NMRF/TR/05/2020

TECHNICAL REPORT

Generation of TC-Vitals at NCMRWF

**Srinivas Desamsetti, Sanjeev Kumar Singh,
C.J. Johny and V.S. Prasad**

July 2020

**National Centre for Medium Range Weather Forecasting
Ministry of Earth Sciences, Government of India
A-50, Sector-62, Noida-201 309, INDIA**

Generation of TC-Vitals at NCMRWF

Srinivas Desamsetti, Sanjeev Kumar Singh,
C.J. Johny and V.S. Prasad

July 2020

National Centre for Medium Range Weather Forecasting
Ministry of Earth Sciences
A-50, Sector 62, Noida-201309, INDIA

Ministry of Earth Sciences
National Centre for Medium Range Weather Forecasting
Document Control Data Sheet

1	Name of the Institute	National Centre for Medium Range Weather Forecasting
2	Document Number	NMRF/TR/05/2020
3	Date of publication	July, 2020
4	Title of the document	Generation of TC-Vitals at NCMRWF
5	Type of Document	Technical Report
6	No. of Pages, Figures and Tables	38 pages, 1 Tables, 4 Annexures
7	Number of References	5
8	Author (S)	Srinivas Desamsetti, Sanjeev Kumar Singh, C.J. Johny and V.S. Prasad
9	Originating Unit	NCMRWF
10	Abstract	Tropical cyclone relocation in the global models is one of the important tasks of numerical modeling system. The details of cyclone records are named as Tropical Cyclone Vital Statistics Records ("TC-Vitals"). TC-Vitals is a database of vitals which describes the tropical cyclones, post-tropical cyclones and potential formation of tropical cyclones. Mainly the database focusses on the storms location, intensity, movement, type, structure, name, forming basin and other related information. This database is a seven-bit ASCII text file contains each storm information in a single line (record) corresponding to one valid time hour. These records generally generated at six-hour interval. The TC-Vitals contains records of storms which had genesis over all the oceans. The details of the generation of these TC-Vitals records in near real-time at NCMRWF are provided in this document. The scripts developed at NCMRWF for the generation of TC-Vitals are provided in the Annexures.
11	Security classification	Non-Secure
12	Distribution	Unrestricted Distribution
13	Keywords	GTS, Tropical Cyclones, WMO, NHC, JTWC, decoding

Table of Contents

Sl. No.	Topic	Page No.
1	Abstract	5
2	Introduction	6
3	Methodology	8
4	Format of TC-Vitals	9
5	Acknowledgements	13
6	References	13
7	Annexure-1	14
8	Annexure-2	20
9	Annexure-3	26
10	Annexure-4	32

Abstract

Tropical cyclone relocation in the global models is one of the important tasks of numerical modeling system. The relocation enables fixing the cyclone center at the correct location as observed, and subsequently leads to better prediction of the movement and intensification. The global operational models require the position of cyclone/ hurricane/ typhoon in real time. These records are generated by using the Tropical Cyclone bulletins received through Global Teleconnection System (GTS) data archive at every six-hour interval. These records are named as Tropical Cyclone Vital Statistics Records ("TC-Vitals"). TC-Vitals is a database of vitals which describes the tropical cyclones, post-tropical cyclones and potential formation of tropical cyclones. Mainly the database focusses on the storms location, intensity, movement, type, structure, name, forming basin and other related information. This database is a seven-bit ASCII text file contains each storm information in a single line (record) corresponding to one valid time hour. These records generally generated at six-hour interval. The TC-Vitals contains records of storms which had genesis over all the oceans. These TC-Vitals are very essential for vortex relocation in the global and as well as in the regional models and used for studying the global cyclones characteristics. The details of the generation of these TC-Vitals records in near real-time at NCMRWF are provided in this document. The scripts developed at NCMRWF for the generation of TC-Vitals are provided in the annexures.

1. Introduction

Tropical cyclones (hurricanes or typhoons) form over world oceans during summer season both in the Northern and Southern hemispheres between 30S-30N. Prediction of the movement and intensification of these storms is crucial for administrators to prevent the damage to properties and reduce the casualties. Numerical models are the best tools for predicting the track and intensification of these storms with three (72 hours) to five days (120 hours) lead time. The numerical models need the best state of atmosphere and ocean and lead to better forecast. Even 1% error in the initial model integration time, may lead to the grown of error by 10% in 72 hours, and the error grows thereafter linearly. So, it is very important to provide the best state of initial position (latitude, longitude) of the tropical storm. National Centre for Medium Range Weather Forecasting (NCMRWF) Global Forecasting System (NGFS), is a state of numerical global model, and is used in operational mode to generate the analysis each day for four cycles (Rajagopal et al., 2007; Prasad et al., 2011). This analysis is being used by India Meteorological Department (IMD) for generating the forecasts at short, medium, and extended range weather forecasts. These analyses also used in the generation of the ensemble members and subsequently the ensemble forecasts by IMD. NCMRWF receives all types of global observations in near real time through Global Telecommunications System (GTS) and other satellite data centers (MOSDAC, EUMETDAT, NRSC etc.). The tropical cyclone bulletins are received in ASCII format and being decoded with in-house developed shell scripts and FORTAN codes. The information related to tropical cyclones like the cyclone center (latitude, longitude), intensity (maximum winds, minimum central sea level pressure), structure (radius of maximum winds of four quadrants at 34, 50, 64 knot wind speeds) and speed and movement of cyclone system (storm speed and direction) are given in a single line record valid for one particular hour, and these records are called Tropical Cyclone Vital Statistics Records ("TC-VITALS"). TC-Vitals are generated if system in the global oceans attain intensity of tropical storm. These TC-Vitals are generated every day for four cycles at the operational NWP centers for their use in the global and regional operational models. The database is a seven-bit ASCII text file with all cyclones that are occurring in all the tropical oceans at six-hour time intervals.

The present document describes the generation of TC-Vitals from GTS bulletins. These TC-Vitals are also used as observations database and being used for studying cyclones globally (Trahan and Sparling, 2012). The description of data is provided in section 2, TC-Vitals generation methodology is given in section 3, section 4 describes the format of TC-Vitals file and references are provided in section 6.

2. Data

The meteorological observations from all over the globe from various observing platforms are received at Regional Telecommunication Hub (RTH), New Delhi through Global telecommunication System (GTS) and the same is made available to NCMRWF through a dedicated link called National knowledge Network (NKN) (Prasad et al., 2011). Prasad (2020) briefly describes about the GTS data reception and processing at NCMRWF. The GTS bulletins for tropical cyclones are in two formats.

ASCII and TAC. The header starts with WT, which stands for world Tropical storm. WMO designated six Regional Specialized Meteorological Centres (RSMCs) located in Honolulu, La Reunion, Miami, Nadi (Fiji), New Delhi and Tokyo, and other National Meteorological Services (NMSs) to provide the forecasts on the behaviour of tropical cyclones like position, size and shape of eye, intensity (maximum wind speed and central surface pressure), direction and speed of movement, and associated gusty winds, flash floods and storm surge (WMO, 2019). There are six meteorological centres and five regional Tropical Cyclone Warning Centres play important role in naming of tropical cyclones and the distribution of tropical cyclone advisories and warnings. The six meteorological centres are as follows:

i. Southwest Pacific Ocean: RSMC Nadi-Tropical Cyclone Centre - Fiji Meteorological Service (Nadi, Fiji)
ii. Southwest Indian Ocean: RSMC La Reunion-Tropical Cyclone Centre/Météo France (Réunion island, French Overseas Department)
iii. Bay of Bengal and Arabian Sea: RSMC - Tropical Cyclones New Delhi/India Meteorological Department (New Delhi, India)
iv. Western North Pacific Ocean and South China Sea - RSMC Tokyo/Japan Meteorological Agency (Tokyo, Japan)
v. Central North Pacific Ocean - RSMC Honolulu Central Pacific Hurricane Center (Honolulu, Hawaii, USA)
vi. Northeast Pacific Ocean, Gulf of Mexico, Caribbean Sea, and north Atlantic Ocean - RSMC Miami / National Hurricane Center

The GTS bulletin shown below is from IMD, New Delhi for the AMPHAN super cyclone which formed over Bay of Bengal region, valid at 0600 UTC of 19 May 2020.

