes that are assigned to this storm in the Best Track b-deck file. 2-letter ATCF identifier for the basin that the storm is located in, from the operational or archive Storm Table. 'WP' = West Pacific, 'IO' = Indian Ocean, 'SH' = Southern Hemisphere, 'CP' = Central Pacific, 'EP' = Eastern Pacific, 'AL' = North Atlantic, 'SL' = South Atlantic. This value is obtained by concatenating all unique values of BT_stormbasin that exist in the Best Track b-deck file for this storm."

char BT_STORM_all_subregions(nstorms, BT_STORM_all_subregions_StrLen)
BT_STORM_all_subregions:units = "none"
BT_STORM_all_subregions:description = "All subregion codes that are assigned to this storm in the Best Track b-deck file. 1-letter ATCF subbasin code (string) to designate the subbasin that the storm is located in: 'A' = Arabian Sea, 'B' = Bay of Bengal, 'C' = Central Pacific, 'E' = Eastern Pacific, 'L' = Atlantic, 'P' = South Pacific (135E - 120W), 'Q' = South Atlantic, 'S' = South Indian Ocean (20E to 135E), and 'W' = West Pacific. This value is obtained by concatenating all unique values of BT_subregion that exist in the Best Track b-deck file for this storm."

char BT_STORM_all_stormnames(nstorms, BT_STORM_all_stormnames_StrLen)
BT_STORM_all_stormnames:units = "none"
BT_STORM_all_stormnames:description = "All stormnames that are assigned to this storm in the Best Track b-deck file. Literal stormname, 'NONAME', or 'INVEST'. Prior to 1999, 'TCcyx' was used, where 'cy' = annual cyclone number 01 through 99, 'x' = subregion code where 'A' = Arabian Sea, 'B' = Bay of Bengal, 'C' = Central Pacific, 'E' = Eastern Pacific, 'L' = Atlantic, 'P' = South Pacific (135E - 120W), 'Q' = South Atlantic, 'S' = South Indian Ocean (20E to 135E), and 'W' = West Pacific. This value is obtained by concatenating all unique values of BT_stormname that exist in the Best Track b-deck file for this storm."

char BT_STORM_all_tys(nstorms, BT_STORM_all_tys_StrLen)
BT_STORM_all_tys:units = "none"
BT_STORM_all_tys:description = "All ATCF codes representing the level of tropical cyclone development during the storm's lifetime, from the operational or archive Storm

Table. 'DB' = disturbance, 'TD' = tropical depression, 'TS' = tropical storm, 'TY' = typhoon, 'ST' = super typhoon, 'TC' = tropical cyclone, 'HU' = hurricane, 'SD' = subtropical depression, 'SS' = subtropical storm, 'EX' = extratropical systems, 'IN' = inland, 'DS' = dissipating, 'LO' = low, 'WV' = tropical wave, 'ET' = extrapolated, 'XX' = unknown. This value is obtained by concatenating all unique values of BT_ty that exist in the Best Track b-deck file for this storm."

BT Dimension Parameters

char BT_yyyymmddhhmm(nstorms, nmax_BT_points, BT_yyyymmddhhmm_StrLen)

BT_yyyymmddhhmm:units = "UTC"

BT_yyyymmddhhmm:description = "Date/time (string representation) corresponding to each point in the Best Track b-deck file, yyyymmddhhmm: yyyy = 4-digit year, mm = 2-digit month, dd = 2-digit day, hh = 2-digit hour, mm = 2-digit minute."

int BT_timeoffset(nstorms, nmax_BT_points)

BT_timeoffset:description = "Time corresponding to each point in the Best Track b-deck file, as measured as an offset from a base time"

BT_timeoffset:units = "seconds since 1970-01-01:00:00:00 UTC"

BT Data Parameters

float BT_lat(nstorms, nmax_BT_points)

BT_lat:description = "Latitude of storm center, for each time in the Best Track b-deck file"

BT_lat:units = "degrees_north"

float BT_lon(nstorms, nmax_BT_points)

BT_lon:description = "Longitude of storm center, for each time in the Best Track b-deck file"

BT_lon:units = ""

int BT_mslp(nstorms, nmax_BT_points)

BT_mslp:description = "Minimum sea level pressure, for each time in the Best Track b-deck file"

BT_mslp:units = "millibar"

int BT_vmax(nstorms, nmax_BT_points)

BT_vmax:description = " Maximum sustained (1-min average) surface wind speed (at 10 m, for an open ocean exposure), for each time in the Best Track b-deck file"

BT_vmax:units = "knot"

char BT_ty(nstorms, nmax_BT_points, BT_ty_StrLen)

BT_ty:units = "none"

BT_ty:description = "ATCF code representing the level of tropical cyclone development, for each time in the Best Track b-deck file. 'DB' = disturbance, 'TD' = tropical depression, 'TS' = tropical storm, 'TY' = typhoon, 'ST' = super typhoon, 'TC' = tropical cyclone, 'HU' = hurricane, 'SD' = subtropical depression, 'SS' = subtropical storm, 'EX' = extratropical systems, 'IN' = inland, 'DS' = dissipating, 'LO' = low, 'WV' = tropical wave, 'ET' = extrapolated, 'XX' = unknown."

int BT_radp(nstorms, nmax_BT_points)
BT_radp:description = "Pressure of the outer closed isobar, for each time in the Best Track b-deck file"
BT_radp:units = "millibar"

int BT_rrp(nstorms, nmax_BT_points)
BT_rrp:description = "Radius of the outer closed isobar, for each time in the Best Track b-deck file"
BT_rrp:units = "nautical_mile"

int BT_mrd(nstorms, nmax_BT_points)
BT_mrd:description = "Radius of surface maximum wind speed, for each time in the Best Track b-deck file"
BT_mrd:units = "nautical_mile"

int BT_gusts(nstorms, nmax_BT_points)
BT_gusts:description = "Estimated wind speed of the maximum gust, for each time in the Best Track b-deck file"
BT_gusts:units = "knot"

int BT_eye(nstorms, nmax_BT_points)
BT_eye:description = "Eye diameter (measured by aircraft data or estimated using satellite imagery), for each time in the Best Track b-deck file"
BT_eye:units = "nautical_mile"

char BT_subregion(nstorms, nmax_BT_points)
BT_subregion:description = "ATCF subregion code assigned to the storm for each time in the Best Track b-deck file. 1-letter code (string) to designate the subbasin that the storm is located in: 'A' = Arabian Sea, 'B' = Bay of Bengal, 'C' = Central Pacific, 'E' = Eastern Pacific, 'L' = Atlantic, 'P' = South Pacific (135E - 120W), 'Q' = South Atlantic, 'S' = South Indian Ocean (20E to 135E), and 'W' = West Pacific."
BT_subregion:units = ""

char BT_stormname(nstorms, nmax_BT_points, BT_stormname_StrLen)
BT_stormname:units = "none"
BT_stormname:description = "Literal stormname, 'NONAME', or 'INVEST', assigned to the storm for each time in the Best Track b-deck file. Prior to 1999, 'TCcyx' was used, where 'cy' = annual cyclone number 01 through 99, 'x' = subregion code where 'A' = Arabian Sea, 'B' = Bay of Bengal, 'C' = Central Pacific, 'E' = Eastern Pacific, 'L' = Atlantic, 'P' = South Pacific (135E - 120W), 'Q' = South Atlantic, 'S' = South Indian Ocean (20E to 135E), and 'W' = West Pacific."

```
        char BT_depth(nstorms, nmax_BT_points)
                BT_depth:description = "System depth code assigned to the storm for each time
in the Best Track b-deck file. 'D' = deep, 'M' = medium, 'S' = shallow, 'X' = unknown."
                BT_depth:units = "none"

        int BT_maxseas(nstorms, nmax_BT_points)
                BT_maxseas:description = "Maximum height of the estimated seas associated
with the storm, for each time in the Best Track b-deck file."
                BT_maxseas:units = "feet"

        char BT_initials(nstorms, nmax_BT_points, BT_initials_StrLen)
                BT_initials:units = "none"
                BT_initials:description = "Forecaster's initials (up to 3 letters) used for the
TAU=0 warning, for each time in the Best Track b-deck file"

        int BT_dir(nstorms, nmax_BT_points)
                BT_dir:description = "Storm direction in compass coordinates (0 - 359 degrees),
for each time in the Best Track b-deck file"
                BT_dir:units = "degrees"

        int BT_speed(nstorms, nmax_BT_points)
                BT_speed:description = "Translation speed of the storm center, for each time in
the Best Track b-deck file"
                BT_speed:units = "knot"

        int BT_rad34_NE(nstorms, nmax_BT_points)
                BT_rad34_NE:description = "Radius of the 34-kt wind speed in the northeast
quadrant of the storm, for each time in the Best Track b-deck file"
                BT_rad34_NE:units = "nautical_mile"

        int BT_rad34_SE(nstorms, nmax_BT_points)
                BT_rad34_SE:description = "Radius of the 34-kt wind speed in the southeast
quadrant of the storm, for each time in the Best Track b-deck file."
                BT_rad34_SE:units = "nautical_mile"

        int BT_rad34_SW(nstorms, nmax_BT_points)
                BT_rad34_SW:description = "Radius of the 34-kt wind speed in the southwest
quadrant of the storm, for each time in the Best Track b-deck file"
                BT_rad34_SW:units = "nautical_mile"

        int BT_rad34_NW(nstorms, nmax_BT_points)
                BT_rad34_NW:description = "Radius of the 34-kt wind speed in the northwest
quadrant of the storm, for each time in the Best Track b-deck file"
                BT_rad34_NW:units = "nautical_mile"
```

```
	int BT_rad50_NE(nstorms, nmax_BT_points)
		BT_rad50_NE:description = "Radius of the 50-kt wind speed in the northeast
quadrant of the storm, for each time in the Best Track b-deck file"
		BT_rad50_NE:units = "nautical_mile"

	int BT_rad50_SE(nstorms, nmax_BT_points)
		BT_rad50_SE:description = "Radius of the 50-kt wind speed in the southeast
quadrant of the storm, for each time in the Best Track b-deck file"
		BT_rad50_SE:units = "nautical_mile"

	int BT_rad50_SW(nstorms, nmax_BT_points)
		BT_rad50_SW:description = "Radius of the 50-kt wind speed in the southwest
quadrant of the storm, for each time in the Best Track b-deck file"
		BT_rad50_SW:units = "nautical_mile"

	int BT_rad50_NW(nstorms, nmax_BT_points)
		BT_rad50_NW:description = "Radius of the 50-kt wind speed in the northwest
quadrant of the storm, for each time in the Best Track b-deck file"
		BT_rad50_NW:units = "nautical_mile"

	int BT_rad64_NE(nstorms, nmax_BT_points)
		BT_rad64_NE:description = "Radius of the 64-kt wind speed in the northeast
quadrant of the storm, for each time in the Best Track b-deck file"
		BT_rad64_NE:units = "nautical_mile"

	int BT_rad64_SE(nstorms, nmax_BT_points)
		BT_rad64_SE:description = "Radius of the 64-kt wind speed in the southeast
quadrant of the storm, for each time in the Best Track b-deck file"
		BT_rad64_SE:units = "nautical_mile"

	int BT_rad64_SW(nstorms, nmax_BT_points)
		BT_rad64_SW:description = "Radius of the 64-kt wind speed in the southwest
quadrant of the storm, for each time in the Best Track b-deck file"
		BT_rad64_SW:units = "nautical_mile"

	int BT_rad64_NW(nstorms, nmax_BT_points)
		BT_rad64_NW:description = "Radius of the 64-kt wind speed in the northwest
quadrant of the storm, for each time in the Best Track b-deck file"
		BT_rad64_NW:units = "nautical_mile"

	int BT_seas_threshold(nstorms, nmax_BT_points)
		BT_seas_threshold:description = "Threshold value of the seas that the
'BT_seas_**' quadrant radii apply to, for each time in the Best Track b-deck file"
		BT_seas_threshold:units = "feet"

	int BT_seas_NE(nstorms, nmax_BT_points)
```

BT_seas_NE:description = "Radius of the seas with height given by 'BT_seas_threshold' in the northeast quadrant of the storm, for each time in the Best Track b-deck file"

BT_seas_NE:units = "nautical_mile"

int BT_seas_SE(nstorms, nmax_BT_points)

BT_seas_SE:description = "Radius of the seas with height given by 'BT_seas_threshold' in the southeast quadrant of the storm, for each time in the Best Track b-deck file"

BT_seas_SE:units = "nautical_mile"

int BT_seas_SW(nstorms, nmax_BT_points)

BT_seas_SW:description = "Radius of the seas with height given by 'BT_seas_threshold' in the southwest quadrant of the storm, for each time in the Best Track b-deck file"

BT_seas_SW:units = "nautical_mile"

int BT_seas_NW(nstorms, nmax_BT_points)

BT_seas_NW:description = "Radius of the seas with height given by 'BT_seas_threshold' in the northwest quadrant of the storm, for each time in the Best Track b-deck file"

BT_seas_NW:units = "nautical_mile"

float BT_distance_to_land(nstorms, nmax_BT_points)

BT_distance_to_land:description = "Distance to the nearest major land mass, where Trinidad is taken to be the smallest area considered to be land. Negative values indicate that the cetner is over land. Note: This parameter does not exist in the Best Track dataset. It has been added from the Extended Best Track."

BT_distance_to_land:units = "kilometer"

VDM Storm Description Parameters

int VDM_datapresent(nstorms)

VDM_datapresent:units = "none"

int VDM_STORM_read_error_condition(nstorms)

VDM_STORM_read_error_condition:description = "An error condition flag that indicates if the file containing the VDM was read: 0 = file was read properly, 1 = file was not found for this storm, 2 = an unexpected duplicative URNT12 header was found while reading the file (data processing halted so that this error could be investigated) "

VDM_STORM_read_error_condition:units = "none"

int VDM_STORM_number_read_messages(nstorms)

VDM_STORM_number_read_messages:description = "Number of Vortex Data Messages that were read from the file for this storm (including duplicates and corrected messages)"

VDM_STORM_number_read_messages:units = "count"

int VDM_STORM_number_unique_fix_times(nstorms)

VDM_STORM_number_unique_fix_times:description = "Number of unique fix times that were read, after removing duplicates and corrected messages. This parameter provides the total number of unique vortex fixes for the given storm."

VDM_STORM_number_unique_fix_times:units = "count"

char VDM_STORM_stormname(nstorms, VDM_STORM_stormname_StrLen)

VDM_STORM_stormname:units = "none"

VDM_STORM_stormname:description = "Stormname for the particular storm obtained from the Storm Table. The lowercase version of this is used to set the filename of the file containing the Vortex Data Messages for this storm. This should be identical to BT_STORM_stormname when a file containing VDMs was found for this storm."

VDM Dimension Parameters

int VDM_FIX_ivortex(nstorms, nmax_VDM_fixes)

VDM_FIX_ivortex:description = "Index indicating the vortex fix number (starts at 0 and increments by 1 for each unique vortex fix)"

VDM_FIX_ivortex:units = "none"

char VDM_FIX_fix_yyyymmddhhmm(nstorms, nmax_VDM_fixes, VDM_FIX_fix_yyyymmddhhmm_StrLen)

VDM_FIX_fix_yyyymmddhhmm:units = ""

VDM_FIX_fix_yyyymmddhhmm:description = "Date/time (string representation) corresponding to the fix time for each unique vortex fix (the time that the aircraft crossed the storm center), yyyymmddhhmm: yyyy = 4-digit year, mm = 2-digit month, dd = 2-digit day, hh = 2-digit hour, mm = 2-digit minute."

int VDM_FIX_fix_timeoffset(nstorms, nmax_VDM_fixes)

VDM_FIX_fix_timeoffset:description = "Time corresponding to the fix time for each unique vortex fix (the time that the aircraft crossed the storm center), as measured as an offset from a base time"

VDM_FIX_fix_timeoffset:units = "seconds since 1970-01-01:00:00:00 UTC"

VDM Data Dimensions

char VDM_FIX_transmission_yyyymmddhhmm(nstorms, nmax_VDM_fixes, VDM_FIX_transmission_yyyymmddhhmm_StrLen)

VDM_FIX_transmission_yyyymmddhhmm:units = ""

VDM_FIX_transmission_yyyymmddhhmm:description = "Date/time (string representation) corresponding to the time of transmission for each unique vortex fix, yyyymmddhhmm: yyyy = 4-digit year, mm = 2-digit month, dd = 2-digit day, hh = 2-digit hour, mm = 2-digit minute."

int VDM_FIX_transmission_timeoffset(nstorms, nmax_VDM_fixes)

VDM_FIX_transmission_timeoffset:description = "Time corresponding to the time of transmission for each unique vortex fix, as measured as an offset from a base time"

VDM_FIX_transmission_timeoffset:units = "seconds since 1970-01-01:00:00:00 UTC"

char VDM_FIX_fix_identifier(nstorms, nmax_VDM_fixes, VDM_FIX_fix_identifier_StrLen)

VDM_FIX_fix_identifier:units = "none"

VDM_FIX_fix_identifier:description = "A full string identifier used to uniquely identify each VDM. The identifier is composed of the stormyear, the basin code, the stormnumber, VDM_FIX_fix_yyyymmddhhmm, VDM_FIX_transmission_yyyymmddhhmm, VDM_FIX_header, VDM_FIX_office, VDM_FIX_transmission_day, VDM_FIX_transmission_time, VDM_FIX_aircraft_identification, VDM_FIX_wx_mission_identification, VDM_FIX_stormname_identification, VDM_FIX_ob_number, VDM_FIX_ob_number, VDM_FIX_corrected_string, VDM_FIX_corrected_number. An example is: 2008 AL17 200810171341 URNT12 KMIA 171350 AF980 0817A PALOMA OB 14 CCA."

