

H
QC
874.3
U63
no.36

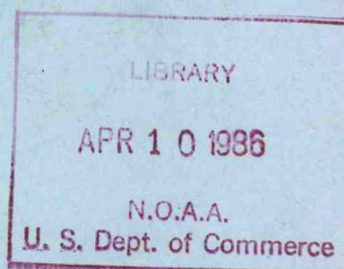
NOAA Eastern Region Computer Programs
and Problems NWS ERCP - No. 36



LAWEB: DATA PROCESSING FOR THE GREAT LAKES

William C. Randel and
Matthew R. Peroutka
National Weather Service Forecast Office
Cleveland, OH

Scientific Services Division
Eastern Region Headquarters
March 1986



**U.S. DEPARTMENT OF
COMMERCE**

/ **National Oceanic and
Atmospheric Administration**

/ **National Weather
Service**

NOAA TECHNICAL MEMORANDUM

National Weather Service, Eastern Region Computer Programs and Problems

The Eastern Region Computer Programs and Problems (ERCP) series is a subset of the Eastern Region Technical Memorandum series. It will serve as the vehicle for the transfer of information about fully documented AFOS application programs. The format ERCP - No. 1 will serve as the model for future issuances in this series.

- 1 An AFOS version of the Flash Flood Checklist. Cynthia M. Scott, March 1981. (PB81 211252).
- 2 An AFOS Applications Program to Compute Three-Hourly Stream Stages. Alan P. Blackburn, September 1981. (PB82 156886).
- 3 PUPPY (AFOS Hydrologic Data Reporting Program). Daniel P. Provost, December 1981. (PB82 199720).
- 4 Special Search Computer Program. Alan P. Blackburn, April 1982. (PB83 175455).
- 5 Conversion of ALEMBICS Workbins. Alan P. Blackburn, October 1982. (PB83 138313).
- 6 Real-Time Quality Control of SAOs. John A. Billet, January 1983. (PB83 166082).
- 7 Automated Hourly Weather Collective from HRR Data Input. Lawrence Cedrone, January 1983. (PB83 167122).
- 8 Decoders for FRH, FTJ and FD Products. Cynthia M. Scott, February 1983. (PB83 176057).
- 9 Stability Analysis Program. Hugh M. Stone, March 1983. (PB83 197947).
- 10 Help for AFOS Message Comp. Alan P. Blackburn, May 1983. (PB83 213561).
- 11 Stability and Other Parameters from the First Transmission RAOB Data. Charles D. Little, May 1983. (PB83 220475).
- 12 TERR, PERR, and BIGC: Three Programs to Compute Verification Statistics. Matthew R. Peroutka, August 1983. (PB84 127521).
- 13 Decoder for Manually Digitized Radar Observations. Matthew R. Peroutka, June 1983. (PB84 127539).
- 14 Slick and Quick Data Entry for AFOS Era Verification (AEV) Program. Alan P. Blackburn, December 1983. (PB84 138726).
- 15 MDR--Processing Manually Digitized Radar Observations. Matthew R. Peroutka, November 1983. (PB84 161462) (Revised June 1985, PB85-220580/AS)
- 16 RANP: Stability Analysis Program. Hugh M. Stone, February 1984. (PB84 161447)
- 17 ZONES. Gerald G. Rigdon, March 1984. (PB84 174325)
- 18 Automated Analysis of Upper Air Soundings to Specify Precipitation Type. Joseph R. Bocchieri and Gerald G. Rigdon, March 1984. (PB84 174333)

(Continued on Inside Rear Cover)

NOAA Eastern Region Computer Programs and Problems NWS ERCP - No. 36

H
QC
874.3
U63
no. 36

LAWEB: DATA PROCESSING FOR THE GREAT LAKES

William C. /Randel and
Matthew R. Peroutka
National Weather Service Forecast Office
Cleveland, OH

Scientific Services Division
Eastern Region Headquarters
March 1986

LAWEB: DATA PROCESSING FOR THE GREAT LAKES
William C. Randel and Matthew R. Peroutka
National Weather Service Forecast Office
Cleveland, OH

I. General Information

Meteorologic data from the Great Lakes region is transmitted in an unbelievable number of different formats. Indeed, almost every form of observation imaginable can be found somewhere around the Lakes. Data buoys report in standard, international ship code, but most Great Lakes vessels use a modified version of that code. Some coastal stations report wind directions using tens of degrees from north while others use sixteen points of the compass. Wave heights are reported in feet or half-meters depending on the source, and wind speeds can be either knots or meters per second. Of course, some temperatures are reported in Fahrenheit while others are Celsius.

This network of observations supplements the usual surface airways (SA) and surface synoptic (SM) reports.