WTIN31 DEMS 190600
QUADRANT WIND DISTRIBUTION IN ASSOCIATION WITH SUPER CYCLONIC
STORM AMPHAN OVER BAY OF BENGAL BASED ON 0000 UTC OF 19TH MAY,
2020 FOR WHICH FORECAST IS PREPARED:
PRESENT DATE AND TIME: 190000 UTC
PRESENT POSITION: 15.6 0N/86.70E
POSITION ACCURATE TO 20 KM
PRESENT MOVEMENT (DDD/FF) PAST SIX HOURS: 020/07 KT
PRESENT WIND DISTRIBUTION:
MAX SUSTAINED WINDS: 120 KT, GUSTS 135 KT
RADIUS OF MAXIMUM WIND 25 NM
WINDS VARY IN EACH QUADRANT
RADII ARE LARGEST RADII EXPECTED ANYWHERE IN THE QUADRANT
WIND RADII VALID OVER OPEN WATER ONLY
FORECASTS:
06 HRS, VALID AT:
190600Z 16.8N /87.00 E
MAX SUSTAINED WINDS: 110 KT, GUSTS 120 KT
RADIUS OF 027 KT WINDS:
170 NM NORTHEAST QUADRANT
160 NM SOUTHEAST QUADRANT
140 NM SOUTHWEST QUADRANT
160 NM NORTHWEST QUADRANT
RADIUS OF 034 KT WINDS:

150 NM NORTHEAST QUADRANT
140 NM SOUTHEAST QUADRANT
120 NM SOUTHWEST QUADRANT
130 NM NORTHWEST QUADRANT
RADIUS OF 050 KT WINDS:
70 NM NORTHEAST QUADRANT
70 NM SOUTHEAST QUADRANT
60 NM SOUTHWEST QUADRANT
60NM NORTHWEST QUADRANT
RADIUS OF 064 KT WINDS:
40 NM NORTHEAST QUADRANT
40 NM SOUTHEAST QUADRANT
30 NM SOUTHWEST QUADRANT
35 NM NORTHWEST QUADRANT

3. Methodology

This section briefly describes on the decoding GTS data and generation of TC-Vitals at NCMRWF. As a first step, the data for tropical cyclones/ hurricanes/typhoons is extracted from GTS data archive with header names of WT, TCAC, SXXT50. After archiving the GTS bulletins of tropical cyclones over the globe, the TC-Vitals related information is extracted using shell scripts. The variables extracted mainly are name of the storm, id of storm, organization id, date, hour, storm central latitude and longitude, storm direction degrees from north, storm speed (knots), central sea level pressure (mb), environment pressure (mb), estimated radius of outer most closed isobar (ROCI), i.e., size of the storm circulation (km), estimated max wind speed (kt), estimated radius of maximum wind (km), radius of maximum winds in the north-east, south-east, south-west, and north-west quadrants with 34 knot wind speed radius, similarly with 50kt wind speed the four quadrant parameters, additionally the estimated pressure of top of cyclone circulation (mb), latitude, longitude, with maximum forecast time of 72-hours. These unavailable parameters are given undefined values. These values are given as input to the "syndat" program. This syndat program generates the TC-Vitals file containing unique information record for each storm. The details of theTC-Vitals file are given in section 4. For the extraction of data from GTS bulletins separate shell scripts are developed at NCMRWF. There are mainly four scripts that extract storm information from different centers like NHC, JTWC, EGRR and others (PGTW, PHNC, WSSS, DEMS, BABJ, RJTD, RKSL, NFFN, PGUM, RPMM, PHFO). The shell script being used to extract storm data from EGRR data is provided in Annexure-I. Similarly, from NHC in Annexure-II, from JTWC in Annexure-III, and from remaining centres are given in Annexure-IV. After running these scripts, we will have the input files with storm name containing the one storm details for one file. These input files are later supplied as input to "syndat" program of GFS system, which creates TC-Vitals file valid for the day and a cycle (hour). These files are crucial for global models (e.g. NGFS, NCUM) and regional models (e.g., WRF, HWRF, UM Regional etc.) for relocating the storm in the model analysis with observed location and/ or strength. These TC-Vitals files are generated at NCMRWF eight times daily before running the analysis (both early run and update run). The NCMRWF analysis is made every day with cyclone/hurricane reentering by using these TC-Vitals. These TC-Vitals are being used for cyclone vortex initialization in NCUM-Regional (NCUM-R). These TC-

Vitals are also being used as observations for cyclone track plotting from NEPS-G, NCUM-G and NCUM-R models in real time.

4. Format of TC-Vitals

This section briefly describes the general format of TC-Vitals file, which contains the tropical cyclone/hurricane/typhoon information as a single record for each storm valid for a day and for the hour (https://www.emc.ncep.noaa.gov/mmb/data_processing/TC-Vitals_description.htm). The first column of the TC-Vitals record shows the organization ID of TC bulletin like NHC (National Hurricane Center, USA), JTWC (Joint Typhoon Warning Center, USA), IMD (India Meteorological Department, New Delhi). The second column is the 8-character storm identification code which uniquely recognizes the storm by the storm number and its origin basin (L - North Atlantic, E - North East Pacific, C - North Central Pacific, W - North West Pacific, B - Bay of Bengal, A - Arabian Sea, Q - South Atlantic, P - South Pacific, S - South Indian Ocean, W - East/South China Sea, S/P - Australian). The third column represents the name of the storm (given after attaining the tropical cyclone intensity) by respective RMSCs. The fourth column is the time of the observation (hour in UTC). The fifth (latitude) and sixth columns (longitude) represent the center of the storm with N/S (North/South), E/W (East/West) and the location is given in decimal form to the nearest tenth of a degree. The columns seventh (direction (degrees)) and eighth (speed (m/s)) indicate the storm motion. The ninth and twelfth columns represent intensity of the storm as central pressure (mbar) and storm maximum wind speed (m/s) respectively. The full description of the TC-Vitals record is presented in Table 1. TC-Vitals file valid on 19 May 2020 for four (00, 06, 12 and 18) cycles are shown below.

NHC 01L ARTHUR 20200519 0000 362N 0732W 050 072 0993 1012 0297 23 093 0185 0185 -999 0148 M
IMD 01B AMPHAN 20200519 0000 156N 0867E 041 045 0940 1002 0400 41 040 0280 0260 0220 0240 D
NHC 01L ARTHUR 20200519 0600 368N 0715W 055 072 0991 1012 0297 26 065 0222 0204 0204 0167 M
IMD 01B AMPHAN 20200519 0600 165N 0869E 010 050 0950 1000 0500 58 035 0300 0280 0240 0260 D
NHC 01L ARTHUR 20200519 1200 369N 0695W -99 -99 0991 -999 -999 26 -99 -999 -999 -999 -999 D
IMD 01B AMPHAN 20200519 1200 174N 0870E 006 045 0950 1000 0500 58 035 0320 0260 0220 0240 D
NHC 01L ARTHUR 20200519 1800 369N 0677W 080 077 0989 1012 0297 28 074 0334 0222 0204 0389 M
IMD 01B AMPHAN 20200519 1800 184N 0871E 016 070 0950 1000 0500 58 035 0320 0260 0220 0240 D

Table 1: Details of TC-Vitals file

Sl. No.	Byte(s)	Meaning	Explanation
1.	1-4	Organization ID	Left-justified organization identifier, maximum of four characters. **NHC** U.S. National Hurricane Center (NOAA/NWS/NCEP/NHC) **JTWC** U.S. Navy Joint Typhoon Warning Center
2.	*5*	*blank space*	*Single space (ASCII character 32).*
3.	6-7	Storm ID in this basin	**01-49** Formed or dissipating storms. **50-79** NHC/JTWC internal use only. **80-89** Testing only: fake storms for testing dataflow. **90-99** Genesis/investigative case.
4.	8	Basin identifier	The following values are presently used and understood by the NCEP suite. Attempting to use any other basin identifiers may cause the NCEP models to fail. **L** North Atlantic **E** North East Pacific **C** North Central Pacific **W** North West Pacific **B** Bay of Bengal (North Indian Ocean) **A** Arabian Sea (North Indian Ocean) **Q** South Atlantic **P** South Pacific **S** South Indian Ocean The following basins are not supported and have never been used in the NCEP TC-Vitals: **O** South China Sea (use W instead) **T** East China Sea (use W instead) **U** "Australian" (use S or P instead)
5.	*9*	*blank space*	*Single space (ASCII character 32).*
6.	10-18	Storm name	Names must be left-justified, begin with a capital letter, and may contain only capital English letters and hyphens ("-"). If the name is less than nine characters long, it must be left-justified: bytes to the right must be blank space (ASCII character 32). There are several special names recognized: **INVEST** investigative/genesis case: a storm that might form **TEST** not a real storm; just for testing dataflow **NAMELESS** no name yet assigned
7.	*19*	*blank space*	*Single space (ASCII character 32).* *Reused by SYNDAT_QCTROPCY:* ***: (colon)*** *first occurrence of a record for this storm* ***Blank*** *second or more occurrence of a record for this storm*
8.	20-23	Year	Year at which the data is valid. (YYYY format)
9.	24-25	Month	01-12 for January-December (MM format)
10.	26-27	Day	Day of month, zero-padded: 01-31 (DD format).
11.	*28*	*blank space*	*Single space (ASCII character 32).*
12.	29-30	Hour	Hour of day, zero-padded. e.g.06 for 06 UTC.
13.	31-32	Minute	Minute of hour, zero-padded. e.g. 00.
14.	*33*	*blank space*	*Single space (ASCII character 32).*
15.	34-36	latitude (decidegrees)	Storm latitude in tenths of a degree, always positive
16.	37	North/South flag	**N** North **S** South
17.	*38*	*blank space*	*Single space (ASCII character 32).*
18.	39-42	longitude (decidegrees)	Storm longitude in tenths of a degree, always positive