char VDM_FIX_month(nstorms, nmax_VDM_fixes, VDM_FIX_month_StrLen)

VDM_FIX_month:units = "none"

VDM_FIX_month:description = "2-digit month (string) of each unique vortex fix, determined by a space/time match between the lat/lon/day of the VDM fix time and the Best Track b-deck file for this storm. 'XX' = unknown."

int VDM_FIX_corrected(nstorms, nmax_VDM_fixes)

VDM_FIX_corrected:units = "none"

VDM_FIX_corrected:description = "A boolean indicating whether the VDM was subsequently corrected with a revised VDM. This parameter is set based on whether a correction code is present in either the MISSION or FLIGHT lines of each unique VDM. Note: A corrected message is normally issued when the maximum wind speed on the outbound leg exceeds the maximum wind speed of the inbound leg, or an error is found."

char VDM_FIX_corrected_string(nstorms, nmax_VDM_fixes, VDM_FIX_corrected_string_StrLen)

VDM_FIX_corrected_string:units = "none"

VDM_FIX_corrected_string:description = "String that contains the correction code (e.g. COR, CCA, CCB, or CORRECTED), parsed from either the MISSION or FLIGHT lines of each unique VDM. This parameter will have an empty string if the VDM is not a corrected message. "

char VDM_FIX_corrected_number(nstorms, nmax_VDM_fixes, VDM_FIX_corrected_number_StrLen)

VDM_FIX_corrected_number:units = "none"

VDM_FIX_corrected_number:description = "Correction number (string) included in either the MISSION or FLIGHT lines of each unique VDM (e.g. 'CCA'). This parameter will have an empty string if the VDM is not a corrected message. "

char VDM_FIX_raw_mission(nstorms, nmax_VDM_fixes, VDM_FIX_raw_mission_StrLen)

VDM_FIX_raw_mission:units = "none"

VDM_FIX_raw_mission:description = "Raw MISSION line from each unique VDM (e.g. 'URNT12 KMIA 131457'). This line contains the VDM header (e.g. 'URNT12' or 'URPN12'), the issuing office (e.g. KMIA, KWBC, or KNHC), the 2-digit transmission day, the 4-digit time of transmission, and the correction code (if the VDM is a corrected message). "

char VDM_FIX_header(nstorms, nmax_VDM_fixes, VDM_FIX_header_StrLen)

VDM_FIX_header:units = "none"

VDM_FIX_header:description = "VDM header for each unique VDM (e.g. 'URNT12' or 'URPN12'), read from the MISSION line of each VDM"

char VDM_FIX_office(nstorms, nmax_VDM_fixes, VDM_FIX_office_StrLen)

VDM_FIX_office:units = "none"

VDM_FIX_office:description = "Issuing office for each unique VDM (e.g. 'KMIA', 'KWBC', or 'KNHC'), read from the MISSION line of the VDM"

char VDM_FIX_transmission_day(nstorms, nmax_VDM_fixes, VDM_FIX_transmission_day_StrLen)

VDM_FIX_transmission_day:units = "none"

VDM_FIX_transmission_day:description = "2-digit day (UTC) that each unique vortex fix was transmitted from the plane, read from the MISSION line of the VDM"

char VDM_FIX_transmission_time(nstorms, nmax_VDM_fixes, VDM_FIX_transmission_time_StrLen)

VDM_FIX_transmission_time:units = "none"

VDM_FIX_transmission_time:description = "4-digit time (UTC) that each unique vortex fix was transmitted from the plane, read from the MISSION line of the VDM"

int VDM_FIX_mission_corrected(nstorms, nmax_VDM_fixes)

VDM_FIX_mission_corrected:units = "none"

VDM_FIX_mission_corrected:description = "Boolean indicating whether the VDM was subsequently corrected with a revised VDM. This parameter is set based on whether a correction code is present in the MISSION line of each unique VDM. Note: A corrected message is normally issued when the maximum wind speed on the outbound leg exceeds the maximum wind speed of the inbound leg, or an error is found."

char VDM_FIX_mission_corrected_string(nstorms, nmax_VDM_fixes, VDM_FIX_mission_corrected_string_StrLen)

VDM_FIX_mission_corrected_string:units = "none"

VDM_FIX_mission_corrected_string:description = "String that contains the correction code (e.g. 'CCA'), parsed from the MISSION line of each unique VDM. This parameter will have an empty string if the VDM is not a corrected message. "

char VDM_FIX_mission_corrected_number(nstorms, nmax_VDM_fixes, VDM_FIX_mission_corrected_number_StrLen)
VDM_FIX_mission_corrected_number:units = "none"
VDM_FIX_mission_corrected_number:description = "Correction number (string), translated from the MISSION line of each unique VDM. This parameter will have an empty string if the VDM is not a corrected message. "

char VDM_FIX_raw_flight(nstorms, nmax_VDM_fixes, VDM_FIX_raw_flight_StrLen)
VDM_FIX_raw_flight:units = "none"
VDM_FIX_raw_flight:description = "Raw (original) FLIGHT line from each unique VDM. This line contains the aircraft identification number (e.g. AF968), the weather mission identifier (e.g. 0302A), literal stormname (or 'CYCLONE' or 'INVEST'), the sequential observation number, and a correction code that indicates whether this VDM is a corrected vortex message. This is typically of the form: 'AF968 0302A CYCLONE OB 17 KMIA'. After 1997, this information is included in the PAPA line: 'P. AF304 1316A NOEL OB 07 CCA')."

char VDM_FIX_aircraft_identification(nstorms, nmax_VDM_fixes, VDM_FIX_aircraft_identification_StrLen)
VDM_FIX_aircraft_identification:units = "none"
VDM_FIX_aircraft_identification:description = "Aircraft identification number (e.g. AF968) read from the FLIGHT line from each unique VDM"

char VDM_FIX_wx_mission_identification(nstorms, nmax_VDM_fixes, VDM_FIX_wx_mission_identification_StrLen)
VDM_FIX_wx_mission_identification:units = "none"
VDM_FIX_wx_mission_identification:description = "Weather mission identifier (e.g. 0302A) read from the FLIGHT line from each unique VDM (*)"

char VDM_FIX_stormname_identification(nstorms, nmax_VDM_fixes, VDM_FIX_stormname_identification_StrLen)
VDM_FIX_stormname_identification:units = "none"
VDM_FIX_stormname_identification:description = "Literal stormname (or 'CYCLONE' or 'INVEST') given in the FLIGHT line from each unique VDM"

char VDM_FIX_ob_number(nstorms, nmax_VDM_fixes, VDM_FIX_ob_number_StrLen)
VDM_FIX_ob_number:units = "none"
VDM_FIX_ob_number:description = "2-digit number (string) determined by the sequential order in which the observation is transmitted from the aircraft, parsed from the FLIGHT line from each unique VDM"

char VDM_FIX_basin_identifier(nstorms, nmax_VDM_fixes, VDM_FIX_basin_identifier_StrLen)

VDM_FIX_basin_identifier:units = "none"

VDM_FIX_basin_identifier:description = "2-letter ATCF basin identifier (AL = Atlantic, EP = Eastern Pacific, CP = Central Pacific, WP = Western Pacific), translated from the 1-letter basin code in the weather mission identifier code of the FLIGHT line from each unique VDM"

char VDM_FIX_designated_cyclone_number(nstorms, nmax_VDM_fixes, VDM_FIX_designated_cyclone_number_StrLen)

VDM_FIX_designated_cyclone_number:units = "none"

VDM_FIX_designated_cyclone_number:description = "2-letter ATCF storm number contained in the weather mission identifer code of the FLIGHT line from each unique VDM"

int VDM_FIX_flight_corrected(nstorms, nmax_VDM_fixes)

VDM_FIX_flight_corrected:units = "none"

VDM_FIX_flight_corrected:description = "Boolean indicating whether the VDM was subsequently corrected with a revised VDM. This parameter is set based on whether a correction code is present in the FLIGHT line of each unique VDM. Note: A corrected message is normally issued when the maximum wind speed on the outbound leg exceeds the maximum wind speed of the inbound leg, or an error is found."

char VDM_FIX_flight_corrected_string(nstorms, nmax_VDM_fixes, VDM_FIX_flight_corrected_string_StrLen)

VDM_FIX_flight_corrected_string:units = "none"

VDM_FIX_flight_corrected_string:description = "Correction code included in the FLIGHT line of each unique VDM (e.g. 'CCA'). This parameter will have an empty string if the VDM is not a corrected message. "

char VDM_FIX_flight_corrected_number(nstorms, nmax_VDM_fixes, VDM_FIX_flight_corrected_number_StrLen)

VDM_FIX_flight_corrected_number:units = "none"

VDM_FIX_flight_corrected_number:description = "Correction number (string) included in the FLIGHT line of each unique VDM (e.g. 'CCA'). This parameter will have an empty string if the VDM is not a corrected message. "

char VDM_FIX_stormid_identification(nstorms, nmax_VDM_fixes, VDM_FIX_stormid_identification_StrLen)

VDM_FIX_stormid_identification:units = "none"

VDM_FIX_stormid_identification:description = "Stormid (e.g. AL132013) read from the STORM line from each unique VDM"

char VDM_FIX_raw_storm(nstorms, nmax_VDM_fixes, VDM_FIX_raw_storm_StrLen)

VDM_FIX_raw_storm:units = "none"

VDM_FIX_raw_storm:description = "Raw STORM line from each unique VDM. This is typically in the format 'VORTEX DATA MESSAGE', 'DETAILED VORTEX DATA MESSAGE', or 'ABBREVIATED VORTEX DATA MESSAGE'. Starting in 2007, the stormid was included as well: 'VORTEX DATA MESSAGE AL162007'."

char VDM_FIX_raw_alpha(nstorms, nmax_VDM_fixes, VDM_FIX_raw_alpha_StrLen)

VDM_FIX_raw_alpha:units = "none"

VDM_FIX_raw_alpha:description = "Raw ALPHA line from each unique VDM, containing the date and time (UTC) of the flight level center fix. This is typically in the format '02/14:15:50Z' or '09/0846Z'. Note: if the flight level center cannot be fixed and the surface center is visible, the time of the surface center fix is given."

char VDM_FIX_day(nstorms, nmax_VDM_fixes, VDM_FIX_day_StrLen)

VDM_FIX_day:units = "none"

VDM_FIX_day:description = "2-digit day of the vortex fix (UTC), read from the ALPHA line of each unique VDM"

char VDM_FIX_time(nstorms, nmax_VDM_fixes, VDM_FIX_time_StrLen)

VDM_FIX_time:units = "none"

VDM_FIX_time:description = "4-digit time of the vortex fix (UTC), read from the ALPHA line of each unique VDM"

char VDM_FIX_raw_bravo_lat(nstorms, nmax_VDM_fixes, VDM_FIX_raw_bravo_lat_StrLen)

VDM_FIX_raw_bravo_lat:units = "none"

VDM_FIX_raw_bravo_lat:description = "Raw BRAVO_LAT line from each unique VDM, which contains the latitude of the flight level center fix associated with the time given in item ALPHA. This is typically of the format: '23 DEG 48 MIN N'. Some malformed lines could be: 'B. 19DEG 25MIN N ' and the hemisphere designator might be missing. Note: if the surface center is fixable, the bearing and range from the flight level center is given in the REMARKS section (e.g. SFC CENTER 270/15 nm), if the centers are separated by over 5 nm)."

float VDM_FIX_latitude(nstorms, nmax_VDM_fixes)

VDM_FIX_latitude:description = "Latitude of the flight level center fix, converted to floating point, from the BRAVO_LAT line of each unique VDM"

VDM_FIX_latitude:units = "degrees_north"

char VDM_FIX_raw_bravo_lon(nstorms, nmax_VDM_fixes, VDM_FIX_raw_bravo_lon_StrLen)

VDM_FIX_raw_bravo_lon:units = "none"

VDM_FIX_raw_bravo_lon:description = "Raw BRAVO_LON line from each unique VDM, which contains the longitude of the flight level center fix. This is typically of the format: '101 DEG 22 MIN W'. Some malformed lines could be: '101DEG 22MIN W ' and the hemisphere designator might be missing. Note: if the surface center is fixable, the bearing and range from the light level center is given in the REMARKS section (e.g. SFC CENTER 270/15 nm), if the centers are separated by over 5 nm)."

float VDM_FIX_longitude(nstorms, nmax_VDM_fixes)
VDM_FIX_longitude:description = "Longitude of the flight level center fix, converted to floating point, from the BRAVO_LON line of each unique VDM"
VDM_FIX_longitude:units = "degrees_east"

char VDM_FIX_raw_charlie(nstorms, nmax_VDM_fixes, VDM_FIX_raw_charlie_StrLen)
VDM_FIX_raw_charlie:units = "none"
VDM_FIX_raw_charlie:description = "Raw CHARLIE line from each unique VDM, which contains the standard atmospheric surface and the minimum height of that surface in the center. This is typically of the format: '925 MB 782 M'. Some malformed lines could be: '925MB 782M' or 'NA' or 'MB NOBS M'. It is possible to report 'NA' if below 1500 ft altitude, or more than 1500 feet away from a flight surface - in that case, it's possible that a minimum height may be reported. Currently, the code reports an 'NA' standard pressure level as missing, but will record the minimum height if it is given."

int VDM_FIX_standard_pressure_level(nstorms, nmax_VDM_fixes)
VDM_FIX_standard_pressure_level:description = "Standard atmospheric pressure surface of the flight level center fix (e.g. '925 MB'), translated from the CHARLIE line from each unique VDM"
VDM_FIX_standard_pressure_level:units = "millibar"

int VDM_FIX_minimum_height(nstorms, nmax_VDM_fixes)
VDM_FIX_minimum_height:description = "Minimum height of the standard pressure surface observed inside the center, for the flight level center fix (e.g. '782M'), translated from the CHARLIE line from each unique VDM. Note: if at 1,500 ft or below, or not within 1,500 ft of a standard pressure surface, 'NA' is given."
VDM_FIX_minimum_height:units = "meter"

char VDM_FIX_raw_delta(nstorms, nmax_VDM_fixes, VDM_FIX_raw_delta_StrLen)
VDM_FIX_raw_delta:units = "none"
VDM_FIX_raw_delta:description = "Raw DELTA line from each unique VDM, which contains the maximum surface wind speed observed during the inbound leg associated with each center fix. This is typically of the format: '75 KT'. Some malformed lines could be: '100KT' or 'NA' or 'N/AKT'. Note: When SFMR surface wind data are unavailable, the surface wind is estimated visually."

int VDM_FIX_inbound_maximum_surface_wind_speed(nstorms, nmax_VDM_fixes)
VDM_FIX_inbound_maximum_surface_wind_speed:description = "Maximum surface windspeed observed during the inbound leg associated with this fix, read from the DELTA line from each unique VDM"
VDM_FIX_inbound_maximum_surface_wind_speed:units = "knot"

char VDM_FIX_raw_echo(nstorms, nmax_VDM_fixes, VDM_FIX_raw_echo_StrLen)
VDM_FIX_raw_echo:units = "none"

VDM_FIX_raw_echo:description = "Raw ECHO line from each unique VDM, which contains the bearing and range of the maximum surface wind speed observed during the inbound leg (item DELTA) associated with each unique center fix. This is ypically of the format: '008 DEG 009 NM'. Some malformed lines could be: ' DEG NM' or 'NA' or 'N/A' or 'DEG NOBNM'."

int VDM_FIX_inbound_maximum_surface_wind_bearing(nstorms, nmax_VDM_fixes)

VDM_FIX_inbound_maximum_surface_wind_bearing:description = "Bearing (from the flight level fix center coordinates given in BRAVO) to the location of the maximum surface wind speed observed during the inbound leg associated with this fix, translated from the ECHO line from each unique VDM"

VDM_FIX_inbound_maximum_surface_wind_bearing:units = "degrees"

float VDM_FIX_inbound_maximum_surface_wind_range(nstorms, nmax_VDM_fixes)

VDM_FIX_inbound_maximum_surface_wind_range:description = "Range (relative to the flight level fix center coordinates given in BRAVO) to the location of the maximum surface wind speed observed during the inbound leg associated with this fix, translated from the ECHO line from each unique VDM"

VDM_FIX_inbound_maximum_surface_wind_range:units = "nautical_mile"

char VDM_FIX_raw_foxtrot(nstorms, nmax_VDM_fixes, VDM_FIX_raw_foxtrot_StrLen)

VDM_FIX_raw_foxtrot:units = "none"

VDM_FIX_raw_foxtrot:description = "Raw FOXTROT line from each unique VDM, which contains the maximum flight level wind speed and wind direction measured during the inbound leg associated with this fix. This is typically of the format '008 DEG 135 KT'. Some malformed lines could be: ' DEG NM', or 'NA', or 'N/A', or 'DEG NOBNM'. Note: If a significant secondary maximum wind is observed, it is reported in the REMARKS section. All winds reported should be 10-s averages."

int VDM_FIX_inbound_maximum_flight_level_wind_direction(nstorms, nmax_VDM_fixes)

VDM_FIX_inbound_maximum_flight_level_wind_direction:description = "Direction of the wind at the location of the maximum flight level wind speed measured during the inbound leg, translated from the FOXTROT line from each unique VDM"