This paper describes a program which gathers all of the available observations on the Great Lakes and presents them in two, unified formats. The first format is a broadcast text which is read over the Great Lakes VHF radio system every six hours during the shipping season. This text contains wind and wave reports as well as weather and visibility when the visibility is less than four miles. Figure 1 is an example of this broadcast text. The second format is that of a standard AFOS plot file, complete with sea level pressure, temperature, dew point, water temperature and wave height, when available. Figure 2 is a sample plot file. National Weather Service (NWS) offices can produce customized graphic displays and analyses from this file using the PMOD system of programs and various mesoscale analysis routines. Figure 3 shows a sample graphic product for Lake Erie, and Figure 4 is a key to interpret the station models plotted.

A great effort has been made to produce a program which will allow major revisions of the output products without altering the FORTRAN code. Experience has shown that such changes occur quite frequently, and that flexibility is a very valuable virtue.

II. Methodology and Software Structure

The LAWEB (LAke WEather Broadcast) program is invoked with one argument. This is the name of the Command File. The Command File, as its name implies, completely directs the operation of the program as it shapes its output products.

The LAWEB program simply reads lines from the Command File and executes them until it reaches the end of the file. Most lines in the Command File begin with two digits. These two digits are command numbers, and they specify which action the program is to perform next. If a line does not begin with two digits, it is assumed to be extra text for the

broadcast product, and it is copied verbatim if a broadcast file is open.

Figure 5 contains a sample Command File, and Figure 6 describes the functions performed and the fields defined for each command number.

The first four codes control the selection of output products. This allows the program to generate multiple broadcast and/or plot products. It is possible to add data to one of the products and omit it from the other using these four commands.

Specialized decoders were developed for most data types for the LAWEB program. This provided assurance that changes in observation formats could be handled quickly and efficiently at the local level. The TDL decoder for surface airways observations, AIRDX, was used with very few modifications. The modified driver was named ADXX.

Overall, the code is not as efficient as it could be, and the program file is quite large. The structure is very modular, however, and the code should prove to be very easy to maintain through the years.

When the program begins execution, the subroutine GTIM computes the valid time of observations which will be accepted. The valid time is generally the closest three-hourly time to the current time. Examples are the best way to illustrate the algorithm used. From 1100 GMT to 1359 GMT the valid time is 1200 GMT. From 1400 GMT to 1659 GMT the valid time is 1500 GMT. The valid time is expressed in two forms. The first form uses integers which represent the traditional month, day, year, and hour. The second form is a floating point number which represents the number of minutes which have elapsed since the year began. This second number is called the Julian minute, and all AFOS products contain this information in the header of their first data block.

The LAWEB program computes a "cut time" which is the Julian minute one half-hour before the valid time. This is the earliest most products can be generated if they are to be considered current by the program. This "cut time" is used extensively by the decoders to find an appropriate version of a given observation. In some cases, Coast Guard observations particularly, the "cut time" is the only reliable way to identify the time of a report. Most of the decoders, however, only use the "cut time" as a coarse measure of the timeliness of an observation; internal coding usually provides a better measure of when an observation was taken.

All decoders return temperatures in Fahrenheit, wave heights in feet, visibilities in sixteenths of a mile, and pressures in tenths of millibars. United States Coast Guard reports contain pressures in hundredths of inches of mercury. These are converted to tenths of millibars by multiplying by a factor of 3.38638. Altimeter settings and sea level pressures are mixed freely in the plot file and its resulting graphics, and this occasionally foils attempts to analyze the pressure field.

III. Cautions and Restrictions

Since the decoders key on AFOS's internal record of when a product was stored in the database, it may be impossible to edit a garbled product and have the program accept it. In such cases, the time the edited version reached the database may be too late for the program to accept the observation. The system can be "fooled" into accepting the observation by changing the system clock while editing, but this practice is to be discouraged (if not eschewed).

A great deal of care was taken to assure that garbled reports would not confuse the decoders. It is possible to produce runtime errors, however, if the data is garbled badly enough. Most reasonable mistakes are handled by the routines without incident.

The command numbers which are entered at the beginning of most lines in the command file must be valid codes from Figure 6. The main program uses a computed GOTO statement to process these codes, and an invalid command number will produce a fatal runtime error.

The map backgrounds developed for the AFOS system do not do justice to the complex coastlines of the Great Lakes. Many coastal stations appear to be plotted well offshore. In most of these instances, the position of the station is correct, and the coastline representation is at fault.

Most reports which plot on the graphic contain either a dew point or a water temperature. Almost none of them report both. It may be hard to tell which is being displayed for a given station, but a little experience helps in this matter. The water temperature is plotted a bit lower than the dew point.

The LAWEB program obtains pixel coordinates from the file STDIR.MS. If a station is not in that file, it will not appear in the plot file. The current version of STDIR.MS available is missing many of the Great Lakes Coastal stations.

A bug in the current GDM firmware prevents the plotting of a station model with two octas of sky cover. The LAWEB program, therefore, silently converts all such reports to a value of three octas.

In testing the graphics produced by the plot products, two bugs were discovered in the current version of the PMOD software. A visibility of zero is not plotted by PMOD. PMOD also plots visibilities of one-sixteenth of a mile as "1/61". Both of these bugs are usually easy to spot on a LAWEB graphic in a fog situation. The next version of PMOD (version 1.12) will correct these.