19.	43	East/West flag	**E** East **W** West
20.	*44*	*blank space*	*Single space (ASCII character 32).* *Reused by SYNDAT_QCTROPCY for storm motion direction data source:* ***C*** *obtained via climatology* ***P*** *obtained via persistence*
21.	45-47	Storm motion direction (degrees)	Degrees from north 0-360 (0=North, 90=East, 180=South, 270=West)
22.	*48*	*blank space*	*Single space (ASCII character 32).* *Reused by SYNDAT_QCTROPCY for storm motion speed data source:* ***C*** *obtained via climatology* ***P*** *obtained via persistence*
23.	49-51	Storm motion speed (dm/s)	Storm motion speed in specified direction. Always positive in tenths of a meter per second.
24.	*52*	*blank space*	*Single space (ASCII character 32).* *Reused by SYNDAT_QCTROPCY for central pressure data source:* ***C*** *obtained via climatology* ***P*** *obtained via persistence* ***Z*** *central pressure is set to reported environmental pressure value and environmental pressure is set to reported central pressure value (they are switched)*
25.	53-56	Storm central pressure (mbar)	Storm central pressure in mbar (hPa).
26.	*57*	*blank space*	*Single space (ASCII character 32).* *Reused by SYNDAT_QCTROPCY for environmental pressure data source:* ***C*** *obtained via climatology* ***P*** *obtained via persistence* ***Z*** *environmental pressure is set to reported central pressure value and central pressure is set to reported environmental pressure value (they are switched)*
27.	58-61	Storm environmental pressure (mbar)	Storm environmental pressure in mbar (hPa).
28.	*62*	*blank space*	*Single space (ASCII character 32).* *Reused by SYNDAT_QCTROPCY for data source of the radius of the outermost closed isobar:* ***C*** *obtained via climatology* ***P*** *obtained via persistence*
29.	63-66	Radius of outermost closed isobar (km)	Estimated radius of outermost closed isobar (i.e., storm circulation) in kilometers.
30.	*67*	*blank space*	*Single space (ASCII character 32).* *Reused by SYNDAT_QCTROPCY for the maximum wind data source:* ***C*** *obtained via climatology* ***P*** *obtained via persistence*
31.	68-69	Maximum wind speed (m/s)	Estimated storm-wide maximum one minute sustained ten meter altitude wind speed in meters per second
32.	*70*	*blank space*	*Single space (ASCII character 32).*
33.	71-73	Radius of maximum wind (km)	Estimated radius at which the maximum wind occurs in kilometers from the storm center
34.	*74*	*blank space*	*Single space (ASCII character 32).*
35.	75-78	Radius of 34kt	**34 Knot Wind Radii**

		winds in NE quadrant (km)	Estimated outermost radius in kilometers at which 34 knot winds occur in the storm in each of four quadrants. The quadrants are defined as: NW NE SW SE Missing values should be filled in as -999.
36.	*79*	*blank space (ASCII character 32)*	
37.	80-83	Radius of 34kt winds in SE quadrant (km)	-999 = no estimate, missing
38.	*84*	*blank space (ASCII character 32)*	
39.	85-88	Radius of 34kt winds in SW quadrant (km)	-999 = no estimate, missing
40.	*89*	*blank space (ASCII character 32)*	
41.	90-93	Radius of 34kt winds in NW quadrant (km)	-999 = no estimate, missing
42.	*94*	*blank space*	*Single space (ASCII character 32).* *Reused by SYNDAT_QCTROPCY for the data source of the storm circulation depth:* ***C*** *obtained via climatology* ***P*** *obtained via persistence*
43.	95	Storm depth	Estimated depth of storm circulation. **S** shallow, estimated top of circulation is 700 mb **M** medium, estimated top of circulation is 400 mb **D** deep, estimated top of circulation is 200 mb **X** no estimate, missing

5. Acknowledgements

The authors acknowledge Head, NCMRWF for his continuous support.

6. References

E.N. Rajagopal, Munmun Das Gupta, Saji Mohandas, V.S. Prasad, John P. George, G.R. Iyengar and D. Preveen Kumar, 2007: Implementation of T254L64 Global Forecast System at NCMRWF, NCMRWF Technical Report, NMRF/TR/1/2007, https://www.ncmrwf.gov.in/gfs_report_final-t264.pdf

https://www.emc.ncep.noaa.gov/mmb/data_processing/TC-Vitals_description.htm

Trahan, S., and L. Sparling, 2012: An analysis of NCEP Tropical Cyclone Vitals and potential effects on forecasting models. Weather Forecasting, 27, 744–756. https://doi.org/10.1175/WAF-D-11-00063.1.

V.S. Prasad, Saji Mohandas, Munmun Das Gupta, E.N. Rajagopal and Surya Kanti Dutta, 2011: Implementation of Upgraded Global Forecasting Systems (T382L64 and T574L64) at NCMRWF, NCMRWF Technical Report, NCMR/TR/5/2011, https://www.ncmrwf.gov.in/gfs_report_final.pdf

V.S. Prasad, 2020: GTS Data Pre-Processing and Archival System at NCMRWF, NCMRWF Technical Report, NMRF/TR/01/2020, 48p. https://www.ncmrwf.gov.in/Reports-eng/GTS_PP_AS_Report.pdf

World Meteorological Organization (WMO), 2019: World Meteorological Organization Technical Document, Typhoon committee operational manual, Meteorological component, WMO/TD-No. 196, Tropical cyclone programme, Report No. TCP-23, 2019 edition. https://www.wmo.int/pages/prog/www/tcp/documents/TCP-23EDITION2019. pdf