VDM_FIX_inbound_maximum_flight_level_wind_direction:units = "degrees"

int VDM_FIX_inbound_maximum_flight_level_wind_speed(nstorms, nmax_VDM_fixes)

VDM_FIX_inbound_maximum_flight_level_wind_speed:description = "Maximum flight level wind speed measured during the inbound leg, translated from the FOXTROT line from each unique VDM"

VDM_FIX_inbound_maximum_flight_level_wind_speed:units = "knot"

char VDM_FIX_raw_golf(nstorms, nmax_VDM_fixes, VDM_FIX_raw_golf_StrLen)

VDM_FIX_raw_golf:units = "none"

VDM_FIX_raw_golf:description = "Raw GOLF line from each unique VDM, which contains the bearing and range of the maximum flight level wind speed measured during the inbound leg (item FOXTROT) of the fix. This is typically of the format '008 DEG 009 NM' Some malformed lines could be: ' DEG NM', or 'NA', or 'N/A', or 'DEG NOBNM'."

int VDM_FIX_inbound_maximum_flight_level_wind_bearing(nstorms, nmax_VDM_fixes)

VDM_FIX_inbound_maximum_flight_level_wind_bearing:description = "Bearing (from the flight level fix center coordinates given in BRAVO) to the location of the maximum flight level wind speed during the inbound leg of the fix, translated from the GOLF line from each unique VDM"

VDM_FIX_inbound_maximum_flight_level_wind_bearing:units = "degrees"

float VDM_FIX_inbound_maximum_flight_level_wind_range(nstorms, nmax_VDM_fixes)

VDM_FIX_inbound_maximum_flight_level_wind_range:description = "Range (from the flight level fix center coordinates given in BRAVO) to the location of the maximum flight level wind speed measured during the inbound leg of the fix, translated from the GOLF line from each unique VDM"

VDM_FIX_inbound_maximum_flight_level_wind_range:units = "nautical_mile"

char VDM_FIX_raw_hotel(nstorms, nmax_VDM_fixes, VDM_FIX_raw_hotel_StrLen)

VDM_FIX_raw_hotel:units = "none"

VDM_FIX_raw_hotel:description = "Raw HOTEL line from each unique VDM, which contains the minimum sea level pressure (SLP) to the nearest millibar, observed at the coordinates reported in item BRAVO. This is typically of the format: '993 MB' or '993 MB EXTRAPOLATED' or 'EXTRAP 993 MB'. Some malformed lines could be: 'EXTRAPOLATED/1011MB', or 'MB', or 'NA', or 'N/A MB', etc. SLP is prefaced with 'EXTRAP' (extrapolated) when the data are not derived from dropsonde, or when the SLP is exrapolated from a dropsonde that terminated early. The difference is clarified in the REMARKS section (e.g. 'SLP EXTRAPOLATED FROM BELOW 1500 FEET/850 hPA/DROPSONDE')."

int VDM_FIX_minimum_sea_level_pressure(nstorms, nmax_VDM_fixes)

VDM_FIX_minimum_sea_level_pressure:description = "Minimum sea level pressure of the fix, translated from the HOTEL line from each unique VDM"

VDM_FIX_minimum_sea_level_pressure:units = "millibar"

int VDM_FIX_minimum_sea_level_pressure_extrapolated(nstorms, nmax_VDM_fixes)

VDM_FIX_minimum_sea_level_pressure_extrapolated:units = "none"

VDM_FIX_minimum_sea_level_pressure_extrapolated:description = "Boolean that indicates if the minimum sea level pressure was extrapolated, either from the plane or from a dropsonde that terminated early (True), or was measured from a dropsonde (False). Translated from the HOTEL line from each unique VDM."

int VDM_FIX_minimum_sea_level_pressure_from_drop(nstorms, nmax_VDM_fixes)

VDM_FIX_minimum_sea_level_pressure_from_drop:units = "none"

VDM_FIX_minimum_sea_level_pressure_from_drop:description = "Boolean that indicates if the minimum sea level pressure was measured from a dropsonde (True), or was extrapolated from the plane or from a dropsonde that terminated early (False). Translated from the HOTEL line from each unique VDM."

char VDM_FIX_raw_india(nstorms, nmax_VDM_fixes, VDM_FIX_raw_india_StrLen)
VDM_FIX_raw_india:units = "none"
VDM_FIX_raw_india:description = "Raw INDIA line from each unique VDM, which contains the maximum temperature at flight level taken just outside of the central region (i.e. just outside the eyewall or just beyond the maximum wind band), along with the height at that location. This is typically of the format: '23 C/ 1599 M'. Some malformed lines could be: ' C/ M', or 'NA C/ M', or 'NOBS'. This temperature may not be the highest value recorded on the inbound leg, but is representative of the environmental temperature just outside of the central region of the storm."

float VDM_FIX_maximum_flight_level_temperature_outside(nstorms, nmax_VDM_fixes)
VDM_FIX_maximum_flight_level_temperature_outside:description = "Maximum temperature measured at flight level outside of the central region, translated from the INDIA line from each unique VDM"
VDM_FIX_maximum_flight_level_temperature_outside:units = "celsius"

int VDM_FIX_maximum_flight_level_temperature_outside_height(nstorms, nmax_VDM_fixes)
VDM_FIX_maximum_flight_level_temperature_outside_height:description = "Pressure altitude taken at the same location as the maximum temperature measured at flight level taken just outside of the central region, translated from the INDIA line from each unique VDM"
VDM_FIX_maximum_flight_level_temperature_outside_height:units = "meter"

char VDM_FIX_raw_juliet(nstorms, nmax_VDM_fixes, VDM_FIX_raw_juliet_StrLen)
VDM_FIX_raw_juliet:units = "none"
VDM_FIX_raw_juliet:description = "Raw JULIET line from each unique VDM, which contains the maximum temperature measured at flight level observed within 5 nm of the center fix coordinates, along with the pressure altitude at that location. This is typically of the format: '23 C/ 1599 M'. Some malformed lines could be: ' C/ M', or 'NA C/ M', or 'NOBS'. If a higher temperature is observed at a location more than 5 nm away from the flight level center (item BRAVO), it is reported in the REMARKS section, including bearing and distance from the flight level center."

float VDM_FIX_maximum_flight_level_temperature_inside(nstorms, nmax_VDM_fixes)
VDM_FIX_maximum_flight_level_temperature_inside:description = "Maximum temperature measured at flight level inside the central region, translated from the JULIET line from each unique VDM"
VDM_FIX_maximum_flight_level_temperature_inside:units = "celsius"

int VDM_FIX_maximum_flight_level_temperature_inside_height(nstorms, nmax_VDM_fixes)

VDM_FIX_maximum_flight_level_temperature_inside_height:description = "Pressure altitude at the same location as the maximum temperature reported in item JULIET, translated from the JULIET line from each unique VDM"

VDM_FIX_maximum_flight_level_temperature_inside_height:units = "meter"

char VDM_FIX_raw_kilo(nstorms, nmax_VDM_fixes, VDM_FIX_raw_kilo_StrLen)

VDM_FIX_raw_kilo:units = "none"

VDM_FIX_raw_kilo:description = "Raw KILO line from each unique VDM, which contains the flight level dew point temperature and sea surrface temperature measured at the same location as the maximum flight level temperature reported in item JULIET. This is typically of the format: '23 C/ 27 C'. Some malformed lines could be: ' C/ C', or '23.5 C/ NA', or '23 C/ NOBS', '// C/ NA'."

float VDM_FIX_flight_level_dewpoint_temperature(nstorms, nmax_VDM_fixes)

VDM_FIX_flight_level_dewpoint_temperature:description = "Dew point temperature at the location of the maximum flight level temperature (item JULIET), translated from the KILO line from each unique VDM"

VDM_FIX_flight_level_dewpoint_temperature:units = "celsius"

float VDM_FIX_sea_surface_temperature(nstorms, nmax_VDM_fixes)

VDM_FIX_sea_surface_temperature:description = "Sea surface temperature below the location of the maximum flight level temperature (item JULIET), translated from the KILO line from each unique VDM. Note: this item is not commonly reported."

VDM_FIX_sea_surface_temperature:units = "celsius"

char VDM_FIX_raw_lima(nstorms, nmax_VDM_fixes, VDM_FIX_raw_lima_StrLen)

VDM_FIX_raw_lima:units = "none"

VDM_FIX_raw_lima:description = "Raw LIMA line from each unique VDM, which contains information about the character of the eye (if present). This is only reported if at least 50 percent of the center has an eyewall, otherwise 'NA'. This is typically free form: 'OPEN S-W', 'CLOSED', 'RAGGED WEAK', 'POORLY DEFINED'. Malformed lines could be 'NA', 'NEG EYE', 'N/A', 'NONE', 'NEG'. A 'CLOSED WALL' is reported if the center has 100 percent coverage with no eyewall weakness. 'OPEN XX' is reported if the center has 50 percent or more but less than 100 percent coverage (the direction of the weakness is stated). 'SPIRAL BAND' is reported with the best approximation of the shape and diameter of the inner core."

char VDM_FIX_eyewall_completeness(nstorms, nmax_VDM_fixes, VDM_FIX_eyewall_completeness_StrLen)

VDM_FIX_eyewall_completeness:units = "none"

VDM_FIX_eyewall_completeness:description = "Descriptor summarizing the completeness of the eyewall ring, as subjectively determined by the flight meterologist by looking at the plane's forward-looking nose radar, translated from the LIMA line from each unique VDM. Valid translated values include: 'CLOSED', 'OPEN', or missing (no eye recorded).

The following descriptors are mapped onto 'OPEN': 'OPENING', 'BREAKS', 'SEMI-CIRCLE'. The 'CLOSE' descriptor is mapped onto 'CLOSED'. If 'BANDING' is reported, this is sent to the 'REMARKS' portion of the codeset and translated into a separate parameter."

char VDM_FIX_eyewall_definition(nstorms, nmax_VDM_fixes, VDM_FIX_eyewall_definition_StrLen)

VDM_FIX_eyewall_definition:units = "none"

VDM_FIX_eyewall_definition:description = "Descriptor summarizing how well defined the eyewall ring is, as subjectively determined by the flight meteorologist by looking at the plane's forward-looking nose radar, translated from the LIMA line from each unique VDM. Valid values include: 'WEAK', 'RAGGED', 'GOOD', and 'FAIR'. The descriptor 'WELL' is mapped onto 'GOOD'. 'RAGGED' will also be mapped onto 'POOR' if the REMARKS section contains a phrase with both the word 'RAGGED' and 'EYE'."

char VDM_FIX_eyewall_weakness_direction(nstorms, nmax_VDM_fixes, VDM_FIX_eyewall_weakness_direction_StrLen)

VDM_FIX_eyewall_weakness_direction:units = "none"

VDM_FIX_eyewall_weakness_direction:description = "Direction (e.g. 'N', 'NE', 'NNE', 'ENE', etc.) of any noted weakness in the eyewall ring, as subjectively determined by the flight meterologist by looking at the plane's forward-looking nose radar, translated form the LIMA line from each unique VDM. Rarely, multiple directions will be given separated by 'AND'. The code does not currently distinguish these additional directions."

char VDM_FIX_raw_mike(nstorms, nmax_VDM_fixes, VDM_FIX_raw_mike_StrLen)

VDM_FIX_raw_mike:units = "none"

VDM_FIX_raw_mike:description = "Raw MIKE line from each unique VDM, which contains information about the shape of the eye, the orientation of the eye (if elliptical), the number of eyewalls present, and their diameters, This is typically of the format: 'E270/30/20', 'E 09/25/15', 'C12', 'C 12', 'C-06', 'CO8-14', 'CO 11 22', 'CO 14-35', or just 'CO' - if not available 'N/A' or 'NA'. Only reported if item LIMA is reported."

char VDM_FIX_eye_shape(nstorms, nmax_VDM_fixes, VDM_FIX_eye_shape_StrLen)

VDM_FIX_eye_shape:units = "none"

VDM_FIX_eye_shape:description = "Shape of eye ('elliptical', 'circular', or 'concentric'), translated from the MIKE line from each unique VDM"

float VDM_FIX_eye_orientation(nstorms, nmax_VDM_fixes)

VDM_FIX_eye_orientation:units = "none"

float VDM_FIX_eye_diameter_major(nstorms, nmax_VDM_fixes)

VDM_FIX_eye_diameter_major:description = "If the eye_shape is elliptical, this gives the length of the major axis. Transalted from the MIKE line from each unique VDM."

VDM_FIX_eye_diameter_major:units = "nautical_mile"

float VDM_FIX_eye_diameter_minor(nstorms, nmax_VDM_fixes)

VDM_FIX_eye_diameter_minor:description = "If the eye_shape is elliptical, this gives the length of the minor axis. Translated from the MIKE line from each unique VDM."

VDM_FIX_eye_diameter_minor:units = "nautical_mile"

float VDM_FIX_eye_diameter_primary(nstorms, nmax_VDM_fixes)

VDM_FIX_eye_diameter_primary:description = "Diameter of the primary (normally the inner or first) eye, if present. Translated from the MIKE line from each unique VDM."

VDM_FIX_eye_diameter_primary:units = "nautical_mile"

float VDM_FIX_eye_diameter_secondary(nstorms, nmax_VDM_fixes)

VDM_FIX_eye_diameter_secondary:description = "Diameter of the secondary (first outer) eye, if present. Translated from the MIKE line from each unique VDM."

VDM_FIX_eye_diameter_secondary:units = "nautical_mile"

float VDM_FIX_eye_diameter_tertiary(nstorms, nmax_VDM_fixes)

VDM_FIX_eye_diameter_tertiary:description = "Diameter of the tertiary (second outer) eye, if present (very rarely observed). Translated from the MIKE line from each unique VDM."

VDM_FIX_eye_diameter_tertiary:units = "nautical_mile"

int VDM_FIX_number_eyewalls(nstorms, nmax_VDM_fixes)

VDM_FIX_number_eyewalls:description = "Number of eyewalls reported, translated from the MIKE line from each unique VDM"

VDM_FIX_number_eyewalls:units = "count"

int VDM_FIX_light_and_variable_wind_center(nstorms, nmax_VDM_fixes)

VDM_FIX_light_and_parameter_wind_center:units = "none"

VDM_FIX_light_and_parameter_wind_center:description = "Boolean indicating if a 'light and variable wind center' was reported (using various abbrevations). This is usually an area of light and variable winds sometimes reported when an eye is not reported, and usually occurs in the disorganized stage). This may be reported in the MIKE section, but is more commonly reported in the remarks section."

float VDM_FIX_light_and_parameter_wind_center_diameter(nstorms, nmax_VDM_fixes)

VDM_FIX_light_and_variable_wind_center_diameter:description = "Diameter of the 'light and variable wind center', if present. See description for 'VDM_FIX_light_and_parameter_wind_center'."

VDM_FIX_light_and_parameter_wind_center_diameter:units = "nautical_mile"

char VDM_FIX_raw_november(nstorms, nmax_VDM_fixes, VDM_FIX_raw_november_StrLen)

VDM_FIX_raw_november:units = "none"

VDM_FIX_raw_november:description = "Raw NOVEMBER line from each unique VDM, which contains information about how the center fix was determined and at what level the fix was made. This is typically of the form: '12345/07'. '1' is always reported. '2' is reported if radar indicates curvature or banding consistent with the fix location. '3' is reported if recorded or observed winds indicate a closed center. '4' is reported if the fix pressure is lower than all reported pressures on the inbound leg. '5' is reported if the fix temperature is at least as high or higher than any of the temperatures reported on the inbound leg. '0' is reported if the fix is made solely on surface winds. '0' and the flight level code are reported if the centers are within 5 nm of each other. Note: Before 1999, this was in the OSCAR line. From 1999 onward, it has been included in the NOVEMBER line."

char VDM_FIX_determination_string(nstorms, nmax_VDM_fixes, VDM_FIX_determination_string_StrLen)

VDM_FIX_determination_string:units = "none"

VDM_FIX_determination_string:description = "String that contains the fix determination method(s) (e.g. '12345'), parsed from the NOVEMBER (or OSCAR) line of each unique VDM"

char VDM_FIX_determination_decoded(nstorms, nmax_VDM_fixes, VDM_FIX_determination_decoded_StrLen)

VDM_FIX_determination_decoded:units = "none"

VDM_FIX_determination_decoded:description = "Human-readable decoding of the fix determination method(s), translated from the NOVEMBER (or OSCAR) line of each unique VDM"

int VDM_FIX_by_penetration(nstorms, nmax_VDM_fixes)

VDM_FIX_by_penetration:units = "none"

VDM_FIX_by_penetration:description = "Boolean to indicate whether the fix was made by penetration of the storm center, translated from the NOVEMBER (or OSCAR) line of each unique VDM"

int VDM_FIX_by_radar(nstorms, nmax_VDM_fixes)

VDM_FIX_by_radar:units = "none"

VDM_FIX_by_radar:description = "Boolean to indicate whether radar information was used to help fix the center of the storm, translated from the NOVEMBER (or OSCAR) line of each unique VDM"

int VDM_FIX_by_wind(nstorms, nmax_VDM_fixes)

VDM_FIX_by_wind:units = "none"

VDM_FIX_by_wind:description = "Boolean to indicate whether wind information was used to help fix the center of the storm, translated from the NOVEMBER (or OSCAR) line of each unique VDM"

int VDM_FIX_by_pressure(nstorms, nmax_VDM_fixes)

VDM_FIX_by_pressure:units = "none"

VDM_FIX_by_pressure:description = "Boolean to indicate whether pressure information was used to help fix the center of the storm, translated from the NOVEMBER (or OSCAR) line of each unique VDM"

int VDM_FIX_by_temperature(nstorms, nmax_VDM_fixes)
VDM_FIX_by_temperature:units = "none"
VDM_FIX_by_temperature:description = "Boolean to indicate whether the temperature information was used to help fix the center of the storm, translated from the NOVEMBER (or OSCAR) line of each unique VDM"

char VDM_FIX_level_string(nstorms, nmax_VDM_fixes, VDM_FIX_level_string_StrLen)
VDM_FIX_level_string:units = "none"
VDM_FIX_level_string:description = "String containing the flight level(s) used to fix the center of the storm, parsed from the NOVEMBER (or OSCAR) line of each unique VDM"

char VDM_FIX_level_flight(nstorms, nmax_VDM_fixes, VDM_FIX_level_flight_StrLen)
VDM_FIX_level_flight:units = "none"
VDM_FIX_level_flight:description = "Designator (with non-integer numbers removed) that indicates the flight level(s) used to fix the center of the storm, parsed from the NOVEMBER (or OSCAR) line of each unique VDM"

char VDM_FIX_level_decoded(nstorms, nmax_VDM_fixes, VDM_FIX_level_decoded_StrLen)
VDM_FIX_level_decoded:units = "(units may vary and are included in the string)"
VDM_FIX_level_decoded:description = "Human-readable decoding of the flight level(s) used to fix the center of the storm, parsed from the NOVEMBER (or OSCAR) line of each unique VDM. Valid values include: 'surface', '1500 ft', '925 mb', '850 mb', '700 mb', '500 mb', '400 mb', '300 mb', '200 mb'. If both the surface and flight level data are used, then this can also take on the value: 'surface and <LEVEL>'."

int VDM_FIX_level_surface_center_visible_and_coincides_with_flight_level(nstorms, nmax_VDM_fixes)
VDM_FIX_level_surface_center_visible_and_coincides_with_flight_level:units = "none"

VDM_FIX_level_surface_center_visible_and_coincides_with_flight_level:description = "Boolean that indicates whether the surface wind center was visible and coincided with the flight level center, translated from the NOVEMBER (or OSCAR) line of each unique VDM"

char VDM_FIX_raw_oscar(nstorms, nmax_VDM_fixes, VDM_FIX_raw_oscar_StrLen)
VDM_FIX_raw_oscar:units = "none"
VDM_FIX_raw_oscar:description = "Raw OSCAR line from each unique VDM, which contains information about how accurate the navigational and meteorological information are in the VDM. This is typically of the format: '0.02 / 2 NM'. Navigational and meteorological accuracy are reported as the upper limit of the probable error. Meteorological accuracy is

normally reported as one-half of the diameter of the light and variable wind center. Note: Prior to 1999, this information was included in the PAPA line. From 1999 onward, it has been included in the OSCAR line."

char VDM_FIX_navigational_accuracy_string(nstorms, nmax_VDM_fixes, VDM_FIX_navigational_accuracy_string_StrLen)

VDM_FIX_navigational_accuracy_string:units = "none"

VDM_FIX_navigational_accuracy_string:description = "String containing the navigational fix accuracy information, parsed from the OSCAR (or PAPA) line from each unique VDM"

float VDM_FIX_navigational_accuracy(nstorms, nmax_VDM_fixes)

VDM_FIX_navigational_accuracy:description = "Navigational fix accuracy converted into a floating point number, translated from the OSCAR (or PAPA) line of each unique VDM"

VDM_FIX_navigational_accuracy:units = "nautical_mile"

char VDM_FIX_meteorological_accuracy_string(nstorms, nmax_VDM_fixes, VDM_FIX_meteorological_accuracy_string_StrLen)

VDM_FIX_meteorological_accuracy_string:units = "none"

VDM_FIX_meteorological_accuracy_string:description = "String containing the meteorological fix accuracy information, parsed from the OSCAR (or PAPA) line from each unique VDM"

float VDM_FIX_meteorological_accuracy(nstorms, nmax_VDM_fixes)

VDM_FIX_meteorological_accuracy:description = "Meteorological fix accuracy converted into a floating point number, translated from the OSCAR (or PAPA) line of each unique VDM"

VDM_FIX_meteorological_accuracy:units = "nautical_mile"

char VDM_FIX_raw_remark(nstorms, nmax_VDM_fixes, VDM_FIX_raw_remark_StrLen)

VDM_FIX_raw_remark:units = "none"

VDM_FIX_raw_remark:description = "Raw REMARK line(s) from each unique VDM to enhance the data reported in the regular VDM sections. This line (or lines) is freeform, often delimited with periods and line breaks, and contains various remarks made by the flight meteorologist. Some of these remarks are required, meaning that they must be reported if a particular phenomena or data are observed, while other remarks are optional. Required remarks include: (type 1) mission identifier and observation number; (type 2) the maximum flight level wind observed, time of observation, and the relative quadrant of the storm of the observed wind on the latest pass through any portion of the storm; (type 3) the maximum flight level wind observed on the outbound leg following the center fix just obtained, if it is higher than the inbound maximum reported in item FOXTROT. The time of the observation and the relative quadrant of gthe storm of the qualifying outbound maximum wind are included. If, after transmission of the vortex message but prior to the aircraft reaching the cross-leg turn point, a higher qualifying outbound wind is observed, then the vortex message will be amdended with the higher outbound wind reported. If the outbound maximum flight level wind becomes the new overall maximum flight level wind, then the two maximum flight level wind remarks are consolidated into one remark; (type 4) the method of deriving the central SLP when extrapolated; and (type 5) the bearing and range of the surface center and/or maximum flight level temperature if not within 5 nm of the flight level center. Note: Before 1999, the remarks were contained in the QUEBEC line. From 1999 onward, they are included in the lines following the PAPA line. The MISSION line is given as the first mandatory remark, however this is not included in this string since it is stored in raw_MISSION. The data are typically in the format: 'MAX FL WIND 43 KT SW QUAD 1824Z.', or 'SLP EXTRAP FROM 1500FT.' or 'MAX FL TEMP 26C 228/9NM FROM FL CNTR.' For a full description of what data are included in the REMARKS section, please see the full documentation."

char VDM_FIX_cleaned_remark(nstorms, nmax_VDM_fixes, VDM_FIX_cleaned_remark_StrLen)

VDM_FIX_cleaned_remark:units = "none"

VDM_FIX_cleaned_remark:description = "A 'cleaned' version of the remarks string, in which underscores, commas, and semicolons have been replaced by spaces, floating point numbers have been identified and 'protected' using hashtags in place of the decimal point, spaces have been inserted around any periods and slashes, and abbreviations (both standard and variants) have been expanded into their full form. These changes make it easier to parse the REMARKS line(s) into the various constituent data parameters."

int VDM_FIX_maximum_flight_level_wind(nstorms, nmax_VDM_fixes)

VDM_FIX_maximum_flight_level_wind:units = "none"

VDM_FIX_maximum_flight_level_wind:description = "Boolean that indicates whether a type 2 required remark is present in the REMARKS section, which provides the observed maximum flight level wind speed on the latest pass through any portion of the storm. This remark type includes information about the maximum flight level wind observed, the time of observation, and the relative quadrant of the storm of the observed wind on the latest pass through any portion of the storm. This parameter is translated from the REMARKS line of each unique VDM."

int VDM_FIX_maximum_flight_level_wind_speed(nstorms, nmax_VDM_fixes)

VDM_FIX_maximum_flight_level_wind_speed:description = "Maximum flight level wind speed on the latest pass through any portion of the storm (a type 2 required remark), translated from the REMARKS section from each unique VDM"

VDM_FIX_maximum_flight_level_wind_speed:units = "knot"

int VDM_FIX_maximum_flight_level_wind_bearing(nstorms, nmax_VDM_fixes)

VDM_FIX_maximum_flight_level_wind_bearing:description = "Bearing (from the flight level fix center coordinates given in BRAVO) to the location of the maximum flight level wind speed on the latest pass through any portion of the storm (a type 2 required remark), translated from the REMARKS section from each unique VDM"

VDM_FIX_maximum_flight_level_wind_bearing:units = "degrees"

float VDM_FIX_maximum_flight_level_wind_range(nstorms, nmax_VDM_fixes)

VDM_FIX_maximum_flight_level_wind_range:description = "Range (from the flight level fix center coordinates given in BRAVO) to the location of the maximum flight level wind speed on the latest pass through any portion of the storm (a type 2 required remark), translated from the REMARKS section from each unique VDM"
VDM_FIX_maximum_flight_level_wind_range:units = "nautical_mile"

char VDM_FIX_maximum_flight_level_wind_relative_quadrant(nstorms, nmax_VDM_fixes, VDM_FIX_maximum_flight_level_wind_relative_quadrant_StrLen)
VDM_FIX_maximum_flight_level_wind_relative_quadrant:units = "none"
VDM_FIX_maximum_flight_level_wind_relative_quadrant:description = "Relative quadrant of the storm of the observed maximum flight level wind speed on the latest pass through any portion of the storm (a type 2 required remark), translated from the REMARKS section from each unique VDM"

char VDM_FIX_maximum_flight_level_wind_day(nstorms, nmax_VDM_fixes, VDM_FIX_maximum_flight_level_wind_day_StrLen)
VDM_FIX_maximum_flight_level_wind_day:units = "UTC"
VDM_FIX_maximum_flight_level_wind_day:description = "2-digit day (string) of the observed maximum flight level wind speed on the latest pass through any portion of the storm (a type 2 required remark), translated from the REMARKS section from each unique VDM"

char VDM_FIX_maximum_flight_level_wind_time(nstorms, nmax_VDM_fixes, VDM_FIX_maximum_flight_level_wind_time_StrLen)
VDM_FIX_maximum_flight_level_wind_time:units = "UTC"
VDM_FIX_maximum_flight_level_wind_time:description = "4-digit day (string) of the observed maximum flight level wind speed on the latest pass through any portion of the storm (a type 2 required remark), translated from the REMARKS section from each unique VDM"

char VDM_FIX_maximum_flight_level_wind_yyyymmddhhmm(nstorms, nmax_VDM_fixes, VDM_FIX_maximum_flight_level_wind_yyyymmddhhmm_StrLen)
VDM_FIX_maximum_flight_level_wind_yyyymmddhhmm:units = "UTC"
VDM_FIX_maximum_flight_level_wind_yyyymmddhhmm:description = "Date/time (string representation, yyyymmddhhmm: yyyy = 4-digit year, mm = 2-digit month, dd = 2-digit day, hh = 2-digit hour, mm = 2-digit minute) corresponding to the observation time of the observed maximum flight level wind speed on the latest pass through any portion of the storm (a type 2 required remark), translated from the REMARKS section from each unique VDM"

int VDM_FIX_maximum_flight_level_wind_timeoffset(nstorms, nmax_VDM_fixes)
VDM_FIX_maximum_flight_level_wind_timeoffset:description = "Timeoffset of the observed maximum flight level wind speed on the latest pass through any portion of the storm (a type 2 required remark), translated from the REMARKS section from each unique VDM"

VDM_FIX_maximum_flight_level_wind_timeoffset:units = "seconds since 1970-01-01:00:00:00 UTC"

int VDM_FIX_outbound_maximum_flight_level_wind(nstorms, nmax_VDM_fixes)
VDM_FIX_outbound_maximum_flight_level_wind:units = "none"
VDM_FIX_outbound_maximum_flight_level_wind:description = "Boolean that indicates whether a type 3 required remark is present in the REMARKS section, which provides the maximum flight level wind speed observed on the outbound leg (if it is higher than the inbound maximum reported in item FOXTROT). This remark type includes information about the maximum flight level wind observed on the outbound leg, the time of observation, and the relative quadrant of the storm of the qualifying outbound wind. If, after transmission of the vortex message, but prior to the aircraft reaching the cross-leg turn point, a higher qualifying outbound wind is observed, then the vortex message will be amended with the higher outbound wind reported. This parameter is translated from the REMARKS line of each unique VDM."

int VDM_FIX_outbound_maximum_flight_level_wind_speed(nstorms, nmax_VDM_fixes)
VDM_FIX_outbound_maximum_flight_level_wind_speed:description = "Maximum flight level wind speed observed on the outbound leg (if it is higher than the inbound maximum reported in item FOXTROT) (a type 3 required remark), translated from the REMARKS section from each unique VDM"
VDM_FIX_outbound_maximum_flight_level_wind_speed:units = "knot"

int VDM_FIX_outbound_maximum_flight_level_wind_bearing(nstorms, nmax_VDM_fixes)
VDM_FIX_outbound_maximum_flight_level_wind_bearing:description = "Bearing (from the flight level fix center coordinates given in BRAVO) to the location of the maximum flight level wind speed observed on the outbound leg (if it is higher than the inbound maximum reported in item FOXTROT) (a type 3 required remark), translated from the REMARKS section from each unique VDM"
VDM_FIX_outbound_maximum_flight_level_wind_bearing:units = "degrees"

float VDM_FIX_outbound_maximum_flight_level_wind_range(nstorms, nmax_VDM_fixes)
VDM_FIX_outbound_maximum_flight_level_wind_range:description = "Range (from the flight level fix center coordinates given in BRAVO) to the location of the maximum flight level wind speed observed on the outbound leg (if it is higher than the inbound maximum reported in item FOXTROT) (a type 3 required remark), translated from the REMARKS section from each unique VDM"
VDM_FIX_outbound_maximum_flight_level_wind_range:units = "nautical_mile"

char VDM_FIX_outbound_maximum_flight_level_wind_relative_quadrant(nstorms, nmax_VDM_fixes, VDM_FIX_outbound_maximum_flight_level_wind_relative_quadrant_StrLen)

VDM_FIX_outbound_maximum_flight_level_wind_relative_quadrant:units = "none"

VDM_FIX_outbound_maximum_flight_level_wind_relative_quadrant:description = "Relative quadrant of the storm of the observed maximum flight level wind speed observed on the outbound leg (if it is higher than the inbound maximum reported in item FOXTROT) (a type 3 required remark), translated from the REMARKS section from each unique VDM"

char VDM_FIX_outbound_maximum_flight_level_wind_day(nstorms, nmax_VDM_fixes, VDM_FIX_outbound_maximum_flight_level_wind_day_StrLen)
VDM_FIX_outbound_maximum_flight_level_wind_day:units = "none"
VDM_FIX_outbound_maximum_flight_level_wind_day:description = "2-digit day (string) of the observed maximum flight level wind speed observed on the outbound leg (if it is higher than the inbound maximum reported in item FOXTROT) (a type 3 required remark), translated from the REMARKS section from each unique VDM"

char VDM_FIX_outbound_maximum_flight_level_wind_time(nstorms, nmax_VDM_fixes, VDM_FIX_outbound_maximum_flight_level_wind_time_StrLen)
VDM_FIX_outbound_maximum_flight_level_wind_time:units = "none"
VDM_FIX_outbound_maximum_flight_level_wind_time:description = "4-digit day (string) of the observed maximum flight level wind speed observed on the outbound leg (if it is higher than the inbound maximum reported in item FOXTROT) (a type 3 required remark), translated from the REMARKS section from each unique VDM"

char VDM_FIX_outbound_maximum_flight_level_wind_yyyymmddhhmm(nstorms, nmax_VDM_fixes, VDM_FIX_outbound_maximum_flight_level_wind_yyyymmddhhmm_StrLen)
VDM_FIX_outbound_maximum_flight_level_wind_yyyymmddhhmm:units = ""

VDM_FIX_outbound_maximum_flight_level_wind_yyyymmddhhmm:description = "Date/time (string representation, yyyymmddhhmm: yyyy = 4-digit year, mm = 2-digit month, dd = 2-digit day, hh = 2-digit hour, mm = 2-digit minute) corresponding to the observation time of the observed maximum flight level wind speed observed on the outbound leg (if it is higher than the inbound maximum reported in item FOXTROT) (a type 3 required remark), translated from the REMARKS section from each unique VDM"

int VDM_FIX_outbound_maximum_flight_level_wind_timeoffset(nstorms, nmax_VDM_fixes)
VDM_FIX_outbound_maximum_flight_level_wind_timeoffset:description = "Timeoffset of the observed maximum flight level wind speed observed on the outbound leg (if it is higher than the inbound maximum reported in item FOXTROT) (a type 3 required remark), translated from the REMARKS section from each unique VDM"
VDM_FIX_outbound_maximum_flight_level_wind_timeoffset:units = "seconds since 1970-01-01:00:00:00 UTC"

int VDM_FIX_secondary_maximum_flight_level_wind(nstorms, nmax_VDM_fixes)

VDM_FIX_secondary_maximum_flight_level_wind:units = "none"

VDM_FIX_secondary_maximum_flight_level_wind:description = "Boolean that indicates whether an optional remark is present in the REMARKS section to indicate the presence of a significant secondary maximum flight level wind. This optional remark normally provides the maximum flight level wind speed of the secondary wind maximum. The time of observation and the relative quadrant of the storm may also be included. This parameter is translated from the REMARKS line of each unique VDM. Note: this will normally at greater radius than the inbound flight level wind maximum reported in item FOXTROT, but this need not always be the case."

int VDM_FIX_secondary_maximum_flight_level_wind_speed(nstorms, nmax_VDM_fixes)