Annexure-I

```
#!/bin/bash
#set -x
###############################################################################
##
##  Name:  grep_hurricane_EGRR_info.sh
##
##  Description:
## This script prepares input from EGRR GTS bulletin to make the cyclone tcvital file
## one arguments are required  date (yyyymmddhh)
###############################################################################
#~~~infile=$1
PDY=$1
if [ $# -eq 0 ]; then
   echo "No arguments provided"
   echo "Provide 'input file' and 'date' (yyyymmddhh)"
   exit 1
fi

CDAY=`echo $PDY| cut -c1-8`
yy=`echo $PDY| cut -c1-4`
mm=`echo $PDY| cut -c5-6`
dd=`echo $PDY| cut -c7-8`
hh=`echo $PDY| cut -c9-10`
grep "$yy $mm $dd $hh" TCA_Bull_SXXT_EGRR.${CDAY} |sort |uniq >egrr.tmp
awk '{print $1}' egrr.tmp |sort -u |sed '/^$/d' >egrr.tclist
for tcname in `cat egrr.tclist`; do
### Grepping orgnization_id, storm_id, present date and hour ####################
orgn_ID="NHC"
echo "${orgn_ID}" | sed -e 's/^[ \t]*//' |uniq
#  storm_id taking from JTWC bulletin ###################################
storm_id=`grep $tcname egrr.tmp |awk '{print $2}' |uniq |sed '/^$/d'`
if [ -z "$storm_id" ]; then
storm_id="NID"
fi
storm_name=`echo $tcname |sed '/^$/d'`

if [ -z "$storm_name" ]; then
storm_name="NONAME"
fi

echo "${storm_id}" | sed -e 's/^[ \t]*//' |uniq
echo "${storm_name}" | sed -e 's/^[ \t]*//' |uniq
echo ${CDAY}
echo ${hh}
###############################################################################
### Grepping the position and removing Characters (N;W;E;.a)
```

```
#storm latitude (negative for south)
#storm longitude (DEG EAST)

latid=`grep -e $tcname egrr.tmp |awk '{print $8}' |cut -d. -f2 |cut -c2-2`
storm_latpos=`grep $tcname egrr.tmp |awk '{print $8}' |sed 's/[NS]//g'`
latid=`echo "$latid" | tr -d $'\r'`
if [[ "$latid" == "S" ]]; then
storm_latpos=-$storm_latpos
else
  storm_latpos=$storm_latpos
fi
echo "${storm_latpos}" | sed -e 's/^[ \t]*//' |uniq

lonid=`grep -e $tcname egrr.tmp |awk '{print $9}' |cut -d. -f2 |cut -c2-2`
storm_lonpos=`grep -e $tcname egrr.tmp |awk '{print $9}' |sed 's/[EW]//g'`
if [[ "$lonid" == "W" ]]; then
storm_lonpos=`echo "scale=1; 360.0 - $storm_lonpos" | tr -d $'\r' |bc`
else
  storm_lonpos=$storm_lonpos
fi
echo "${storm_lonpos}" | sed -e 's/^[ \t]*//' |uniq
#############################################################################
# storm direction (DEG FROM NORTH)
#storm speed (KNOTS)
#Storm central pressure
#storm environmental pressure (MB)
#estimated radius of outermost closed isobar (ROCI), i.e. size of the storm circulation (KM)
###### Grepping storm direction, speed and central pressure ############

####~~~~~~storm_dir=`grep -e $PDY tmp |awk '{print $6}' |uniq`
if [ -z "$storm_dir" ]; then
 storm_dir=-99
else
 storm_dir=$storm_dir
fi
echo "${storm_dir}" | sed -e 's/^[ \t]*//' |uniq
#~~~~~~~~~~~storm_mov=`grep -e $PDY tmp |awk '{print $7}'`
#---------------------------------------------------------------------
#~~~~~~~~~~sed -n "/RJTD/,/MXWD \|MAX WIND/p" TCA_Bull.${PDY} > rjtd_info
sed -n "/RJTD/,/MXWD \|MAX WIND/p" TCA_Bull.${CDAY}_${hh} > rjtd_info
num1=`sed -n "/$tcname/{=;q;}" rjtd_info`
if [ -z "$num1" ]; then
 storm_mov=-99
else
num2=`expr $num1 + 5`
sed -n ''$num1','$num2'p' rjtd_info >rjtd.tmp
#~~grep PRES rjtd.tmp |awk '{print $2}' |sed 's/[HPA].*//'
storm_mov=`grep MOVE rjtd.tmp |awk '{print $3}' |sed 's/[KT].*//' |tr -d $'\r'`
```

```
#~~~  storm_mov=$storm_mov
fi
#if [[ "$storm_mov" == "SLOWLY" ]]; then
if [[ $storm_mov = "SLOWLY" ]]; then
storm_mov=-99
elif [ -z "$storm_mov" ]; then
storm_mov=-99
fi
echo "${storm_mov}" | sed -e 's/^[ \t]*//' |uniq
#_____
_________

storm_CEN_PRES=`grep -e $tcname egrr.tmp |awk '{print $14}' |uniq |tr -d $'\r'`
if [ -z "$storm_CEN_PRES" ]; then
  storm_CEN_PRES=-999
else
  storm_CEN_PRES=$storm_CEN_PRES
fi
echo "${storm_CEN_PRES}" | sed -e 's/^[ \t]*//' |uniq
storm_penv=-999
echo "${storm_penv}" | sed -e 's/^[ \t]*//' |uniq
if [ -z "$storm_rmax" ]; then
  storm_rmax=-999
else
  storm_rmax=$storm_rmax
fi
echo "${storm_rmax}" | sed -e 's/^[ \t]*//' |uniq
###############################################################################################
#  Estimated maximum wind (KNOTS)
# Estimated radius of maximum wind (KM)
#storm_intensity=`grep -r "T000" tmp |awk '{print $4}' | tr -d $'\r' |uniq`
storm_intensity=`grep -e $tcname egrr.tmp |awk '{print $7}' |uniq | tr -d $'\r'`
if [ -z "$storm_intensity" ]; then
  storm_vmax=-9
else
  storm_vmax=$storm_intensity
fi
echo "${storm_vmax}" | sed -e 's/^[ \t]*//' |uniq
##storm_gust=`grep -r "MAX SUSTAINED WINDS" tmpintens |awk '{print $8}'`

if [ -z "$storm_rmw" ]; then
  storm_rmw=-99
else
  storm_rmw=$storm_rmw
fi
echo "${storm_rmw}" | sed -e 's/^[ \t]*//' |uniq
###############################################################################################
##### Grepping the estimated radius of 34 knot  and 50 knot windis in (KM)
```

```
#~~~~~sed -n "/BABJ/,/MOVE/p" TCA_Bull.${PDY} > babj_info
sed -n "/BABJ/,/MOVE/p" TCA_Bull.${CDAY}_${hh} > babj_info
num1=`sed -n "/$tcname/{=;q;}" babj_info`
if [ -z "$num1" ]; then
  r34ne=-999
  r34se=-999
  r34sw=-999
  r34nw=-999

  r50ne=-999
  r50se=-999
  r50sw=-999
  r50nw=-999
else
num2=`expr $num1 + 1`
#sed -n ''$num2','$num2'p' babj_info |awk '{print $4}' |sed 's/[HPA].*//'
#sed -n ''$num2','$num2'p' babj_info |awk '{print $5}' |sed 's/[M/S].*//'
num3=`expr $num1 + 2`
num4=`expr $num3 + 3`
sed -n ''$num3','$num4'p' babj_info >babj.tmp
  r34ne=`grep "NORTHEAST" babj.tmp |awk '{print $3}' |sed 's/[KM].*//'`
  r34se=`grep "SOUTHEAST" babj.tmp |awk '{print $1}' |sed 's/[KM].*//'`
  r34sw=`grep "SOUTHWEST" babj.tmp |awk '{print $1}' |sed 's/[KM].*//'`
  r34nw=`grep "NORTHWEST" babj.tmp |awk '{print $1}' |sed 's/[KM].*//'`

str=`grep "34KT" babj.tmp |awk '{print $1}' |sed 's/[KT].*//'`
if [ -z "$str" ]; then
  r34ne=-999
  r34se=-999
  r34sw=-999
  r34nw=-999
  fi

num5=`expr $num4 + 1`
num6=`expr $num5 + 3`
sed -n ''$num5','$num6'p' babj_info >babj.tmp
  r50ne=`grep "NORTHEAST" babj.tmp |awk '{print $3}' |sed 's/[KM].*//'`
  r50se=`grep "SOUTHEAST" babj.tmp |awk '{print $1}' |sed 's/[KM].*//'`
  r50sw=`grep "SOUTHWEST" babj.tmp |awk '{print $1}' |sed 's/[KM].*//'`
  r50nw=`grep "NORTHWEST" babj.tmp |awk '{print $1}' |sed 's/[KM].*//'`

str1=`grep "50KT" babj.tmp |awk '{print $1}' |sed 's/[KT].*//'`
if [ -z "$str1" ]; then
  r50ne=-999
  r50se=-999
  r50sw=-999
  r50nw=-999
 fi
```

```
fi

echo $r34ne" "$r34se" "$r34sw" "$r34nw
echo $r50ne" "$r50se" "$r50sw" "$r50nw
#########################################################
# estimated top of cyclonic circulation (mb)
#estimated latitude at maximum forecast time
#estimated longitude at maximum forecast time
# maximum forecast time (hours, e.g. 72)
if [ $storm_vmax -gt 0 ] && [ $storm_vmax -le 33 ]; then
storm_ptop=700
elif [ $storm_vmax -ge 34 ] && [ $storm_vmax -le 47 ]; then
  storm_ptop=400
elif [ $storm_vmax -ge 48 ]; then
  storm_ptop=200
fi
#storm_ptop=-999
echo "${storm_ptop}" | sed -e 's/^[ \t]*//' |uniq

storm_fclat=-99.0
echo "${storm_fclat}" | sed -e 's/^[ \t]*//' |uniq

storm_fclon=-99.0
echo "${storm_fclon}" | sed -e 's/^[ \t]*//' |uniq
storm_fcstep=-99
echo "${storm_fcstep}" | sed -e 's/^[ \t]*//' |uniq
######## Making output file ###################################
cat << EOF > input.EGRR_$storm_name.$PDY
${storm_name}
${storm_id}
${orgn_ID}
${CDAY}
${hh}
${storm_latpos}
${storm_lonpos}
${storm_dir}
${storm_mov}
${storm_CEN_PRES}
${storm_penv}
${storm_rmax}
${storm_vmax}
${storm_rmw}
$r34ne $r34se $r34sw $r34nw
$r50ne $r50se $r50sw $r50nw
${storm_ptop}
${storm_fclat}
${storm_fclon}
${storm_fcstep}
```

```
EOF
#rm -f tmp*
echo "done for Cyclone::"${storm_name}
done
exit
```