VDM_FIX_secondary_maximum_flight_level_wind_speed:description = "Maximum flight level wind speed associated with a significant secondary wind maximum at flight level (an optional remark), translated from the REMARKS section of each unique VDM"

VDM_FIX_secondary_maximum_flight_level_wind_speed:units = "knot"

int VDM_FIX_secondary_maximum_flight_level_wind_bearing(nstorms, nmax_VDM_fixes)

VDM_FIX_secondary_maximum_flight_level_wind_bearing:description = "Bearing (from the flight level fix center coordinates given in BRAVO) to the location of the maximum flight level wind speed associated with a significant secondary wind maximum at flight level (an optional remark), translated from the REMARKS section of each unique VDM"

VDM_FIX_secondary_maximum_flight_level_wind_bearing:units = "degrees"

float VDM_FIX_secondary_maximum_flight_level_wind_range(nstorms, nmax_VDM_fixes)

VDM_FIX_secondary_maximum_flight_level_wind_range:description = "Range (from the flight level fix center coordinates given in BRAVO) to the location of the maximum flight level wind speed associated with a significant secondary wind maximum at flight level (an optional remark), translated from the REMARKS section of each unique VDM"

VDM_FIX_secondary_maximum_flight_level_wind_range:units = "nautical_mile"

char VDM_FIX_secondary_maximum_flight_level_wind_relative_quadrant(nstorms, nmax_VDM_fixes, VDM_FIX_secondary_maximum_flight_level_wind_relative_quadrant_StrLen)

VDM_FIX_secondary_maximum_flight_level_wind_relative_quadrant:units = "none"

VDM_FIX_secondary_maximum_flight_level_wind_relative_quadrant:description = "Relative quadrant of the storm of the observed maximum flight level wind speed associated with a significant secondary wind maximum at flight level (an optional remark), translated from the REMARKS section of each unique VDM" char VDM_FIX_secondary_maximum_flight_level_wind_day(nstorms, nmax_VDM_fixes, VDM_FIX_secondary_maximum_flight_level_wind_day_StrLen)

VDM_FIX_secondary_maximum_flight_level_wind_day:units = "none"

VDM_FIX_secondary_maximum_flight_level_wind_day:description = "2-digit day (string) of the observed maximum flight level wind speed associated with a significant secondary wind maximum at flight level (an optional remark), translated from the REMARKS section of each unique VDM"

char VDM_FIX_secondary_maximum_flight_level_wind_time(nstorms, nmax_VDM_fixes, VDM_FIX_secondary_maximum_flight_level_wind_time_StrLen)

VDM_FIX_secondary_maximum_flight_level_wind_time:units = "none"

VDM_FIX_secondary_maximum_flight_level_wind_time:description = "4-digit day (string) of the observed maximum flight level wind speed associated with a significant secondary wind maximum at flight level (an optional remark), translated from the REMARKS section of each unique VDM"

char VDM_FIX_secondary_maximum_flight_level_wind_yyyymmddhhmm(nstorms, nmax_VDM_fixes, VDM_FIX_secondary_maximum_flight_level_wind_yyyymmddhhmm_StrLen)

VDM_FIX_secondary_maximum_flight_level_wind_yyyymmddhhmm:units = "UTC"

VDM_FIX_secondary_maximum_flight_level_wind_yyyymmddhhmm:description = "Date/time (string representation, yyyymmddhhmm: yyyy = 4-digit year, mm = 2-digit month, dd = 2-digit day, hh = 2-digit hour, mm = 2-digit minute) corresponding to the observation time of the observed maximum flight level wind speed associated with a significant secondary wind maximum at flight level (an optional remark), translated from the REMARKS section of each unique VDM"

int VDM_FIX_secondary_maximum_flight_level_wind_timeoffset(nstorms, nmax_VDM_fixes)

VDM_FIX_secondary_maximum_flight_level_wind_timeoffset:description = "Timeoffset of the observed maximum flight level wind speed associated with a significant secondary wind maximum at flight level (an optional remark), translated from the REMARKS section of each unique VDM"

VDM_FIX_secondary_maximum_flight_level_wind_timeoffset:units = "seconds since 1970-01-01:00:00:00 UTC"

int VDM_FIX_maximum_surface_wind(nstorms, nmax_VDM_fixes)

VDM_FIX_maximum_surface_wind:units = "none"

VDM_FIX_maximum_surface_wind:description = "Boolean that indicates whether an optional remark is present in the REMARKS section to provide the maximum surface wind on the latest pass through any portion of the storm. This optional remark normally provides the maximum surface wind speed, the time of observation, and the relative quadrant of the storm. This parameter is translated from the REMARKS line of each unique VDM."

int VDM_FIX_maximum_surface_wind_by_sfmr(nstorms, nmax_VDM_fixes)
VDM_FIX_maximum_surface_wind_by_sfmr:units = "none"
VDM_FIX_maximum_surface_wind_by_sfmr:description = "Boolean that indicates whether the maximum surface wind observation was measured by SFMR. 'True' means that it was, 'False' means that it was determined visually. Translated from the optional surface maximum wind report in the REMARKS section."

int VDM_FIX_maximum_surface_wind_speed(nstorms, nmax_VDM_fixes)
VDM_FIX_maximum_surface_wind_speed:description = "Maximum surface wind speed on the latest pass through any portion of the storm, translated from an optional report in the REMARKS section of each unique VDM"
VDM_FIX_maximum_surface_wind_speed:units = ""

int VDM_FIX_maximum_surface_wind_bearing(nstorms, nmax_VDM_fixes)
VDM_FIX_maximum_surface_wind_bearing:description = "Bearing (from the flight level fix center coordinates given in BRAVO) to the location of the maximum surface wind speed on the latest pass through any portion of the storm, translated from an optional report in the REMARKS section of each unique VDM"
VDM_FIX_maximum_surface_wind_bearing:units = "degrees"

float VDM_FIX_maximum_surface_wind_range(nstorms, nmax_VDM_fixes)
VDM_FIX_maximum_surface_wind_range:description = "Range (from the flight level fix center coordinates given in BRAVO) to the location of the maximum surface wind speed on the latest pass through any portion of the storm, translated from an optional report in the REMARKS section of each unique VDM"
VDM_FIX_maximum_surface_wind_range:units = "nautical_mile"

char VDM_FIX_maximum_surface_wind_relative_quadrant(nstorms, nmax_VDM_fixes, VDM_FIX_maximum_surface_wind_relative_quadrant_StrLen)
VDM_FIX_maximum_surface_wind_relative_quadrant:units = "none"
VDM_FIX_maximum_surface_wind_relative_quadrant:description = "Relative quadrant of the storm of the observed maximum surface wind speed on the latest pass through any portion of the storm, translated from an optional report in the REMARKS section of each unique VDM"

char VDM_FIX_maximum_surface_wind_day(nstorms, nmax_VDM_fixes, VDM_FIX_maximum_surface_wind_day_StrLen)
VDM_FIX_maximum_surface_wind_day:units = "UTC"
VDM_FIX_maximum_surface_wind_day:description = "2-digit day (string) of the observed maximum surface wind speed on the latest pass through any portion of the storm, translated from an optional report in the REMARKS section of each unique VDM"

char VDM_FIX_maximum_surface_wind_time(nstorms, nmax_VDM_fixes, VDM_FIX_maximum_surface_wind_time_StrLen)
VDM_FIX_maximum_surface_wind_time:units = "UTC"

VDM_FIX_maximum_surface_wind_time:description = "4-digit day (string) of the observed maximum surface wind speed on the latest pass through any portion of the storm, translated from an optional report in the REMARKS section of each unique VDM"

char VDM_FIX_maximum_surface_wind_yyyymmddhhmm(nstorms, nmax_VDM_fixes, VDM_FIX_maximum_surface_wind_yyyymmddhhmm_StrLen)

VDM_FIX_maximum_surface_wind_yyyymmddhhmm:units = "UTC"

VDM_FIX_maximum_surface_wind_yyyymmddhhmm:description = "Date/time (string representation, yyyymmddhhmm: yyyy = 4-digit year, mm = 2-digit month, dd = 2-digit day, hh = 2-digit hour, mm = 2-digit minute) corresponding to the observation time of the observed maximum surface wind speed on the latest pass through any portion of the storm, translated from an optional report in the REMARKS section of each unique VDM"

int VDM_FIX_maximum_surface_wind_timeoffset(nstorms, nmax_VDM_fixes)

VDM_FIX_maximum_surface_wind_timeoffset:description = "Timeoffset of the observed maximum surface wind speed on the latest pass through any portion of the storm, translated from an optional report in the REMARKS section of each unique VDM"

VDM_FIX_maximum_surface_wind_timeoffset:units = "seconds since 1970-01-01:00:00:00 UTC"

int VDM_FIX_outbound_maximum_surface_wind(nstorms, nmax_VDM_fixes)

VDM_FIX_outbound_maximum_surface_wind:units = "none"

VDM_FIX_outbound_maximum_surface_wind:description = "Boolean that indicates the whether an optional remark is present in the REMARKS section, which provides the maximum surface wind speed observed on the outbound leg (if it is higher than the inbound maximum reported in item DELTA). This remark type includes information about the maximum surface wind observed on the outbound leg, the time of observation, and the relative quadrant of the storm of the qualifying outbound wind. If, after transmission of the vortex message, but prior to the aircraft reaching the cross-leg turn point, a higher qualifying outbound wind is observed, then the vortex message will be amended with the higher outbound wind reported. This parameter is translated from the REMARKS line of each unique VDM."

int VDM_FIX_outbound_maximum_surface_wind_speed(nstorms, nmax_VDM_fixes)

VDM_FIX_outbound_maximum_surface_wind_speed:description = "Maximum surface wind speed observed on the outbound leg (if it is higher than the inbound maximum reported in item DELTA) (an optional remark), translated from the REMARKS section from each unique VDM"

VDM_FIX_outbound_maximum_surface_wind_speed:units = "knot"

int VDM_FIX_outbound_maximum_surface_wind_bearing(nstorms, nmax_VDM_fixes)

VDM_FIX_outbound_maximum_surface_wind_bearing:description = "Bearing (from the flight level fix center coordinates given in BRAVO) to the location of the maximum surface wind speed observed on the outbound leg (if it is higher than the inbound maximum reported in item DELTA) (an optional remark), translated from the REMARKS section from each unique VDM"

VDM_FIX_outbound_maximum_surface_wind_bearing:units = "degrees"

float VDM_FIX_outbound_maximum_surface_wind_range(nstorms, nmax_VDM_fixes)
VDM_FIX_outbound_maximum_surface_wind_range:description = "Range (from the flight level fix center coordinates given in BRAVO) to the location of the maximum surface wind speed observed on the outbound leg (if it is higher than the inbound maximum reported in item DELTA) (an optional remark), translated from the REMARKS section from each unique VDM"
VDM_FIX_outbound_maximum_surface_wind_range:units = "nautical_mile"

char VDM_FIX_outbound_maximum_surface_wind_relative_quadrant(nstorms, nmax_VDM_fixes, VDM_FIX_outbound_maximum_surface_wind_relative_quadrant_StrLen)
VDM_FIX_outbound_maximum_surface_wind_relative_quadrant:units = "none"
VDM_FIX_outbound_maximum_surface_wind_relative_quadrant:description = "Relative quadrant of the storm of the observed maximum surface wind speed observed on the outbound leg (if it is higher than the inbound maximum reported in item DELTA) (an optional remark), translated from the REMARKS section from each unique VDM"

char VDM_FIX_outbound_maximum_surface_wind_day(nstorms, nmax_VDM_fixes, VDM_FIX_outbound_maximum_surface_wind_day_StrLen)
VDM_FIX_outbound_maximum_surface_wind_day:units = "UTC"
VDM_FIX_outbound_maximum_surface_wind_day:description = "2-digit day (string) of the observed maximum surface wind speed observed on the outbound leg (if it is higher than the inbound maximum reported in item DELTA) (an optional remark), translated from the REMARKS section from each unique VDM"

char VDM_FIX_outbound_maximum_surface_wind_time(nstorms, nmax_VDM_fixes, VDM_FIX_outbound_maximum_surface_wind_time_StrLen)
VDM_FIX_outbound_maximum_surface_wind_time:units = "UTC"
VDM_FIX_outbound_maximum_surface_wind_time:description = "4-digit day (string) of the observed maximum surface wind speed observed on the outbound leg (if it is higher than the inbound maximum reported in item DELTA) (an optional remark), translated from the REMARKS section from each unique VDM"

char VDM_FIX_outbound_maximum_surface_wind_yyyymmddhhmm(nstorms, nmax_VDM_fixes, VDM_FIX_outbound_maximum_surface_wind_yyyymmddhhmm_StrLen)
VDM_FIX_outbound_maximum_surface_wind_yyyymmddhhmm:units = ""
VDM_FIX_outbound_maximum_surface_wind_yyyymmddhhmm:description = "Date/time (string representation, yyyymmddhhmm: yyyy = 4-digit year, mm = 2-digit month, dd = 2-digit day, hh = 2-digit hour, mm = 2-digit minute) corresponding to the observation time of the observed maximum surface wind speed observed on the outbound leg (if it is higher than the inbound maximum reported in item DELTA) (an optional remark), translated from the REMARKS section from each unique VDM"

int VDM_FIX_outbound_maximum_surface_wind_timeoffset(nstorms, nmax_VDM_fixes)

VDM_FIX_outbound_maximum_surface_wind_timeoffset:description = "Timeoffset of the observed maximum surface wind speed observed on the outbound leg (if it is higher than the inbound maximum reported in item DELTA) (an optional remark), translated from the REMARKS section from each unique VDM"
VDM_FIX_outbound_maximum_surface_wind_timeoffset:units = "seconds since 1970-01-01:00:00:00 UTC"

int VDM_FIX_combined_maximum_flight_level_wind_speed(nstorms, nmax_VDM_fixes)
VDM_FIX_combined_maximum_flight_level_wind_speed:description = "A combined maximum flight level wind speed, obtained by taking the overall maximum of the four different types of maximum flight level wind speed reports from items FOXTROT and the REMARKS section ('VDM_FIX_inbound_maximum_flight_level_wind_speed', 'VDM_FIX_maximum_flight_level_wind_speed', 'VDM_FIX_outbound_maximum_flight_level_wind_speed', and 'VDM_FIX_secondary_maximum_flight_level_wind_speed'). In order to be considered as a candidate for the overall maximum, each of the given wind speed values must have been no more than 120 minutes prior to the time of the current center fix, or no more than 30 minutes after the time of the current center fix. These time constraint criteria generally ensure that the maxima are associated with this particular center fix and not a previous or future fix. The forward time constraint criteria ensures that the maximum values from amended or corrected VDMs are included here."
VDM_FIX_combined_maximum_flight_level_wind_speed:units = "knot"

char VDM_FIX_combined_maximum_flight_level_wind_yyyymmddhhmm(nstorms, nmax_VDM_fixes, VDM_FIX_combined_maximum_flight_level_wind_yyyymmddhhmm_StrLen)
VDM_FIX_combined_maximum_flight_level_wind_yyyymmddhhmm:units = "UTC"

VDM_FIX_combined_maximum_flight_level_wind_yyyymmddhhmm:description = "Date/time (string representation, yyyymmddhhmm: yyyy = 4-digit year, mm = 2-digit month, dd = 2-digit day, hh = 2-digit hour, mm = 2-digit minute) corresponding to the observation time of the combined maximum flight level wind speed"

int VDM_FIX_combined_maximum_flight_level_wind_timeoffset(nstorms, nmax_VDM_fixes)
VDM_FIX_combined_maximum_flight_level_wind_timeoffset:description = "Timeoffset of the observation time of the observed combined maximum flight level wind speed"
VDM_FIX_combined_maximum_flight_level_wind_timeoffset:units = "seconds since 1970-01-01:00:00:00 UTC"

int VDM_FIX_combined_maximum_surface_wind_speed(nstorms, nmax_VDM_fixes)
VDM_FIX_combined_maximum_surface_wind_speed:description = "A combined maximum flight level wind speed, obtained by taking the overall maximum of the

three different types of maximum surface wind speed reports from items DELTA and the REMARKS section ('VDM_FIX_inbound_maximum_surface_wind_speed', 'VDM_FIX_maximum_surface_wind_speed', 'VDM_FIX_outbound_maximum_surface_wind_speed', and 'VDM_FIX_secondary_maximum_surface_wind_speed'). In order to be considered as a candidate for the overall maximum, each of the given wind speed values must have been no more than 120 minutes prior to the time of the current center fix, or no more than 30 minutes after the time of the current center fix. These time constraint criteria generally ensure that the maxima are associated with this particular center fix and not a previous or future fix. The forward time constraint criteria ensures that the maximum values from amended or corrected VDMs are included here."

VDM_FIX_combined_maximum_surface_wind_speed:units = "knot"

char VDM_FIX_combined_maximum_surface_wind_yyyymmddhhmm(nstorms, nmax_VDM_fixes, VDM_FIX_combined_maximum_surface_wind_yyyymmddhhmm_StrLen)

VDM_FIX_combined_maximum_surface_wind_yyyymmddhhmm:units = "UTC"

VDM_FIX_combined_maximum_surface_wind_yyyymmddhhmm:description = "Date/time (string representation, yyyymmddhhmm: yyyy = 4-digit year, mm = 2-digit month, dd = 2-digit day, hh = 2-digit hour, mm = 2-digit minute) corresponding to the observation time of the combined maximum surface wind speed"

int VDM_FIX_combined_maximum_surface_wind_timeoffset(nstorms, nmax_VDM_fixes)

VDM_FIX_combined_maximum_surface_wind_timeoffset:description = "Timeoffset of the observation time of the combined maximum surface wind speed"

VDM_FIX_combined_maximum_surface_wind_timeoffset:units = "seconds since 1970-01-01:00:00:00 UTC"

float VDM_FIX_supplementary_maximum_flight_level_temperature(nstorms, nmax_VDM_fixes)

VDM_FIX_supplementary_maximum_flight_level_temperature:description = "Supplementary maximum flight level temperature, if not within 5 nm of the flight level center and higher than the maximum flight level temperature reported for the central region in item JULIET) (a type 5 required remark), translated from the REMARKS section from each unique VDM"

VDM_FIX_supplementary_maximum_flight_level_temperature:units = "celsius"

int VDM_FIX_supplementary_maximum_flight_level_temperature_bearing(nstorms, nmax_VDM_fixes)

VDM_FIX_supplementary_maximum_flight_level_temperature_bearing:description = "Bearing (from the flight leel fix center coordinates given in BRAVO) to the location of the supplementary maximum flight level temperature (a type 5 required remark), translated from the REMARKS section from each unique VDM"

VDM_FIX_supplementary_maximum_flight_level_temperature_bearing:units = "degrees"

float VDM_FIX_supplementary_maximum_flight_level_temperature_range(nstorms, nmax_VDM_fixes)

VDM_FIX_supplementary_maximum_flight_level_temperature_range:description = "Range (from the flight level fix center coordinates given in BRAVO) to the location of the supplementary maximum flight level temperature (a type 5 required remark), translated from the REMARKS section from each unique VDM"
VDM_FIX_supplementary_maximum_flight_level_temperature_range:units = "nautical_mile"

char VDM_FIX_minimum_sea_level_pressure_extrapolated_level_string(nstorms, nmax_VDM_fixes, VDM_FIX_minimum_sea_level_pressure_extrapolated_level_string_StrLen)
VDM_FIX_minimum_sea_level_pressure_extrapolated_level_string:units = "none"
VDM_FIX_minimum_sea_level_pressure_extrapolated_level_string:description = "String containing the information on how the minimum sea level pressure was extrapolated, and what level it was extrapolated from (a type 4 required remark). This is typically of the format: 'SLP EXTRAPOLATED FROM BELOW 1500 FT', however there are many other variants. Sometimes a secondary pressure center is noted (e.g. 'SECONDARY MIN EXTRAP SLP OF 1002MB RECORDED 10NM EAST OF FLT LEVEL CIRCULATION CENTER.'), or the extrapolation message does not indicate the level the extrapolation was made from (e.g. 'EXTRAPOLATED SLP 982 MB'). These variant instances are not currently read and translated in the dataset."

int VDM_FIX_minimum_sea_level_pressure_extrapolated_level(nstorms, nmax_VDM_fixes)
VDM_FIX_minimum_sea_level_pressure_extrapolated_level:description = "Level from which the sea level pressure was extrapolated from (a type 4 required remark), translated from the REMARKS section of each unique VDM (if present). Valid values include: '14' = BELOW 1500 FEET, '15' = 1500 FEET, '9' = 925 MB, '8' = 850 MB, '7' = 700 MB."
VDM_FIX_minimum_sea_level_pressure_extrapolated_level:units = "none"

int VDM_FIX_surface_center_displaced(nstorms, nmax_VDM_fixes)
VDM_FIX_surface_center_displaced:units = "none"
VDM_FIX_surface_center_displaced:description = "Boolean indicating whether the surface center was noted to be displaced 5 or more nm from the flight level center (a type 5 required remark) is present in the REMARKS section of each unique VDM. When present, the bearing and range of the surface center is given."

int VDM_FIX_surface_center_displaced_bearing(nstorms, nmax_VDM_fixes)

VDM_FIX_surface_center_displaced_bearing:description = "Bearing (from the flight level fix center coordinates given in BRAVO) to the location of the displaced surface center (a type 5 required remark), translated from the REMARKS section of each unique VDM"
VDM_FIX_surface_center_displaced_bearing:units = "degrees"

float VDM_FIX_surface_center_displaced_range(nstorms, nmax_VDM_fixes)
VDM_FIX_surface_center_displaced_range:description = "Range (from the flight level fix center coordinates given in BRAVO) to the location of the displaced surface center (a type 5 required remark), translated from the REMARKS section of each unique VDM"
VDM_FIX_surface_center_displaced_range:units = "nautical_mile"

int VDM_FIX_lightning_observed(nstorms, nmax_VDM_fixes)
VDM_FIX_lightning_observed:units = "none"
VDM_FIX_lightning_observed:description = "Boolean indicating whether the presence of lightning (an optional report) was indicated in the REMARKS section, translated from each unique VDM. Note: this is reported fairly infrequently, and may not always be noted when the flight occurs during the daylight hours. "

char VDM_FIX_lightning_direction(nstorms, nmax_VDM_fixes, VDM_FIX_lightning_direction_StrLen)
VDM_FIX_lightning_direction:units = "none"
VDM_FIX_lightning_direction:description = "Direction of the lightning noted in the optional report in the REMARKS section, translated from each unique VDM. Due to challenges in semantic processing of the free-form remarks, it is not guaranteed that the quadrant noted is positively identified with the location of the lightning, but this will normally be the case."

int VDM_FIX_hail_observed(nstorms, nmax_VDM_fixes)
VDM_FIX_hail_observed:units = "none"
VDM_FIX_hail_observed:description = "Boolean indicating whether the presence of hail (an optional report) was indicated in the REMARKS section, translated from each unique VDM. Note: hail is rarely reported and there is no guarantee that every instance of hail or graupel will be reported."

char VDM_FIX_hail_direction(nstorms, nmax_VDM_fixes, VDM_FIX_hail_direction_StrLen)
VDM_FIX_hail_direction:units = "none"
VDM_FIX_hail_direction:description = "Direction of the hail noted in the optional report in the REMARKS secion from each unique VDM."

int VDM_FIX_banding_observed(nstorms, nmax_VDM_fixes)
VDM_FIX_banding_observed:units = "none"
VDM_FIX_banding_observed:description = "Boolean indicating whether the presence of banding was indicated either in the LIMA line or in the REMARKS section, translated from each unique VDM. Note: banding is reported fairly frequently prior to the formation of an eyewall. It is often indicated by: 'BANDING', 'SPIRAL BANDING', 'SPIRAL

BANDS', 'BANDS'. Due to the semantic challenges of translating the free-form remarks, it is not guaranteed that every instance of BANDING will correspond to actual banding (for instance, would 'RAINBANDS' qualify?), nevertheless, this is normally the case."

char VDM_FIX_banding_presentation(nstorms, nmax_VDM_fixes, VDM_FIX_banding_presentation_StrLen)

VDM_FIX_banding_presentation:units = "none"

VDM_FIX_banding_presentation:description = "Descriptor of the radar presentation of the noted banding, indicated either in the LIMA line or in the REMARKS section, translated from each unique VDM. The following descriptors are mapped onto 'GOOD': 'EXCELLENT', 'GOOD', 'DECENT', 'STRONG', WELL DEFINED', and 'FAIR'. The following descriptors are mapped onto 'POOR': 'WEAK', 'SOME', 'THIN', DISORGANIZED', 'POOR'."

char VDM_FIX_banding_direction(nstorms, nmax_VDM_fixes, VDM_FIX_banding_direction_StrLen)

VDM_FIX_banding_direction:units = "none"

VDM_FIX_banding_direction:description = "Direction of the reported banding (e.g. 'NW'), indicated either in the LIMA line or in the REMARKS seciton, translated from each unique VDM"

char VDM_FIX_radar_presentation(nstorms, nmax_VDM_fixes, VDM_FIX_radar_presentation_StrLen)

VDM_FIX_radar_presentation:units = "none"

VDM_FIX_radar_presentation:description = "Description of the radar presentation, as subjectively determined by the flight meteorologist, indicated either in the MIKE line or in the REMARKS section, translated from each unique VDM. The following descriptors are mapped onto 'EXCELLENT': 'EXCELLENT', 'TEXTBOOK EYEWALL', 'WELL DEFINED', and 'CLASSIC'. The following descriptors are mapped onto 'GOOD': 'GOOD' and 'FAIR'. The following descriptors are mapped onto 'POOR': 'POOR' and 'WEAK'. 'POOR' is not indicated however, if it was noted that there could be attenuation of the radar, or if the presentation was only poor in one direction."

int VDM_FIX_stadium_effect(nstorms, nmax_VDM_fixes)

VDM_FIX_stadium_effect:units = "none"

VDM_FIX_stadium_effect:description = "Boolean that indicates whether the phrase 'STADIUM EFFECT' was noted either in the MIKE item or in the REMARKS section, translated from each unique VDM"

int VDM_FIX_eye_concentricity(nstorms, nmax_VDM_fixes)

VDM_FIX_eye_concentricity:units = "none"

VDM_FIX_eye_concentricity:description = "Boolean that indicates whether concentric eyewalls were noted, based on information either in the MIKE item or in the REMARKS section, translated from each unique VDM. This will be set to 'True' if any of the following words occur in item MIKE or a given phrase of the REMARKS section in conjunction with the word 'EYE': 'INNER', 'OUTER', 'SECONDARY', or 'DOUBLE', or if more than one eyewall diameters are given in item MIKE. Due to the uncertainties in semantically decoding the free-form remarks, it cannot be guaranteed that a concentric eyewall will always be present, but this is normally the case."

Calculated Data Parameters

double CALC_BT_coriolis_parameter(nstorms, nmax_BT_points)

CALC_BT_coriolis_parameter:description = "Corolis parameter (2*OMEGA*sine(LAT), where OMEGA is the earth's rotation rate in radians per second and LAT is the latitude of the storm center from the Best Track b-deck file, for each time"

CALC_BT_coriolis_parameter:units = "1/second"

double CALC_BT_surface_rossby_radius_shrinkage_factor(nstorms, nmax_BT_points)

CALC_BT_surface_rossby_radius_shrinkage_factor:description = "Rossby radius shrinkage factor computed using the maximum wind speed value and radii of the maximum wind speed given in the Best Track b-deck file, for each time. This is computed as follows: (f + 2*VMAX/RMW) / f, where VMAX and RMW are the maximum wind speed (converted to meters/second) and radius of maximum wind (converted to meters) from the Best Track b-deck file. Note: this factor gives an idea of how much the Rossby length has shrunk from the background value. It is effectively a poor man's inertial stability, as it is the average vorticity within the RMW divided by the Coriolis parameter."

CALC_BT_surface_rossby_radius_shrinkage_factor:units = "dimensionless"

double CALC_BT_minimum_surface_rossby_radius(nstorms, nmax_BT_points)

CALC_BT_minimum_surface_rossby_radius:description = "Estimated Rossby radius, computed as 10E6 meters / CALC_BT_surface_rossby_radius_shrinkage_factor, for each time"

CALC_BT_minimum_surface_rossby_radius:units = "meter"

double CALC_BT_dynamical_eye_size(nstorms, nmax_BT_points)

CALC_BT_dynamical_eye_size:description = "Dynamical eye size, computed as (EYE_DIAMETER/2) / CALC_BT_minimum_surface_rossby_radius, where the eye diameter has been converted to meters, for each time"

CALC_BT_dynamical_eye_size:units = "dimensionless"

double CALC_BT_RMW_minus_eye_radius(nstorms, nmax_BT_points)

CALC_BT_RMW_minus_eye_radius:description = "Difference between the eye radius and the radius of maximum winds, using the eyewall diameter and radius of surface maximum wind information from the Best Track b-deck file, for each time"

CALC_BT_RMW_minus_eye_radius:units = "meter"

double CALC_BT_relative_angular_momentum(nstorms, nmax_BT_points)

CALC_BT_relative_angular_momentum:description = "Relative angular momentum, computed as VMAX * RMW, where VMAX is the maximum surface wind speed (converted to meters/second) and RMW is the radius of that wind speed (converted to meters), using the information from the Best Track b-deck file, for each time"

CALC_BT_relative_angular_momentum:units = "meter^2/second"

double CALC_BT_absolute_angular_momentum(nstorms, nmax_BT_points)
CALC_BT_absolute_angular_momentum:description = "Absolute angular momentum, computed as VMAX * RMW + (f * RMW^2)/2, where VMAX is the maximum surface wind speed (converted to meters/second), RMW is the radius of that wind speed (converted to meters), and f is the Coriolis parameter, computed using the information from the Best Track b-deck file, for each time"
CALC_BT_absolute_angular_momentum:units = "meter^2/second"

double CALC_VDM_coriolis_parameter(nstorms, nmax_VDM_fixes)
CALC_VDM_coriolis_parameter:description = "Corolis parameter (2*OMEGA*sine(LAT), where OMEGA is the earth's rotation rate in radians per second and LAT is the latitude of the fixed flight level center from the VDM, for each fix time"
CALC_VDM_coriolis_parameter:units = "1/second"

double CALC_VDM_flight_level_rossby_radius_shrinkage_factor(nstorms, nmax_VDM_fixes)
CALC_VDM_flight_level_rossby_radius_shrinkage_factor:description = "Rossby radius shrinkage factor computed using the inbound maximum flight level wind speed value and range of the inbound maximum flight level wind speed associated with the fix from each unique VDM. This is computed as follows: (f + 2*VMAX/RMW) / f, where VMAX and RMW are the inbound maximum wind speed (converted to meters/second) and the associated range for that value (converted to meters), from the VDM. Note: this factor gives an idea of how much the Rossby length has shrunk from the background value. It is effectively a poor man's inertial stability, as it is the average vorticity within the RMW divided by the Coriolis parameter."
CALC_VDM_flight_level_rossby_radius_shrinkage_factor:units = "dimensionless"

double CALC_VDM_flight_level_rossby_radius_shrinkage_factor_combined(nstorms, nmax_VDM_fixes)

CALC_VDM_flight_level_rossby_radius_shrinkage_factor_combined:description = "Rossby radius shrinkage factor computed using the combined maximum flight level wind speed value and range of the inbound maximum flight level wind speed associated with the fix from each unique VDM. This is computed as follows: (f + 2*VMAX/RMW) / f, where VMAX and RMW are the combined flight level maximum wind speed (converted to meters/second) and the range of the inbound flight level maximum (converted to meters), from the VDM. Note: this factor gives an idea of how much the Rossby length has shrunk from the background value. It is effectively a poor man's inertial stability, as it is the average vorticity within the RMW divided by the Coriolis parameter. Because there is no guarantee that the combined maximum wind speed will have a range similar to the inbound maximum flight level wind speed, this quantity may not be very accurate, however it provides an upper bound on the shrinkage factor, provided that the combined wind maximum occurs at a larger radius than the inbound wind maximum."

CALC_VDM_flight_level_rossby_radius_shrinkage_factor_combined:units = "dimensionless"

double CALC_VDM_minimum_flight_level_rossby_radius(nstorms, nmax_VDM_fixes)
CALC_VDM_minimum_flight_level_rossby_radius:description = "Estimated minimum Rossby radius, computed as 10E6 / CALC_VDM_flight_level_rossby_radius_shrinkage_factor_combined, for each time."
CALC_VDM_minimum_flight_level_rossby_radius:units = "meter"

double CALC_VDM_minimum_flight_level_rossby_radius_combined(nstorms, nmax_VDM_fixes)
CALC_VDM_minimum_flight_level_rossby_radius_combined:description = "Estimated minimum Rossby radius, computed as 10E6 / CALC_VDM_flight_level_rossby_radius_shrinkage_factor_combined, for each time"
CALC_VDM_minimum_flight_level_rossby_radius_combined:units = "meter"

double CALC_VDM_dynamical_eye_size(nstorms, nmax_VDM_fixes)
CALC_VDM_dynamical_eye_size:description = "Dynamical eye size, computed as (EYE_DIAMETER/2) / CALC_VDM_minimum_flight_level_rossby_radius, where the eye diameter has been converted to meters, for each unique VDM"
CALC_VDM_dynamical_eye_size:units = "dimensionless"

double CALC_VDM_dynamical_eye_size_combined(nstorms, nmax_VDM_fixes)
CALC_VDM_dynamical_eye_size_combined:description = "Dynamical eye size, computed as (EYE_DIAMETER/2) / CALC_VDM_minimum_flight_level_rossby_radius_combined, where the eye diameter has been converted to meters, for each unique VDM"
CALC_VDM_dynamical_eye_size_combined:units = "dimensionless"

double CALC_VDM_RMW_minus_eye_radius(nstorms, nmax_VDM_fixes)
CALC_VDM_RMW_minus_eye_radius:description = "Difference between the eye radius and the radius of maximum winds, using the eyewall diameter and radius of flight level maximum wind information from each unique VDM"
CALC_VDM_RMW_minus_eye_radius:units = "meter"

double CALC_VDM_flight_level_relative_angular_momentum(nstorms, nmax_VDM_fixes)
CALC_VDM_flight_level_relative_angular_momentum:description = "Relative angular momentum, computed as VMAX * RMW, where VMAX is the maximum flight level wind speed (converted to meters/second) and RMW is the radius of that wind speed (converted to meters), using the information from each unique VDM"
CALC_VDM_flight_level_relative_angular_momentum:units = "meter^2/second"

double CALC_VDM_flight_level_absolute_angular_momentum(nstorms, nmax_VDM_fixes)