Annexure-II

```
#set -x
###############################################################################
##  Name:  grep_hurricane_NHC_info.sh
##  Description:
## This script preparing input from GTS bulletin (JTWC & SXXT50_EGRR) to make the cyclone tcvital file
## Two arguments are required input file and date (yyyymmddhh)
###############################################################################


infile=$1
PDY=$2
if [ $# -eq 0 ]; then
   echo "No arguments provided"
   echo "Provide 'input file' and 'date' (yyyymmddhh)"
   exit 1
fi
export SCRIPTDIR=/home/gfsprod/nwprod/global_cyclone_monitoring/script
CDAY=`echo $PDY| cut -c1-8`
yy=`echo $PDY| cut -c1-4`
mm=`echo $PDY| cut -c5-6`
dd=`echo $PDY| cut -c7-8`
hh=`echo $PDY| cut -c9-10`
### Grepping orgnization_id, storm_id, present date and hour ####################
orgn_ID="NHC"
echo "${orgn_ID}" | sed -e 's/^[ \t]*//' |uniq

# storm_name and storm_id taking from (JTWC & SXXT50_EGRR) bulletin
###################################
#storm_id=`grep -r "FORECAST/ADVISORY" $infile |awk '{print $2}' |sed 's/[()]//g'`

string=`sed -n "/FORECAST[/]ADVISORY/{p;q;}" $infile |awk '{print $1}'`
if [[ "$string" == "TROPICAL" ]]; then
#~~~storm_name=`sed -n "/FORECAST[/]ADVISORY/{p;q;}" $infile |awk '{print $3}' |sed 's/[-].*//'`
cyc_name=`sed -n "/FORECAST[/]ADVISORY/{p;q;}" $infile |awk '{print $3}' | tr -d $'\r'`
else
cyc_name=`sed -n "/FORECAST[/]ADVISORY/{p;q;}" $infile |awk '{print $2}' | tr -d $'\r'`
fi
storm_name=`echo $cyc_name |sed 's/[-].*//' | tr -d $'\r'`
if [ -z "$storm_name" ]; then
storm_name="NONAME"
fi

rm -f nhc_stormid
${SCRIPTDIR}/get_NHC_cycID.sh $CDAY $cyc_name
storm_id=`cat nhc_stormid | awk '{print $1}' | tr -d $'\r'`
if [ -z "$storm_id" ]; then
storm_id="NID"
```

```
fi
echo "${storm_id}" | sed -e 's/^[ \t]*//' |uniq


echo "${storm_name}" | sed -e 's/^[ \t]*//'  |uniq
echo ${CDAY}
echo ${hh}
################################################################################
### Grepping the position and removing Characters (N;W;E;.a)
#storm latitude (negative for south)
#storm longitude (DEG EAST)
#~~~~latid=`sed -n "/${dd}[/]${hh}00Z CENTER WAS LOCATED NEAR/{p;q;}" $infile |awk '{print $7}' |cut -d. -
f2 |cut -c2-2`
latid=`sed -n "/${dd}[/]${hh}00Z CENTER WAS LOCATED NEAR/{p;q;}" $infile |awk '{print $7}' | tr -d $'\r'
|awk '{print substr($0,length,1)}'`
storm_latpos=`sed -n "/${dd}[/]${hh}00Z CENTER WAS LOCATED NEAR/{p;q;}" $infile |awk '{print $7}' | tr -d
$'\r' |sed 's/[NS]//g'`
latid=`echo "$latid" | tr -d $'\r'`
if [[ "$latid" == "S" ]]; then
storm_latpos=-$storm_latpos
else
  storm_latpos=$storm_latpos
fi
echo "${storm_latpos}" | sed -e 's/^[ \t]*//' |uniq
lonid=`sed -n "/${dd}[/]${hh}00Z CENTER WAS LOCATED NEAR/{p;q;}" $infile |awk '{print $8}' | tr -d $'\r'
|awk '{print substr($0,length,1)}'`
# To delete everything in a line followed by a character (sed 's/a.*//' file)
storm_lonpos=`sed -n "/${dd}[/]${hh}00Z CENTER WAS LOCATED NEAR/{p;q;}" $infile |awk '{print $8}' | tr -d
$'\r' |sed 's/[EW].*//'`
if [[ "$lonid" == "W" ]]; then
storm_lonpos=`echo "scale=1; 360.0 - $storm_lonpos" | tr -d $'\r' |bc`
else
  storm_lonpos=$storm_lonpos
fi
echo "${storm_lonpos}" | sed -e 's/^[ \t]*//' |uniq
################################################################################
# storm direction (DEG FROM NORTH)
#storm speed (KNOTS)
#Storm central pressure
#storm environmental pressure (MB)
#estimated radius of outermost closed isobar (ROCI), i.e. size of the storm circulation (KM)
###### Grepping storm direction, speed and central pressure ############
awk '/PRESENT MOVEMENT/{x=NR+3}(NR<=x){print}' $infile >tmpmov
#storm_dir=`grep -e "PRESENT MOVEMENT TOWARD" tmpmov |awk '{print $7}' |uniq`
storm_dir=`sed -n "/PRESENT MOVEMENT TOWARD/{p;q;}" $infile |awk '{print $7}' | tr -d $'\r'`
if [ -z "$storm_dir" ]; then
 storm_dir=-99
else
```

```
 storm_dir=$storm_dir
fi
echo "${storm_dir}" | sed -e 's/^[ \t]*//' |uniq
storm_mov=`sed -n "/PRESENT MOVEMENT TOWARD/{p;q;}" $infile |awk '{print $10}' | tr -d $'\r'`
if [ -z "$storm_mov" ]; then
 storm_mov=-99
else
 storm_mov=$storm_mov
fi
echo "${storm_mov}" | sed -e 's/^[ \t]*//' |uniq

#storm_CEN_PRES=`grep -e "ESTIMATED MINIMUM CENTRAL PRESSURE" tmpmov |awk '{print $5}' |uniq`
storm_CEN_PRES=`sed -n "/ESTIMATED MINIMUM CENTRAL PRESSURE/{p;q;}" $infile |awk '{print $5}' | tr -d
$'\r'`
if [ -z "$storm_CEN_PRES" ]; then
 storm_CEN_PRES=-999
else
 storm_CEN_PRES=$storm_CEN_PRES
fi
echo "${storm_CEN_PRES}" | sed -e 's/^[ \t]*//' |uniq

storm_penv=-999
echo "${storm_penv}" | sed -e 's/^[ \t]*//' |uniq

if [ -z "$storm_rmax" ]; then
 storm_rmax=-999
else
 storm_rmax=$storm_rmax
fi
echo "${storm_rmax}" | sed -e 's/^[ \t]*//' |uniq
###############################################################################################
#  Estimated maximum wind (KNOTS)
# Estimated radius of maximum wind (KM)
awk '/MAX SUSTAINED WINDS/{x=NR+7}(NR<=x){print}' $infile >tmp1
#storm_intensity=`grep -r "MAX SUSTAINED WINDS" tmp1 |awk '{print $4}' |tr -d $'\r' |uniq`
storm_intensity=`sed -n "/MAX SUSTAINED WINDS/{p;q;}" tmp1 |awk '{print $4}' |tr -d $'\r'`
if [ -z "$storm_intensity" ]; then
 storm_vmax=-9
else
 storm_vmax=$storm_intensity
fi
echo "${storm_vmax}" | sed -e 's/^[ \t]*//' |uniq

######storm_gust=`grep -r "MAX SUSTAINED WINDS" tmp1 |awk '{print $9}'`

if [ -z "$storm_rmw" ]; then
 storm_rmw=-99
else
```

```
  storm_rmw=$storm_rmw
fi
echo "${storm_rmw}" | sed -e 's/^[ \t]*//' |uniq
################################################################################################
##### Grepping the estimated radius of 34 knot  and 50 knot windis in (KM)
#r34ne=`grep -r "^34 KT" tmp1 |tr -s '[.] [:blank:]' ' ' |awk '{print $3}' |sed 's/[NE]//g' |uniq`
#r34se=`grep -r "^34 KT" tmp1 |tr -s '[.] [:blank:]' ' ' |awk '{print $4}' |sed 's/[SE]//g' |uniq`
#r34sw=`grep -r "^34 KT" tmp1 |tr -s '[.] [:blank:]' ' ' |awk '{print $5}' |sed 's/[SW]//g' |uniq`
#r34nw=`grep -r "^34 KT" tmp1 |tr -s '[.] [:blank:]' ' ' |awk '{print $6}' |sed 's/[NW]//g' |uniq`

r34ne=`sed -n "/^34 KT/{p;q;}" tmp1 |tr -s '[.] [:blank:]' ' ' |awk '{print $3}' |sed 's/[NE]//g'`
r34se=`sed -n "/^34 KT/{p;q;}" tmp1 |tr -s '[.] [:blank:]' ' ' |awk '{print $4}' |sed 's/[SE]//g'`
r34sw=`sed -n "/^34 KT/{p;q;}" tmp1 |tr -s '[.] [:blank:]' ' ' |awk '{print $5}' |sed 's/[SW]//g'`
r34nw=`sed -n "/^34 KT/{p;q;}" tmp1 |tr -s '[.] [:blank:]' ' ' |awk '{print $6}' |sed 's/[NW]//g'`
if [ -z "$r34ne" ]; then
  r34ne=-999
elif [ $r34ne -eq 0 ];then
r34ne=-999
else
#r34ne=`printf %.$2f $(echo "1.852 * $r34ne" | tr -d $'\r' |bc)`
r34ne=`printf %.0f $(echo "1.852 * $r34ne" | tr -d $'\r' | bc)`
fi
if [ -z "$r34se" ]; then
  r34se=-999
elif [ $r34se -eq 0 ];then
r34se=-999
else
r34se=`printf %.0f $(echo "1.852 * $r34se" | tr -d $'\r' |bc)`
fi
if [ -z "$r34sw" ]; then
  r34sw=-999
elif [ $r34sw -eq 0 ];then
r34sw=-999
else
r34sw=`printf %.0f $(echo "1.852 * $r34sw" | tr -d $'\r' |bc)`
fi
if [ -z "$r34nw" ]; then
  r34nw=-999
elif [ $r34nw -eq 0 ];then
r34nw=-999
else
r34nw=`printf %.0f $(echo "1.852 * $r34nw" | tr -d $'\r' |bc)`
fi
r50ne=`sed -n "/^50 KT/{p;q;}" tmp1 |tr -s '[.] [:blank:]' ' ' |awk '{print $3}' |sed 's/[NE]//g'`
r50se=`sed -n "/^50 KT/{p;q;}" tmp1 |tr -s '[.] [:blank:]' ' ' |awk '{print $4}' |sed 's/[SE]//g'`
r50sw=`sed -n "/^50 KT/{p;q;}" tmp1 |tr -s '[.] [:blank:]' ' ' |awk '{print $5}' |sed 's/[SW]//g'`
r50nw=`sed -n "/^50 KT/{p;q;}" tmp1 |tr -s '[.] [:blank:]' ' ' |awk '{print $6}' |sed 's/[NW]//g'`
if [ -z "$r50ne" ]; then
```

```
  r50ne=-999
else
r50ne=`printf %.0f $(echo "1.852 * $r50ne" | tr -d $'\r' |bc)`
fi
if [ -z "$r50se" ]; then
  r50se=-999
else
r50se=`printf %.0f $(echo "1.852 * $r50se" | tr -d $'\r' |bc)`
fi
if [ -z "$r50sw" ]; then
  r50sw=-999
else
r50sw=`printf %.0f $(echo "1.852 * $r50sw" | tr -d $'\r' |bc)`
fi
if [ -z "$r50nw" ]; then
  r50nw=-999
else
r50nw=`printf %.0f $(echo "1.852 * $r50nw" | tr -d $'\r' |bc)`
fi

echo $r34ne" "$r34se" "$r34sw" "$r34nw
echo $r50ne" "$r50se" "$r50sw" "$r50nw
#########################################################
# estimated top of cyclonic circulation (mb)
#estimated latitude at maximum forecast time
#estimated longitude at maximum forecast time
# maximum forecast time (hours, e.g. 72)
if [ $storm_vmax -gt 0 ] && [ $storm_vmax -le 33 ]; then
storm_ptop=700
elif [ $storm_vmax -ge 34 ] && [ $storm_vmax -le 47 ]; then
   storm_ptop=400
elif [ $storm_vmax -ge 48 ]; then
   storm_ptop=200
fi
#storm_ptop=-999
echo "${storm_ptop}" | sed -e 's/^[ \t]*//' |uniq

storm_fclat=-99.0
echo "${storm_fclat}" | sed -e 's/^[ \t]*//' |uniq

storm_fclon=-99.0
echo "${storm_fclon}" | sed -e 's/^[ \t]*//' |uniq

storm_fcstep=-99
echo "${storm_fcstep}" | sed -e 's/^[ \t]*//' |uniq
######## Making output file ##################################
cat << EOF > input.NHC_$storm_name.$PDY
${storm_name}
```

```
${storm_id}
${orgn_ID}
${CDAY}
${hh}
${storm_latpos}
${storm_lonpos}
${storm_dir}
${storm_mov}
${storm_CEN_PRES}
${storm_penv}
${storm_rmax}
${storm_vmax}
${storm_rmw}
$r34ne $r34se $r34sw $r34nw
$r50ne $r50se $r50sw $r50nw
${storm_ptop}
${storm_fclat}
${storm_fclon}
${storm_fcstep}
EOF
rm -f tmp*
exit
```