CALC_VDM_flight_level_absolute_angular_momentum:description = "Absolute angular momentum, computed as VMAX * RMW + (f * RMW^2)/2, where VMAX is the maximum flight level wind speed (converted to meters/second), RMW is the radius of that wind speed (converted to meters), and f is the Coriolis parameter, computed using the information from each unique VDM"

CALC_VDM_flight_level_absolute_angular_momentum:units = "meter^2/second"

SHIPS Storm Description Parameters

int SHIPS_datapresent(nstorms)

SHIPS_datapresent:units = "none"

int SHIPS_STORM_npts(nstorms)

SHIPS_STORM_npts:description = "Number of SHIPS data points that exist for this storm"

SHIPS_STORM_npts:units = "count"

SHIPS Dimension Parameters

char SHIPS_yyyymmddhhmm(nstorms, nmax_SHIPS_points, SHIPS_yyyymmddhhmm_StrLen)

SHIPS_yyyymmddhhmm:units = "UTC"

SHIPS_yyyymmddhhmm:description = "Date/time (string representation) corresponding to the analysis time (hour 0) for this case, yyyymmddhhmm: yyyy = 4-digit year, mm = 2-digit month, dd = 2-digit day, hh = 2-digit hour, mm = 2-digit minute"

int SHIPS_timeoffset(nstorms, nmax_SHIPS_points)

SHIPS_timeoffset:description = "Time corresponding to the analysis time (hour 0) for this case, as measured as an offset"

SHIPS_timeoffset:units = "seconds since 1970-01-01 00:00:00.0 UTC"

SHIPS Data Parameters

int SHIPS_VMAX(nstorms, nmax_SHIPS_points)

SHIPS_VMAX:description = "Maximum sustained (1-min average) surface wind speed at the analysis time (copied from Best Track)"

SHIPS_VMAX:units = "knot"

int SHIPS_MSLP(nstorms, nmax_SHIPS_points)

SHIPS_MSLP:description = "Minimum sea level pressure at the analysis time (from Best Track)"

SHIPS_MSLP:units = "millibar"

```
	int SHIPS_TYPE(nstorms, nmax_SHIPS_points)
		SHIPS_TYPE:description = "Storm type at the analysis time(0 = wave, remnant low, dissipating low, 1 = tropical, 2 = subtropical, 3 = extra-tropical). Note that the SHIPS parameters are set to missing for all cases except type = 1 or 2, since these are not included in the SHIPS developmental sample for estimating model coefficients."
		SHIPS_TYPE:units = "none"

	int SHIPS_HIST(nstorms, nmax_SHIPS_points)
		SHIPS_HIST:description = "Storm history parameter. The number of 6 hr periods the storm maximum wind has been above 20, 25, ..., 120 kt."
		SHIPS_HIST:units = "hour"

	int SHIPS_DELV(nstorms, nmax_SHIPS_points)
		SHIPS_DELV:description = "Intensity change relative to the analysis time (hour 0) of the case, i.e., from -12 to 0, -6 to 0, 0 to 0, 0 to 6, 0 to 120 hr. NOTE: this should be zero in the aggregated structure and intensity dataset since only the analysis time value is included there."
		SHIPS_DELV:units = "knot"

	int SHIPS_INCV(nstorms, nmax_SHIPS_points)
		SHIPS_INCV:description = "Intensity change in preceeding 6-hr period relative to the analysis time, i.e., from -6 to 0 hr."
		SHIPS_INCV:units = "knot"

	float SHIPS_LAT(nstorms, nmax_SHIPS_points)
		SHIPS_LAT:description = "Latitude of the storm center at the analysis time"
		SHIPS_LAT:units = "degrees_north"

	float SHIPS_LON(nstorms, nmax_SHIPS_points)
		SHIPS_LON:description = "Longitude of the storm center at the analysis time"
		SHIPS_LON:units = "degrees_east"

	float SHIPS_CSST(nstorms, nmax_SHIPS_points)
		SHIPS_CSST:description = "Climatological SST at the analysis time"
		SHIPS_CSST:units = "celsius"

	float SHIPS_CD20(nstorms, nmax_SHIPS_points)
		SHIPS_CD20:description = "Climatological depth of the 20 deg C isotherm at the analysis time, from 2005-2010 NCODA analyses"
		SHIPS_CD20:units = "meter"

	float SHIPS_CD26(nstorms, nmax_SHIPS_points)
		SHIPS_CD26:description = "Climatological depth of the 26 deg C isotherm at the analysis time, from 2005-2010 NCODA analyses"
		SHIPS_CD26:units = "meter"
	float SHIPS_COHC(nstorms, nmax_SHIPS_points)
		SHIPS_COHC:description = "Climatological ocean heat content at the analysis time, from 2005-2010 NCODA analyses"
		SHIPS_COHC:units = "kilojoule/centimeter^2"

	int SHIPS_DTL(nstorms, nmax_SHIPS_points)
		SHIPS_DTL:description = "Distance to nearest major land mass at the analysis time"
		SHIPS_DTL:units = "kilometer"

	float SHIPS_RSST(nstorms, nmax_SHIPS_points)
		SHIPS_RSST:description = "Reynolds SST at the analysis time"
		SHIPS_RSST:units = "celsius"

	int SHIPS_PHCN(nstorms, nmax_SHIPS_points)
		SHIPS_PHCN:description = "Estimated ocean heat content at the analysis time from climatological Ocean Heat Content (OHC) and current SST anomaly"
		SHIPS_PHCN:units = "kilojoule/centimeter^2"

	float SHIPS_U200(nstorms, nmax_SHIPS_points)
		SHIPS_U200:description = "200 hPa zonal wind, averaged over an annulus from r = 200 to 800 km, at the analysis time"
		SHIPS_U200:units = "knot"

	float SHIPS_U20C(nstorms, nmax_SHIPS_points)
		SHIPS_U20C:description = "200 hPa zonal wind, averaged over an circle from r = 0 to 500 km, at the analysis time"
		SHIPS_U20C:units = "knot"

	float SHIPS_V20C(nstorms, nmax_SHIPS_points)
		SHIPS_V20C:description = "200 hPa meridional wind, averaged over an circle from r = 0 to 500 km, at the analysis time"
		SHIPS_V20C:units = "knot"

	float SHIPS_E000(nstorms, nmax_SHIPS_points)
		SHIPS_E000:description = "1000 hPa theta_e, averaged over an annulus from r = 200 to 800 km, at the analysis time"
		SHIPS_E000:units = "kelvin"

	float SHIPS_EPOS(nstorms, nmax_SHIPS_points)
		SHIPS_EPOS:description = "The average theta_e difference between a parcel lifted from the surface and its environment, averaged over an annulus from r = 200 to 800 km, at the analysis time. Only positive differences are included in the average."
		SHIPS_EPOS:units = "celsius"
	float SHIPS_ENEG(nstorms, nmax_SHIPS_points)
		SHIPS_ENEG:description = "The average theta_e difference between a parcel lifted from the surface and its environment, averaged over an annulus from r = 200 to 800 km, at the analysis time. Only negative differences are included in the average. The minus sign is not included."
		SHIPS_ENEG:units = "celsius"

	float SHIPS_EPSS(nstorms, nmax_SHIPS_points)
		SHIPS_EPSS:description = "The average theta_e difference between a parcel lifted from the surface and the saturated theta_e of the environment, averaged over an annulus from r = 200 to 800 km, at the analysis time. Only positive differences are included in the average."
		SHIPS_EPSS:units = "celsius"

	float SHIPS_ENSS(nstorms, nmax_SHIPS_points)
		SHIPS_ENSS:description = "The average theta_e difference between a parcel lifted from the surface and the saturated theta_e of the environment, averaged over an annulus from r = 200 to 800 km, at the analysis time. Only negative differences are included in the average. The minus sign is not included."
		SHIPS_ENSS:units = "celsius"

	int SHIPS_RHLO(nstorms, nmax_SHIPS_points)
		SHIPS_RHLO:description = "850-700 hPa relative humidity, averaged over an annulus from r = 200 to 800 km, at the analysis time"
		SHIPS_RHLO:units = "percent"

	int SHIPS_RHMD(nstorms, nmax_SHIPS_points)
		SHIPS_RHMD:description = "700-500 hPa relative humidity, averaged over an annulus from r = 200 to 800 km, at the analysis time"
		SHIPS_RHMD:units = "percent"

	int SHIPS_RHHI(nstorms, nmax_SHIPS_points)
		SHIPS_RHHI:description = "500-300 hPa relative humidity, averaged over an annulus from r = 200 to 800 km, at the analysis time"
		SHIPS_RHHI:units = "percent"

	int SHIPS_PSLV(nstorms, nmax_SHIPS_points)
		SHIPS_PSLV:description = "Pressure of the center of mass of the layer where storm motion best matches environmental flow, at the analysis time"
		SHIPS_PSLV:units = "millibar"

	float SHIPS_Z850(nstorms, nmax_SHIPS_points)
		SHIPS_Z850:description = "850 hPa vorticity, averaged over a circle from r = 0 to 1000 km, at the analysis time"
		SHIPS_Z850:units = "1/second"
	float SHIPS_D200(nstorms, nmax_SHIPS_points)
		SHIPS_D200:description = "200 hPa divergence, averaged over a circle from r = 0 to 1000 km, at the analysis time"
		SHIPS_D200:units = "1/second"

	int SHIPS_REFC(nstorms, nmax_SHIPS_points)
		SHIPS_REFC:description = "Relative eddy momentum flux convergence, averaged over an annulus from r = 100 to 600 km, at the analysis time"
		SHIPS_REFC:units = "meter/(second.day)"

	int SHIPS_PEFC(nstorms, nmax_SHIPS_points)
		SHIPS_PEFC:description = "Planetary eddy momentum flux convergence, averaged over an annulus from r = 100 to 600 km, at the analysis time"
		SHIPS_PEFC:units = "meter/(second.day)"

	float SHIPS_T000(nstorms, nmax_SHIPS_points)
		SHIPS_T000:description = "1000 hPa temperature, averaged over an annulus from r = 200 to 800 km, at the analysis time"
		SHIPS_T000:units = "celsius"

	int SHIPS_R000(nstorms, nmax_SHIPS_points)
		SHIPS_R000:description = "1000 hPa relative humidity, averaged over an annulus from r = 200 to 800 km, at the analysis time"
		SHIPS_R000:units = "percent"

	int SHIPS_Z000(nstorms, nmax_SHIPS_points)
		SHIPS_Z000:description = "1000 hPa height deviation deviation from the U.S. standard atmosphere, at the analysis time"
		SHIPS_Z000:units = "meter"

	float SHIPS_TLAT(nstorms, nmax_SHIPS_points)
		SHIPS_TLAT:description = "Latitude of the 850 hPa vortex center in NCEP analysis, at the analysis time"
		SHIPS_TLAT:units = "degrees_north"

	float SHIPS_TLON(nstorms, nmax_SHIPS_points)
		SHIPS_TLON:description = "Longitude of the 850 hPa vortex center in NCEP analysis, at the analysis time"
		SHIPS_TLON:units = "degrees_east"

	float SHIPS_TWAC(nstorms, nmax_SHIPS_points)
		SHIPS_TWAC:description = "Symmetric tangential wind at 850 hPa from NCEP analysis, averaged over a circle from r = 0 to 600 km, at the analysis time"
		SHIPS_TWAC:units = "meter/second"

	float SHIPS_TWXC(nstorms, nmax_SHIPS_points)
```

```
                SHIPS_TWXC:description = "Maximum Symmetric tangential wind at 850 hPa
from NCEP analysis, at the analysis time"
                SHIPS_TWXC:units = "meter/second"

        float SHIPS_PENC(nstorms, nmax_SHIPS_points)
                SHIPS_PENC:description = "Azimuthally averaged surface pressure at outer
edge of vortex, at the analysis time"
                SHIPS_PENC:units = "hPascal"

        float SHIPS_SHDC(nstorms, nmax_SHIPS_points)
                SHIPS_SHDC:description = "850-200 hPa shear magnitude, averaged over a
circle from r = 0 to 500 km relative to the 850 hPa vortex center, at the analysis time"
                SHIPS_SHDC:units = "knot"

        int SHIPS_SDDC(nstorms, nmax_SHIPS_points)
                SHIPS_SDDC:description = "850-200 hPa shear direction, averaged over a circle
from r = 0 to 500 km relative to the 850 hPa vortex center, at the analysis time. Note: this is the
heading of the 'SHDC' shear vector."
                SHIPS_SDDC:units = "degree"

        float SHIPS_SHGC(nstorms, nmax_SHIPS_points)
                SHIPS_SHGC:description = "Generalized 850-200 hPa shear magnitude (takes
into account all levels), but with the vortex removed and averaged over a circle from r = 0 to 500
km relative to the 850 hPa vortex center, at the analysis time"
                SHIPS_SHGC:units = "knot"

        float SHIPS_DIVC(nstorms, nmax_SHIPS_points)
                SHIPS_DIVC:description = "200 hPa divergence, but centered at the 850 hPa
vortex location, at the analysis time"
                SHIPS_DIVC:units = "1/second"

        float SHIPS_T150(nstorms, nmax_SHIPS_points)
                SHIPS_T150:description = "150 hPa temperature, averaged over an annulus from
r = 200 to 800 km, at the analysis time"
                SHIPS_T150:units = "celsius"

        float SHIPS_T200(nstorms, nmax_SHIPS_points)
                SHIPS_T200:description = "200 hPa temperature, averaged over an annulus from
r = 200 to 800 km, at the analysis time"
                SHIPS_T200:units = "celsius"

        float SHIPS_T250(nstorms, nmax_SHIPS_points)
                SHIPS_T250:description = "250 hPa temperature, averaged over an annulus from
r = 200 to 800 km, at the analysis time"
                SHIPS_T250:units = "celsius"
```

float SHIPS_SHRD(nstorms, nmax_SHIPS_points)
SHIPS_SHRD:description = "850-200 hPa shear magnitude, averaged over an annulus from r = 200 to 800 km, at the analysis time"
SHIPS_SHRD:units = "knot"

int SHIPS_SHTD(nstorms, nmax_SHIPS_points)
SHIPS_SHTD:description = "850-200 hPa shear direction, averaged over an annulus from r = 200 to 800 km, at the analysis time. Note: this is the heading of the 'SHRD' shear vector."
SHIPS_SHTD:units = "degree"

float SHIPS_SHRS(nstorms, nmax_SHIPS_points)
SHIPS_SHRS:description = "850-500 hPa shear magnitude, averaged over an annulus from r = 200 to 800 km, at the analysis time"
SHIPS_SHRS:units = "knot"

int SHIPS_SHTS(nstorms, nmax_SHIPS_points)
SHIPS_SHTS:description = "850-500 hPa shear direction, averaged over an annulus from r = 200 to 800 km, at the analysis time. Note: this is the heading of the 'SHTS' shear vector."
SHIPS_SHTS:units = "degree"

float SHIPS_SHRG(nstorms, nmax_SHIPS_points)
SHIPS_SHRG:description = "Generalized 850-200 hPa shear magnitude (takes into account all levels), averaged over an annulus from r = 200 to 800 km, at the analysis time"
SHIPS_SHRG:units = "knot"

float SHIPS_PENV(nstorms, nmax_SHIPS_points)
SHIPS_PENV:description = "Surface pressure, averaged over an annulus from r = 200 to 800 km, at the analysis time"
SHIPS_PENV:units = "hPascal"

int SHIPS_VMPI(nstorms, nmax_SHIPS_points)
SHIPS_VMPI:description = "Maximum potential intensity from Kerry Emanuel equation"
SHIPS_VMPI:units = "knot"

float SHIPS_VVAV(nstorms, nmax_SHIPS_points)
SHIPS_VVAV:description = "Vertical velocity of a parcel lifted from the surface where entrainment, the ice phase, and the condensate weight are accounted for, averaged vertically from z = 0 to 15 km, at the analysis time. Note: moisture and temperature biases between the operational and reanlaysis files make this parameter inconsistent in the 2001-2007 sample, compared to 2000 and before."
SHIPS_VVAV:units = "meter/second"

float SHIPS_VMFX(nstorms, nmax_SHIPS_points)