Annexure-III

```
set -x
#############################################################################
##  Name:  grep_hurricane_JTWC_info.sh
##  Description:
##  This script preparing input from GTS bulletin to make the cyclone tcvital file
##  Two arguments are required input file and date (yyyymmddhh)
##  Authors:
##  Dr. Sanjeev Kumar Singh & Dr. V.S. Prasad, NCMRWF, NOIDA
##  DATE: August 02, 2016
#############################################################################
infile=$1
PDY=$2
if [ $# -eq 0 ]; then
   echo "No arguments provided"
   echo "Provide 'input file' and 'date' (yyyymmddhh)"
   exit 1
fi

CDAY=`echo $PDY| cut -c1-8`
yy=`echo $PDY| cut -c1-4`
mm=`echo $PDY| cut -c5-6`
dd=`echo $PDY| cut -c7-8`
hh=`echo $PDY| cut -c9-10`
### Grepping orgnization_id, storm_id, present date and hour ###################
orgn_ID="JTWC"
echo "${orgn_ID}" | sed -e 's/^[ \t]*//' |uniq
#  storm_id taking from JTWC bulletin ###################################
###awk '/WARNING    ATCG MIL/{x=NR+3}(NR<=x){print}' $infile >tmp
num1=`sed -n "/WARNING    ATCG MIL/{=;q;}" $infile`
num2=`expr $num1 + 2`
sed -n ''$num1','$num2'p' $infile >tmp
storm_id=`grep -r $PDY tmp |awk '{print $2}' | tr -d $'\r' |uniq`
if [ -z "$storm_id" ]; then
storm_id="NID"
fi
storm_name=`grep -r $PDY tmp |awk '{print $3}' |uniq`

if [ -z "$storm_name" ]; then
storm_name="NONAME"
fi

echo "${storm_id}" | sed -e 's/^[ \t]*//' |uniq
echo "${storm_name}" | sed -e 's/^[ \t]*//' |uniq
echo ${CDAY}
echo ${hh}
#############################################################################
### Grepping the position and removing Characters (N;W;E;.a)
```

```
#storm latitude (negative for south)
#storm longitude (DEG EAST)

#~~latid=`sed -n "/T000/{p;q;}" tmp |awk '{print $2}' |cut -c4-4`
latid=`sed -n "/T000/{p;q;}" tmp |awk '{print $2}' | tr -d $'\r' |awk '{print substr($0,length,1)}'`
#storm_latpos=`grep -e "T000" tmp |awk '{print $2}' |sed 's/[NS]//g'`
storm_latpos=`sed -n "/T000/{p;q;}" tmp |awk '{printf $2/10.}' |sed 's/[NS]//g'`
latid=`echo "$latid" | tr -d $'\r'`
if [[ "$latid" == "S" ]]; then
storm_latpos=-$storm_latpos
else
  storm_latpos=$storm_latpos
fi
echo "${storm_latpos}" | sed -e 's/^[ \t]*//' |uniq

lonid=`sed -n "/T000/{p;q;}" tmp |awk '{print $3}' | tr -d $'\r' |awk '{print substr($0,length,1)}'`
storm_lonpos=`sed -n "/T000/{p;q;}" tmp |awk '{printf $3/10.}' |sed 's/[EW]//g'`
if [[ "$lonid" == "W" ]]; then
storm_lonpos=`echo "scale=1; 360.0 - $storm_lonpos" | tr -d $'\r' |bc`
else
  storm_lonpos=$storm_lonpos
fi
echo "${storm_lonpos}" | sed -e 's/^[ \t]*//' |uniq
#########################################################################
# storm direction (DEG FROM NORTH)
#storm speed (KNOTS)
#Storm central pressure
#storm environmental pressure (MB)
#estimated radius of outermost closed isobar (ROCI), i.e. size of the storm circulation (KM)
###### Grepping storm direction, speed and central pressure ############

storm_dir=`grep -e $PDY tmp |awk '{print $6}' | tr -d $'\r' |uniq`
if [ -z "$storm_dir" ]; then
 storm_dir=-99
else
 storm_dir=$storm_dir
fi
echo "${storm_dir}" | sed -e 's/^[ \t]*//' |uniq
storm_mov=`grep -e $PDY tmp |awk '{print $7}' | tr -d $'\r'`
if [ -z "$storm_mov" ]; then
 storm_mov=-99
else
 storm_mov=$storm_mov
fi
echo "${storm_mov}" | sed -e 's/^[ \t]*//' |uniq
#_____________________________________________________________________________________
_________
basin_id=`echo ${storm_id} |cut -c3-3 |uniq`
```

```
if [ "$basin_id" == "A" ] || [ "$basin_id" == "B" ]; then
awk '/TCAC: NEW DELHI/{x=NR+13}(NR<=x){print}' $infile >tmp1
#storm_CEN_PRES=`grep -r "HPA" tmp1 |awk '{print $2}'|sed 's/[HPA]//g' |uniq` ## from IMD bulletin
storm_CEN_PRES=`sed -n "/HPA/{p;q;}" tmp1 |awk '{print $2}'|sed 's/[HPA]//g' |tr -d $'\r'` ## from IMD
bulletin
fi
if [ "$basin_id" == "C" ] || [ "$basin_id" == "W" ]; then
grep -A 10 -B 5 "RJTD" $infile |grep -A 10 -B 5 "$dd$hh00UTC" |grep -A 10 -B 5 $storm_name \
|grep -A 10 -B 5 "RSMC TROPICAL CYCLONE ADVISORY" >tmp2
storm_CEN_PRES=`sed -n "/HPA/{p;q;}" tmp2 |awk '{print $2}'|sed 's/[HPA]//g' |tr -d $'\r'` ## from RJTD
bulletin
fi
#_____
_________
if [ -z "$storm_CEN_PRES" ]; then
  storm_CEN_PRES=-999
else
  storm_CEN_PRES=$storm_CEN_PRES
fi
echo "${storm_CEN_PRES}" | sed -e 's/^[ \t]*//' |uniq
storm_penv=-999
echo "${storm_penv}" | sed -e 's/^[ \t]*//' |uniq
if [ -z "$storm_rmax" ]; then
  storm_rmax=-999
else
  storm_rmax=$storm_rmax
fi
echo "${storm_rmax}" | sed -e 's/^[ \t]*//' |uniq
##############################################################################################
#  Estimated maximum wind (KNOTS)
# Estimated radius of maximum wind (KM)
#storm_intensity=`grep -r "T000" tmp |awk '{print $4}' | tr -d $'\r' |uniq`
storm_intensity=`sed -n "/T000/{p;q;}" tmp |awk '{print $4}' | tr -d $'\r'`
if [ -z "$storm_intensity" ]; then
  storm_vmax=-9
else
  storm_vmax=$storm_intensity
fi
echo "${storm_vmax}" | sed -e 's/^[ \t]*//' |uniq
##storm_gust=`grep -r "MAX SUSTAINED WINDS" tmpintens |awk '{print $8}'`

if [ "$basin_id" == "A" ] || [ "$basin_id" == "B" ]; then
storm_rmwkt=`sed -n "/RADIUS OF MAXIMUM WIND/{p;q;}" $infile |awk '{print $5}' | tr -d $'\r' |uniq`
storm_rmw=`printf %.0f $(echo "1.852 * $storm_rmwkt" | tr -d $'\r' |bc -l )`
fi
if [ -z "$storm_rmw" ] || [ "$storm_rmw" == 0 ]; then
  storm_rmw=-99
else
```

```
 storm_rmw=$storm_rmw
fi
echo "${storm_rmw}" | sed -e 's/^[ \t]*//' |uniq
#################################################################################################
r34ne=`sed -n -e 's/^.*R034//p' tmp | awk '{print $1}' | tr -d $'\r'`
r34se=`sed -n -e 's/^.*R034//p' tmp | awk '{print $4}' | tr -d $'\r'`
r34sw=`sed -n -e 's/^.*R034//p' tmp | awk '{print $7}' | tr -d $'\r'`
r34nw=`sed -n -e 's/^.*R034//p' tmp | awk '{print $10}' | tr -d $'\r'`
if [ -z "$r34ne" ]; then
 r34ne=-999
else
r34ne=`printf %.0f $(echo "1.852 * $r34ne" | tr -d $'\r' | bc)`
fi
if [ -z "$r34se" ]; then
 r34se=-999
else
r34se=`printf %.0f $(echo "1.852 * $r34se" | tr -d $'\r' |bc)`
fi
if [ -z "$r34sw" ]; then
 r34sw=-999
else
r34sw=`printf %.0f $(echo "1.852 * $r34sw" | tr -d $'\r' |bc)`
fi
if [ -z "$r34nw" ]; then
 r34nw=-999
else
r34nw=`printf %.0f $(echo "1.852 * $r34nw" | tr -d $'\r' |bc)`
fi
r50ne=`sed -n -e 's/^.*R050//p' tmp | awk '{print $1}' | tr -d $'\r'`
r50se=`sed -n -e 's/^.*R050//p' tmp | awk '{print $4}' | tr -d $'\r'`
r50sw=`sed -n -e 's/^.*R050//p' tmp | awk '{print $7}' | tr -d $'\r'`
r50nw=`sed -n -e 's/^.*R050//p' tmp | awk '{print $10}' | tr -d $'\r'`

if [ -z "$r50ne" ]; then
 r50ne=-999
else
r50ne=`printf %.0f $(echo "1.852 * $r50ne" | tr -d $'\r' |bc)`
fi
if [ -z "$r50se" ]; then
 r50se=-999
else
r50se=`printf %.0f $(echo "1.852 * $r50se" | tr -d $'\r' |bc)`
fi
if [ -z "$r50sw" ]; then
 r50sw=-999
else
r50sw=`printf %.0f $(echo "1.852 * $r50sw" | tr -d $'\r' |bc)`
fi
```

```
if [ -z "$r50nw" ]; then
  r50nw=-999
else
r50nw=`printf %.0f $(echo "1.852 * $r50nw" | tr -d $'\r' |bc)`
fi
echo $r34ne" "$r34se" "$r34sw" "$r34nw
echo $r50ne" "$r50se" "$r50sw" "$r50nw
# estimated top of cyclonic circulation (mb)
#estimated latitude at maximum forecast time
#estimated longitude at maximum forecast time
# maximum forecast time (hours, e.g. 72)
if [ $storm_vmax -gt 0 ] && [ $storm_vmax -le 33 ]; then
storm_ptop=700
elif [ $storm_vmax -ge 34 ] && [ $storm_vmax -le 47 ]; then
   storm_ptop=400
elif [ $storm_vmax -ge 48 ]; then
   storm_ptop=200
fi
#storm_ptop=-999
echo "${storm_ptop}" | sed -e 's/^[ \t]*//' |uniq

storm_fclat=-99.0
echo "${storm_fclat}" | sed -e 's/^[ \t]*//' |uniq

storm_fclon=-99.0
echo "${storm_fclon}" | sed -e 's/^[ \t]*//' |uniq

storm_fcstep=-99
echo "${storm_fcstep}" | sed -e 's/^[ \t]*//' |uniq
######## Making output file ##################################
cat << EOF > input.JTWC_$storm_name.$PDY
${storm_name}
${storm_id}
${orgn_ID}
${CDAY}
${hh}
${storm_latpos}
${storm_lonpos}
${storm_dir}
${storm_mov}
${storm_CEN_PRES}
${storm_penv}
${storm_rmax}
${storm_vmax}
${storm_rmw}
$r34ne $r34se $r34sw $r34nw
$r50ne $r50se $r50sw $r50nw
${storm_ptop}
```

```
${storm_fclat}
${storm_fclon}
${storm_fcstep}
EOF
rm -f tmp*
exit
```

Annexure-IV

```
set -x
###############################################################################
##  Name: grep_hurricane_JTWC-Second_info.sh
##  Description:
## This script preparing input from GTS bulletin to make the cyclone tcvital file
## Two arguments are required input file and date (yyyymmddhh)
##  Authors:
##  Dr. Sanjeev Kumar Singh & Dr. V.S. Prasad, NCMRWF, NOIDA
##  DATE: August 02, 2016
###############################################################################
export infile=$1
export PDY=$2
if [ $# -eq 0 ]; then
   echo "No arguments provided"
   echo "Provide 'input file' and 'date' (yyyymmddhh)"
   exit 1
fi

CDAY=`echo $PDY| cut -c1-8`
yy=`echo $PDY| cut -c1-4`
mm=`echo $PDY| cut -c5-6`
dd=`echo $PDY| cut -c7-8`
hh=`echo $PDY| cut -c9-10`
### Grepping orgnization_id, storm_id, present date and hour ####################
orgn_ID="JTWC"
echo "${orgn_ID}" | sed -e 's/^[ \t]*//' |uniq
#awk '/SUBJ[/]HURRICANE/{x=NR+10}(NR<=x){print}' $infile >trimedfile
#export infile='trimedfile'
#  storm_id taking from JTWC bulletin ###################################
num1=`sed -n "/SUBJ[/]TROPICAL CYCLONE/{=;q;}" $infile`
if [ -z "$num1" ]; then
num1=`sed -n "/SUBJ[/]TYPHOON/{=;q;}" $infile`
fi
if [ -z "$num1" ]; then
num1=`sed -n "/1. TYPHOON/{=;q;}" $infile`
fi
num2=`expr $num1 + 2`
sed -n ''$num1','$num2'p' $infile >tmp
storm_id=`sed -n "/1. TYPHOON/{p;q;}" tmp |awk '{print $3}' |sed 's/[()[]//g' | tr -d $'\r'`
if [ -z "$storm_id" ]; then
storm_id=`sed -n "/1. SUPER TYPHOON/{p;q;}" tmp |awk '{print $4}' |sed 's/[()[]//g' | tr -d $'\r'`
fi
if [ -z "$storm_id" ]; then
storm_id=`sed -n "/1. TROPICAL DEPRESSION/{p;q;}" tmp |awk '{print $4}' |sed 's/[()[]//g' | tr -d $'\r'`
fi
if [ -z "$storm_id" ]; then
storm_id=`sed -n "/1. TROPICAL CYCLONE/{p;q;}" tmp |awk '{print $4}' |sed 's/[()[]//g' | tr -d $'\r'`
```

```
fi
if [ -z "$storm_id" ]; then
storm_id=`sed -n "/1. TROPICAL STORM/{p;q;}" tmp |awk '{print $4}' |sed 's/[)(]//g' | tr -d $'\r'`
fi
echo "${storm_id}" | sed -e 's/^[ \t]*//' |uniq

storm_name=`sed -n "/1. TYPHOON/{p;q;}" tmp |awk '{print $4}' |sed 's/[)(]//g' | tr -d $'\r'`
if [ -z "$storm_name" ]; then
storm_name=`sed -n "/1. SUPER TYPHOON/{p;q;}" tmp |awk '{print $5}' |sed 's/[)(]//g' | tr -d $'\r'`
fi
if [ -z "$storm_name" ]; then
storm_name=`sed -n "/1. TROPICAL DEPRESSION/{p;q;}" tmp |awk '{print $5}' |sed 's/[)(]//g' | tr -d $'\r'`
fi
if [ -z "$storm_id" ]; then
storm_id=`sed -n "/1. TROPICAL CYCLONE/{p;q;}" tmp |awk '{print $4}' |sed 's/[)(]//g' | tr -d $'\r'`
fi
if [ -z "$storm_id" ]; then
storm_id=`sed -n "/1. TROPICAL STORM/{p;q;}" tmp |awk '{print $4}' |sed 's/[)(]//g' | tr -d $'\r'`
fi
echo "${storm_id}" | sed -e 's/^[ \t]*//' |uniq

storm_name=`sed -n "/1. TYPHOON/{p;q;}" tmp |awk '{print $4}' |sed 's/[)(]//g' | tr -d $'\r'`
if [ -z "$storm_name" ]; then
storm_name=`sed -n "/1. SUPER TYPHOON/{p;q;}" tmp |awk '{print $5}' |sed 's/[)(]//g' | tr -d $'\r'`
fi
if [ -z "$storm_name" ]; then
storm_name=`sed -n "/1. TROPICAL DEPRESSION/{p;q;}" tmp |awk '{print $5}' |sed 's/[)(]//g' | tr -d $'\r'`
fi
if [ -z "$storm_name" ]; then
storm_name=`sed -n "/1. TROPICAL CYCLONE/{p;q;}" tmp |awk '{print $5}' |sed 's/[)(]//g' | tr -d $'\r'`
fi
if [ -z "$storm_name" ]; then
storm_name=`sed -n "/1. TROPICAL STORM/{p;q;}" tmp |awk '{print $5}' |sed 's/[)(]//g' | tr -d $'\r'`
fi
echo "${storm_name}" | sed -e 's/^[ \t]*//' |uniq
echo ${CDAY}
echo ${hh}

################################################################################
### Grepping the position and removing Characters (N;W;E;.a)
#storm latitude (negative for south)
#storm longitude (DEG EAST)
num3=`sed -n "/WARNING POSITION:/{=;q;}" $infile`
num4=`expr $num3 + 2`
sed -n ''$num3','$num4'p' $infile >tmp1
#awk '/WARNING POSITION:/{x=NR+3}(NR<=x){print}' $infile >tmp
#~~~~~latid=`sed -n "/${dd}${hh}00Z --- NEAR/{p;q;}" tmp1 |awk '{print $4}' |cut -d. -f2 |cut -c2-2`
```

```
latid=`sed -n "/${dd}${hh}00Z --- NEAR/{p;q;}" tmp1 |awk '{print $4}' | tr -d $'\r' |awk '{print
substr($0,length,1)}'`
storm_latpos=`sed -n "/${dd}${hh}00Z --- NEAR/{p;q;}" tmp1 |awk '{print $4}' |sed 's/[NS]//g'`
latid=`echo "$latid" | tr -d $'\r'`
if [[ "$latid" == "S" ]]; then
storm_latpos=-$storm_latpos
else
  storm_latpos=$storm_latpos
fi
echo "${storm_latpos}" | sed -e 's/^[ \t]*//' |uniq

#~~~~lonid=`sed -n "/${dd}${hh}00Z --- NEAR/{p;q;}" tmp1 |awk '{print $5}' |cut -d. -f2 |cut -c2-2`
lonid=`sed -n "/${dd}${hh}00Z --- NEAR/{p;q;}" tmp1 |awk '{print $5}' | tr -d $'\r' |awk '{print
substr($0,length,1)}'`
storm_lonpos=`sed -n "/${dd}${hh}00Z --- NEAR/{p;q;}" tmp1 |awk '{print $5}' |sed 's/[EW]//g' | tr -d $'\r'`
if [[ "$lonid" == "W" ]]; then
storm_lonpos=`echo "scale=1; 360.0 - $storm_lonpos" | tr -d $'\r' |bc`
else
  storm_lonpos=$storm_lonpos
fi
echo "${storm_lonpos}" | sed -e 's/^[ \t]*//' |uniq
#########################################################################
# storm direction (DEG FROM NORTH)
#storm speed (KNOTS)
#Storm central pressure
#storm environmental pressure (MB)
#estimated radius of outermost closed isobar (ROCI), i.e. size of the storm circulation (KM)
###### Grepping storm direction, speed and central pressure ############
storm_dir=`sed -n "/MOVEMENT PAST SIX HOURS/{p;q;}" tmp1 |awk '{print $6}' | tr -d $'\r'`
if [ -z "$storm_dir" ]; then
 storm_dir=-99
else
 storm_dir=$storm_dir
fi
echo "${storm_dir}" | sed -e 's/^[ \t]*//' |uniq

storm_mov=`sed -n "/MOVEMENT PAST SIX HOURS/{p;q;}" tmp1 |awk '{print $9}' | tr -d $'\r'`
if [ -z "$storm_mov" ]; then
 storm_mov=-99
else
 storm_mov=$storm_mov
fi
echo "${storm_mov}" | sed -e 's/^[ \t]*//' |uniq
#_____________________________________________________________________________________________