SHIPS_VMFX:description = "Vertical velocity of a parcel lifted from the surface where entrainment, the ice phase, and the condensate weight are accounted for, taking a density weighted average from z = 0 to 15 km, at the analysis time. Note: moisture and temperature biases between the operational and reanlaysis files make this parameter inconsistent in the 2001-2007 sample, compared to 2000 and before. Note: this is the same as VVAV, except the vertical averaging uses a density weighted average."
SHIPS_VMFX:units = "meter/second"

float SHIPS_VVAC(nstorms, nmax_SHIPS_points)
SHIPS_VVAC:description = "Vertical velocity of a parcel lifted from the surface where entrainment, the ice phase, and the condensate weight are accounted for, but the GFS vortex is removed from soundings for r = 0 to 500 km, averaged vertically from z = 0 to 15 km, at the analysis time. Note: moisture and temperature biases between the operational and reanlaysis files make this parameter inconsistent in the 2001-2007 sample, compared to 2000 and before. Note: this is the same as VVAV, but with soundings from r = 0 to 500 km with the GFS vortex removed."
SHIPS_VVAC:units = "meter/second"

float SHIPS_GOES_IRXX_IR01(nstorms, nmax_SHIPS_points)
SHIPS_GOES_IRXX_IR01:description = "Time of the alternate predictors, relative to this case"
SHIPS_GOES_IRXX_IR01:units = "hour"

float SHIPS_GOES_IRXX_IR02(nstorms, nmax_SHIPS_points)
SHIPS_GOES_IRXX_IR02:description = "Alternative predictor (not satellite data) for the average GOES channel 4 brightness temperature, averaged over a circle from r = 0 to 200 km"
SHIPS_GOES_IRXX_IR02:units = "celsius"

float SHIPS_GOES_IRXX_IR03(nstorms, nmax_SHIPS_points)
SHIPS_GOES_IRXX_IR03:description = "Alternative predictor (not satellite data) for the standard deviation of GOES channel 4 brightness temperature, taken over a circle from r = 0 to 200 km"
SHIPS_GOES_IRXX_IR03:units = "celsius"

float SHIPS_GOES_IRXX_IR04(nstorms, nmax_SHIPS_points)
SHIPS_GOES_IRXX_IR04:description = "Alternative predictor (not satellite data) for the average GOES channel 4 brightness temperature, averaged over a circle from r = 100 to 300 km"
SHIPS_GOES_IRXX_IR04:units = "celsius"

float SHIPS_GOES_IRXX_IR05(nstorms, nmax_SHIPS_points)
SHIPS_GOES_IRXX_IR05:description = "Alternative predictor (not satellite data) for the standard deviation of GOES channel 4 brightness temperature, taken over a circle from r = 100 to 300 km"
SHIPS_GOES_IRXX_IR05:units = "celsius"

```
        int SHIPS_GOES_IRXX_IR06(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IRXX_IR06:description = "Alternative predictor (not satellite
data) for the percent area of GOES channel 4 brightness temperature < -10 C, for the annulus
from r = 50 to 200 km"
                SHIPS_GOES_IRXX_IR06:units = "percent"

        int SHIPS_GOES_IRXX_IR07(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IRXX_IR07:description = "Alternative predictor (not satellite
data) for the percent area of GOES channel 4 brightness temperature < -20 C, for the annulus
from r = 50 to 200 km"
                SHIPS_GOES_IRXX_IR07:units = "percent"

        int SHIPS_GOES_IRXX_IR08(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IRXX_IR08:description = "Alternative predictor (not satellite
data) for the percent area of GOES channel 4 brightness temperature < -30 C, for the annulus
from r = 50 to 200 km"
                SHIPS_GOES_IRXX_IR08:units = "percent"

        int SHIPS_GOES_IRXX_IR09(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IRXX_IR09:description = "Alternative predictor (not satellite
data) for the percent area of GOES channel 4 brightness temperature < -40 C, for the annulus
from r = 50 to 200 km"
                SHIPS_GOES_IRXX_IR09:units = "percent"

        int SHIPS_GOES_IRXX_IR10(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IRXX_IR10:description = "Alternative predictor (not satellite
data) for the percent area of GOES channel 4 brightness temperature < -50 C, for the annulus
from r = 50 to 200 km"
                SHIPS_GOES_IRXX_IR10:units = "percent"

        int SHIPS_GOES_IRXX_IR11(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IRXX_IR11:description = "Alternative predictor (not satellite
data) for the percent area of GOES channel 4 brightness temperature < -60 C, for the annulus
from r = 50 to 200 km"
                SHIPS_GOES_IRXX_IR11:units = "percent"

        float SHIPS_GOES_IRXX_IR12(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IRXX_IR12:description = "Alternative predictor (not satellite
data) for the maximum GOES channel 4 brightness temperature in the circle from r = 0 to 30
km"
                SHIPS_GOES_IRXX_IR12:units = "celsius"

        float SHIPS_GOES_IRXX_IR13(nstorms, nmax_SHIPS_points)
```

```
                SHIPS_GOES_IRXX_IR13:description = "Alternative predictor (not satellite
data) for the average GOES channel 4 brightness temperature, averaged over the circle from r =
0 to 30 km"
                SHIPS_GOES_IRXX_IR13:units = "celsius"

        int SHIPS_GOES_IRXX_IR14(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IRXX_IR14:description = "Alternate predictor (not satellite data)
for the radius of the maximum GOES channel 4 brightness temperature"
                SHIPS_GOES_IRXX_IR14:units = "kilometer"

        float SHIPS_GOES_IRXX_IR15(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IRXX_IR15:description = "Alternative predictor (not satellite
data) for the minimum GOES channel 4 brightness temperature in the annulus r = 20 to 120 km"
                SHIPS_GOES_IRXX_IR15:units = "celsius"

        float SHIPS_GOES_IRXX_IR16(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IRXX_IR16:description = "Alternative predictor (not satellite
data) for the average GOES channel 4 brightness temperature, averaged over the annulus r = 20
to 120 km"
                SHIPS_GOES_IRXX_IR16:units = "celsius"

        int SHIPS_GOES_IRXX_IR17(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IRXX_IR17:description = "Alternate predictor (not satellite data)
for the radius of the minimum GOES channel 4 brightness temperature"
                SHIPS_GOES_IRXX_IR17:units = "kilometer"

        float SHIPS_GOES_IR00_IR01(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IR00_IR01:description = "Time of the predictors, relative to this
case"
                SHIPS_GOES_IR00_IR01:units = "hour"

        float SHIPS_GOES_IR00_IR02(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IR00_IR02:description = "Average GOES channel 4 brightness
temperature, averaged over a circle from r = 0 to 200 km, from near the initial time of this case"
                SHIPS_GOES_IR00_IR02:units = "celsius"

        float SHIPS_GOES_IR00_IR03(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IR00_IR03:description = "Standard deviation of GOES channel 4
brightness temperature, taken over a circle from r = 0 to 200 km, from near the initial time of this
case"
                SHIPS_GOES_IR00_IR03:units = "celsius"

        float SHIPS_GOES_IR00_IR04(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IR00_IR04:description = "Average GOES channel 4 brightness
temperature, averaged over a circle from r = 100 to 300 km, from near the initial time of this
case"
```

SHIPS_GOES_IR00_IR04:units = "celsius"

float SHIPS_GOES_IR00_IR05(nstorms, nmax_SHIPS_points)
SHIPS_GOES_IR00_IR05:description = "Standard deviation of GOES channel 4 brightness temperature, taken over a circle from r = 100 to 300 km, from near the initial time of this case"
SHIPS_GOES_IR00_IR05:units = "celsius"

int SHIPS_GOES_IR00_IR06(nstorms, nmax_SHIPS_points)
SHIPS_GOES_IR00_IR06:description = "Percent area of GOES channel 4 brightness temperature < -10 C, for the annulus from r = 50 to 200 km, from near the initial time of this case"
SHIPS_GOES_IR00_IR06:units = "percent"

int SHIPS_GOES_IR00_IR07(nstorms, nmax_SHIPS_points)
SHIPS_GOES_IR00_IR07:description = "Percent area of GOES channel 4 brightness temperature < -20 C, for the annulus from r = 50 to 200 km, from near the initial time of this case"
SHIPS_GOES_IR00_IR07:units = "percent"

int SHIPS_GOES_IR00_IR08(nstorms, nmax_SHIPS_points)
SHIPS_GOES_IR00_IR08:description = "Percent area of GOES channel 4 brightness temperature < -30 C, for the annulus from r = 50 to 200 km, from near the initial time of this case"
SHIPS_GOES_IR00_IR08:units = "percent"

int SHIPS_GOES_IR00_IR09(nstorms, nmax_SHIPS_points)
SHIPS_GOES_IR00_IR09:description = "Percent area of GOES channel 4 brightness temperature < -40 C, for the annulus from r = 50 to 200 km, from near the initial time of this case"
SHIPS_GOES_IR00_IR09:units = "percent"

int SHIPS_GOES_IR00_IR10(nstorms, nmax_SHIPS_points)
SHIPS_GOES_IR00_IR10:description = "Percent area of GOES channel 4 brightness temperature < -50 C, for the annulus from r = 50 to 200 km, from near the initial time of this case"
SHIPS_GOES_IR00_IR10:units = "percent"

int SHIPS_GOES_IR00_IR11(nstorms, nmax_SHIPS_points)
SHIPS_GOES_IR00_IR11:description = "Percent area of GOES channel 4 brightness temperature < -60 C, for the annulus from r = 50 to 200 km, from near the initial time of this case"
SHIPS_GOES_IR00_IR11:units = "percent"

float SHIPS_GOES_IR00_IR12(nstorms, nmax_SHIPS_points)

```
                SHIPS_GOES_IR00_IR12:description = "Maximum GOES channel 4 brightness
temperature in the circle from r = 0 to 30 km, from near the initial time of this case"
                SHIPS_GOES_IR00_IR12:units = "celsius"

        float SHIPS_GOES_IR00_IR13(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IR00_IR13:description = "Average GOES channel 4 brightness
temperature, averaged over the circle from r = 0 to 30 km, from near the initial time of this case"
                SHIPS_GOES_IR00_IR13:units = "celsius"

        int SHIPS_GOES_IR00_IR14(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IR00_IR14:description = "Radius of the maximum GOES
channel 4 brightness temperature, from near the initial time of this case"
                SHIPS_GOES_IR00_IR14:units = "kilometer"

        float SHIPS_GOES_IR00_IR15(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IR00_IR15:description = "Minimum GOES channel 4 brightness
temperature in the annulus r = 20 to 120 km, from near the initial time of this case"
                SHIPS_GOES_IR00_IR15:units = "celsius"

        float SHIPS_GOES_IR00_IR16(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IR00_IR16:description = "Average GOES channel 4 brightness
temperature, averaged over the annulus r = 20 to 120 km, from near the initial time of this case"
                SHIPS_GOES_IR00_IR16:units = "celsius"

        int SHIPS_GOES_IR00_IR17(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IR00_IR17:description = "Radius of the minimum GOES channel
4 brightness temperature, from near the initial time of this case"
                SHIPS_GOES_IR00_IR17:units = "kilometer"

        float SHIPS_GOES_IRM3_IR01(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IRM3_IR01:description = "Time of the predictors, relative to this
case"
                SHIPS_GOES_IRM3_IR01:units = "hour"

        float SHIPS_GOES_IRM3_IR02(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IRM3_IR02:description = "Average GOES channel 4 brightness
temperature, averaged over a circle from r = 0 to 200 km, from three hours before the initial time
of this case"
                SHIPS_GOES_IRM3_IR02:units = "celsius"

        float SHIPS_GOES_IRM3_IR03(nstorms, nmax_SHIPS_points)
                SHIPS_GOES_IRM3_IR03:description = "Standard deviation of GOES channel
4 brightness temperature, taken over a circle from r = 0 to 200 km, from three hours before the
initial time of this case"
                SHIPS_GOES_IRM3_IR03:units = "celsius"
```

float SHIPS_GOES_IRM3_IR04(nstorms, nmax_SHIPS_points)

SHIPS_GOES_IRM3_IR04:description = "Average GOES channel 4 brightness temperature, averaged over a circle from r = 100 to 300 km, from three hours before the initial time of this case"

SHIPS_GOES_IRM3_IR04:units = "celsius"

float SHIPS_GOES_IRM3_IR05(nstorms, nmax_SHIPS_points)

SHIPS_GOES_IRM3_IR05:description = "Standard deviation of GOES channel 4 brightness temperature, taken over a circle from r = 100 to 300 km, from three hours before the initial time of this case"

SHIPS_GOES_IRM3_IR05:units = "celsius"

int SHIPS_GOES_IRM3_IR06(nstorms, nmax_SHIPS_points)

SHIPS_GOES_IRM3_IR06:description = "Percent area of GOES channel 4 brightness temperature < -10 C, for the annulus from r = 50 to 200 km, from three hours before the initial time of this case"

SHIPS_GOES_IRM3_IR06:units = "percent"

int SHIPS_GOES_IRM3_IR07(nstorms, nmax_SHIPS_points)

SHIPS_GOES_IRM3_IR07:description = "Percent area of GOES channel 4 brightness temperature < -20 C, for the annulus from r = 50 to 200 km, from three hours before the initial time of this case"

SHIPS_GOES_IRM3_IR07:units = "percent"

int SHIPS_GOES_IRM3_IR08(nstorms, nmax_SHIPS_points)

SHIPS_GOES_IRM3_IR08:description = "Percent area of GOES channel 4 brightness temperature < -30 C, for the annulus from r = 50 to 200 km, from three hours before the initial time of this case"

SHIPS_GOES_IRM3_IR08:units = "percent"

int SHIPS_GOES_IRM3_IR09(nstorms, nmax_SHIPS_points)

SHIPS_GOES_IRM3_IR09:description = "Percent area of GOES channel 4 brightness temperature < -40 C, for the annulus from r = 50 to 200 km, from three hours before the initial time of this case"

SHIPS_GOES_IRM3_IR09:units = "percent"

int SHIPS_GOES_IRM3_IR10(nstorms, nmax_SHIPS_points)

SHIPS_GOES_IRM3_IR10:description = "Percent area of GOES channel 4 brightness temperature < -50 C, for the annulus from r = 50 to 200 km, from three hours before the initial time of this case"

SHIPS_GOES_IRM3_IR10:units = "percent"

int SHIPS_GOES_IRM3_IR11(nstorms, nmax_SHIPS_points)

SHIPS_GOES_IRM3_IR11:description = "Percent area of GOES channel 4 brightness temperature < -60 C, for the annulus from r = 50 to 200 km, from three hours before the initial time of this case"

SHIPS_GOES_IRM3_IR11:units = "percent"

float SHIPS_GOES_IRM3_IR12(nstorms, nmax_SHIPS_points)

SHIPS_GOES_IRM3_IR12:description = "Maximum GOES channel 4 brightness temperature in the circle from r = 0 to 30 km, from three hours before the initial time of this case"

SHIPS_GOES_IRM3_IR12:units = "celsius"

float SHIPS_GOES_IRM3_IR13(nstorms, nmax_SHIPS_points)

SHIPS_GOES_IRM3_IR13:description = "Average GOES channel 4 brightness temperature, averaged over the circle from r = 0 to 30 km, from three hours before the initial time of this case"

SHIPS_GOES_IRM3_IR13:units = "celsius"

int SHIPS_GOES_IRM3_IR14(nstorms, nmax_SHIPS_points)

SHIPS_GOES_IRM3_IR14:description = "Radius of the maximum GOES channel 4 brightness temperature, from three hours before the initial time of this case"

SHIPS_GOES_IRM3_IR14:units = "kilometer"

float SHIPS_GOES_IRM3_IR15(nstorms, nmax_SHIPS_points)

SHIPS_GOES_IRM3_IR15:description = "Minimum GOES channel 4 brightness temperature in the annulus r = 20 to 120 km, from three hours before the initial time of this case"

SHIPS_GOES_IRM3_IR15:units = "celsius"

float SHIPS_GOES_IRM3_IR16(nstorms, nmax_SHIPS_points)

SHIPS_GOES_IRM3_IR16:description = "Average GOES channel 4 brightness temperature, averaged over the annulus r = 20 to 120 km, from three hours before the initial time of this case"

SHIPS_GOES_IRM3_IR16:units = "celsius"

int SHIPS_GOES_IRM3_IR17(nstorms, nmax_SHIPS_points)

SHIPS_GOES_IRM3_IR17:description = "Radius of the minimum GOES channel 4 brightness temperature, from three hours before the initial time of this case"

SHIPS_GOES_IRM3_IR17:units = "kilometer"

EBT Storm Description Parameters

int EB_datapresent(nstorms)

EB_datapresent:units = ""

EB_datapresent:description = "A boolean that indicates whether Extended Best Track data is available for this storm. If set to 'False', the BT_xxx parameters will contain only what was included in the Best Track dataset, with no supplementation from the Extended Best Track dataset."

Errors and Questions

If you find any errors or have any questions regarding the data or methodology, please kindly send an email to Jonathan: jvigh@ucar.edu.

Acknowledgements

All residents of hurricane-prone areas in the U.S. owe an enormous debt to the brave flight crews of the 53rd Weather Reconnaissance Squadron and NOAA's Aircraft Operation Center who put themselves at risk each time they collect these vital data. Without their dedication and diligence, this data set would not have been possible. I first thank my former graduate advisor, Wayne Schubert, for his patience, support, and encouragement through this lengthy project. I thank Steve Feuer, A. Barry Damiano, John Pavone, Chris Sisko, Christopher Juckins, Mark Zimmer, Christopher Landsea, and Neal Dorst for their assistance in obtaining the many VDM messages. They, along with Eric Blake, Jonathan Talbot, Jack Parrish, and Nicholas Carrasco, graciously answered my many questions about the VDM contents, history, and usage. Buck Sampson and James Franklin provided information about the ATCF b-decks. I thank Mark DeMaria for allowing this present dataset to include substantially all of his Extended Best Track and Statistical Hurricane Intensity Prediction Scheme datasets. Mary Haley, David Brown, and others on the NCL Development Team provided very helpful programming advice. Christopher Williams provided tedious and detailed checks on an earlier version of the manuscript. Daniel Stern provided insightful comments that led to important improvements to the metadata and documentation.

The initial compilation of this dataset was accomplished at Colorado State University with support from NASA/TCSP Grant NNG06GA54G and by NSF Grants ATM-0332197 and ATM-0837932. The author gratefully acknowledges additional support from the Advanced Study Program and the NCAR Earth System Laboratory, which enabled further improvements to be made to the dataset. The present phase of the work, including finalization of the documentation and preparation of the dataset for public release, was supported by a grant from Risk Prediction Initiative of the Bermuda Institute of Ocean Sciences. The views expressed herein do not necessarily reflect the views of any of these organizations. NCAR is sponsored by the National Science Foundation.