_________
#if [[ -s ${KHNC}  ]]; then
#awk '/PRESENT MOVEMENT/{x=NR+3}(NR<=x){print}' $KHNC >tmpmov
```

```
#storm_CEN_PRES=`grep -e "ESTIMATED MINIMUM CENTRAL PRESSURE" tmpmov |awk '{print $5}'` ## from
KHNC bulletin
#fi
basin_id=`echo ${storm_id} |cut -c3-3 |uniq`

if [ "$basin_id" == "A" ] || [ "$basin_id" == "B" ]; then
awk '/TCAC: NEW DELHI/{x=NR+13}(NR<=x){print}' $infile >tmp1
#storm_CEN_PRES=`grep -r "HPA" tmp1 |awk '{print $2}'|sed 's/[HPA]//g' |uniq` ## from IMD bulletin
storm_CEN_PRES=`sed -n "/HPA/{p;q;}" tmp1 |awk '{print $2}'|sed 's/[HPA]//g' | tr -d $'\r'` ## from IMD
bulletin
fi

if [ "$basin_id" == "C" ] || [ "$basin_id" == "W" ]; then
grep -A 10 -B 5 "RJTD" ${infile} |grep -A 10 -B 5 "${dd}${hh}00UTC" |grep -A 10 -B 5 "${storm_name}" \
|grep -A 10 -B 5 "RSMC TROPICAL CYCLONE ADVISORY" >tmp2
storm_CEN_PRES=`sed -n "/HPA/{p;q;}" tmp2 |awk '{print $2}'|sed 's/[HPA]//g' | tr -d $'\r'` ## from RJTD
bulletin
fi
#_____
_________
if [ -z "$storm_CEN_PRES" ]; then
  storm_CEN_PRES=-999
else
  storm_CEN_PRES=$storm_CEN_PRES
fi
echo "${storm_CEN_PRES}" | sed -e 's/^[ \t]*//' |uniq

storm_penv=-999
echo "${storm_penv}" | sed -e 's/^[ \t]*//' |uniq

if [ -z "$storm_rmax" ]; then
  storm_rmax=-999
else
  storm_rmax=$storm_rmax
fi
echo "${storm_rmax}" | sed -e 's/^[ \t]*//' |uniq
################################################################################################
#  Estimated maximum wind (KNOTS)
# Estimated radius of maximum wind (KM)
num5=`sed -n "/PRESENT WIND DISTRIBUTION:/{=;q;}" $infile`
num6=`expr $num5 + 2`
sed -n ''$num5','$num6'p' $infile >tmp3
storm_intensity=`sed -n "/MAX SUSTAINED WINDS -/{p;q;}" tmp3 |awk '{print $5}' | tr -d $'\r'`
if [ -z "$storm_intensity" ]; then
  storm_vmax=-9
else
  storm_vmax=$storm_intensity
fi
```

```
echo "${storm_vmax}" | sed -e 's/^[ \t]*//' |uniq
##storm_gust=`grep -r "MAX SUSTAINED WINDS" tmpintens |awk '{print $8}'`

if [ "$basin_id" == "A" ] || [ "$basin_id" == "B" ]; then
storm_rmwkt=`sed -n "/RADIUS OF MAXIMUM WIND/{p;q;}" $infile |awk '{print $5}' | tr -d $'\r' |uniq`
storm_rmw=`printf %.0f $(echo "1.852 * $storm_rmwkt" | tr -d $'\r' | bc)`
fi

if [ -z "$storm_rmw" ]; then
  storm_rmw=-99
else
  storm_rmw=$storm_rmw
fi
echo "${storm_rmw}" | sed -e 's/^[ \t]*//' |uniq
################################################################################################
##### Grepping the estimated radius of 34 knot  and 50 knot windis in (KM)
num7=`sed -n "/RADIUS OF 034 KT WINDS -/{=;q;}" $infile`
num8=`expr $num7 + 3`
sed -n ''$num7','$num8'p' $infile >tmp4
r34ne=`sed -n "/NORTHEAST QUADRANT/{p;q;}" tmp4 |awk '{print $7}' | tr -d $'\r'` ## from PGTW bulletin
r34se=`sed -n "/SOUTHEAST QUADRANT/{p;q;}" tmp4 |awk '{print $1}' | tr -d $'\r'` ## from PGTW bulletin
r34sw=`sed -n "/SOUTHWEST QUADRANT/{p;q;}" tmp4 |awk '{print $1}' | tr -d $'\r'` ## from PGTW bulletin
r34nw=`sed -n "/NORTHWEST QUADRANT/{p;q;}" tmp4 |awk '{print $1}' | tr -d $'\r'` ## from PGTW bulletin
if [ -z "$r34ne" ]; then
  r34ne=-999
else
r34ne=`printf %.0f $(echo "1.852 * $r34ne" | tr -d $'\r' | bc)`
fi
if [ -z "$r34se" ]; then
  r34se=-999
else
r34se=`printf %.0f $(echo "1.852 * $r34se" | tr -d $'\r' |bc)`
fi
if [ -z "$r34sw" ]; then
  r34sw=-999
else
r34sw=`printf %.0f $(echo "1.852 * $r34sw" | tr -d $'\r' |bc)`
fi
if [ -z "$r34nw" ]; then
  r34nw=-999
else
r34nw=`printf %.0f $(echo "1.852 * $r34nw" | tr -d $'\r' |bc)`
fi

num9=`sed -n "/RADIUS OF 050 KT WINDS -/{=;q;}" $infile`
num10=`expr $num10 + 3`
sed -n ''$num9','$num8'p' $infile >tmp5
r50ne=`sed -n "/NORTHEAST QUADRANT/{p;q;}" tmp5 |awk '{print $7}' | tr -d $'\r'` ## from PGTW bulletin
```

```
r50se=`sed -n "/SOUTHEAST QUADRANT/{p;q;}" tmp5 |awk '{print $1}' | tr -d $'\r'` ## from PGTW bulletin
r50sw=`sed -n "/SOUTHWEST QUADRANT/{p;q;}" tmp5 |awk '{print $1}' | tr -d $'\r'` ## from PGTW bulletin
r50nw=`sed -n "/NORTHWEST QUADRANT/{p;q;}" tmp5 |awk '{print $1}' | tr -d $'\r'` ## from PGTW bulletin
if [ -z "$r50ne" ]; then
  r50ne=-999
else
r50ne=`printf %.0f $(echo "1.852 * $r50ne" | tr -d $'\r' |bc)`
fi
if [ -z "$r50se" ]; then
  r50se=-999
else
r50se=`printf %.0f $(echo "1.852 * $r50se" | tr -d $'\r' |bc)`
fi
if [ -z "$r50sw" ]; then
  r50sw=-999
else
r50sw=`printf %.0f $(echo "1.852 * $r50sw" | tr -d $'\r' |bc)`
fi
if [ -z "$r50nw" ]; then
  r50nw=-999
else
r50nw=`printf %.0f $(echo "1.852 * $r50nw" | tr -d $'\r' |bc)`
fi
echo $r34ne" "$r34se" "$r34sw" "$r34nw
echo $r50ne" "$r50se" "$r50sw" "$r50nw
#########################################################
# estimated top of cyclonic circulation (mb)
#estimated latitude at maximum forecast time
#estimated longitude at maximum forecast time
# maximum forecast time (hours, e.g. 72)
if [ $storm_vmax -gt 0 ] && [ $storm_vmax -le 33 ]; then
storm_ptop=700
elif [ $storm_vmax -ge 34 ] && [ $storm_vmax -le 47 ]; then
   storm_ptop=400
elif [ $storm_vmax -ge 48 ]; then
   storm_ptop=200
fi

#storm_ptop=-999
echo "${storm_ptop}" | sed -e 's/^[ \t]*//' |uniq

storm_fclat=-99
echo "${storm_fclat}" | sed -e 's/^[ \t]*//' |uniq

storm_fclon=-99
echo "${storm_fclon}" | sed -e 's/^[ \t]*//' |uniq

storm_fcstep=-99
```

```
echo "${storm_fcstep}" | sed -e 's/^[ \t]*//' |uniq
######## Making output file #################################
cat << EOF > input.JTWC-Second_$storm_name.$PDY
${storm_name}
${storm_id}
${orgn_ID}
${CDAY}
${hh}
${storm_latpos}
${storm_lonpos}
${storm_dir}
${storm_mov}
${storm_CEN_PRES}
${storm_penv}
${storm_rmax}
${storm_vmax}
${storm_rmw}
$r34ne $r34se $r34sw $r34nw
$r50ne $r50se $r50sw $r50nw
${storm_ptop}
${storm_fclat}
${storm_fclon}
${storm_fcstep}
EOF
exit
############################################################
```