IV. References

Acknowledgement: Although this program is a total re-write, we'd like to acknowledge the outstanding cooperation and assistance of Charles A. Smith, HIC, RFC Hartford, in providing WSFO Cleveland with the original version of a LAWEB program.

National Weather Service AFOS Operations Division, 1983: Computer Program NWS AOD CP 83-1, PMOD Plotting System for AFOS, National Oceanic and Atmospheric Administration, U.S. Department of Commerce pages 3-1 to 3-34.

National Weather Service, 1982: Great Lakes Observing Handbook Number 1, Marine Surface Weather Observations, National Oceanic and Atmospheric Administration, U.S. Department of Commerce, 117 pages

National Weather Service Eastern Region Data Acquisition Division, 1979: Marine Weather Observing and Reporting, National Oceanic and Atmospheric Administration, U.S. Department of Commerce, 11 pages.

National Weather Service, 1982: Observing Handbook Number 1, Marine Surface Weather Observations, National Oceanic and Atmospheric Administration, U.S. Department of Commerce, 176 pages

Perrotti, Herman P., 1984: Computer Program NWS TDL CP 84-2, AFOS Surface Observation Decoding, Techniques Development Laboratory, National Weather Service, National Oceanic and Atmospheric Administration, U.S. Department of Commerce, 35 pages.

Scott, Cynthia, 1984: "Accessing the Station Directory File (STDIR)", AFOS Application Quarterly Report, May, 1984, National Weather Service, National Oceanic and Atmospheric Administration, pages 25-29.

U.S. Coast Guard, 1983: U.S. Coast Guard Commandant Instruction 3140.3C, U.S. Coast Guard, U.S. Department of Transportation, 17 pages.

LAWEB: DATA PROCESSING FOR THE GREAT LAKES

PART A: INFORMATION AND INSTALLATIONPROGRAM NAME: LAWEBAAL ID:REVISION NO.: 2.00

PURPOSE: Reads marine data from a large number of database products in a wide variety of formats. Produces two types of database products. First, a broadcast text which is read over the Great Lakes VHF Radio System, and, second, a plot file which is compatible with the PMOD system.

PROGRAM INFORMATION:

Development Programmer(s):

William C. Randel

Matthew R. Peroutka

Location: WSFO Cleveland, OH

Phone: (FTS) 942-4949

Language: FORTRAN/Rev. 5.57

Save File Creation Date: February 28, 1986

Running Time: about four minutes for all available data

Disk Space:

Program Files 113 RDOS blocks

Data Files 20 RDOS blocks

Maintenance Programmer(s):

William C. Randel

Matthew R. Peroutka

Location: WSFO Cleveland, OH

Phone: (FTS) 942-4949

Type: Standard

PROGRAM REQUIREMENTSProgram Files:Name

LAWEB.SV

Comments

Main program

Data Files:Name

(various)

DP Location

DP0

Read/Write

Read

Comments

Command File

Name specified in command line

cccnnnxxx

DP0

Write

Temporary output for database products
Names specified in command file

AFOS Products:

<u>ID</u>	<u>Action</u>	<u>Comments</u>
(various)	Input and output products are specified by Command File.	

LOAD LINE

```
RLDR LAWEB GCOMF GTIME JDATE IGCOD IOPNB CLSB IASCL IOPNP CLSP^
WMOHD PRNUM HDRB PLTLB PRB WLL SHPLOC WWSBY WWX PCMAN CANDEC^
GETP OBTIM SWIND ATEMP PKWND PRES MASK GPIX BNSCH IW2N N2IWX^
PRP PCGR CGRDEC CGRSR BACKUP GBLK0 ICGSKY ICGWX ICGVS ICGWN ICGWV^
ICGT ICGPR POMR OMRDEC LATLON SWX WTEMP BOUYWV PBOY LAKLOC DSHP^
GETPS PSHP PPRES ALTDEC PSYN PSA SADC SCNWX ADXX VSBYWX TEMPDP^
DRSP DECVIS BKLCHK WIND INTGR SKY STRING INSTALL IFAH ICEL WX^
BFILL <AG TOP BG UTIL FORT>.LB
```

PROGRAM INSTALLATION

1. Move LAWEB.SV to APPL1 on DZ0 with a link to DP0.
2. Create a command file using the codes in Figure 6.
3. Most sites create a number of command files for various purposes and run the program through macros. This allows one command to run the LAWEB program followed by various combinations of PMOD, HCOPI and GENUTF.

LAWEB: DATA PROCESSING FOR THE GREAT LAKES

PART B: EXECUTION AND ERROR CONDITIONS

PROGRAM NAME: LAWEB

AAL ID:
REVISION NO.: 1.00

PROGRAM EXECUTION:

1. RUN:LAWEB (COMMANDFILE) where COMMANDFILE is the name of the appropriate Command File.

ERROR CONDITIONS

Messages from ADM

Meaning

1. JOB LAWEB COMPLETE PRODUCT CCCNNNXXX STORED

Dasher Messages

Meaning

- | | |
|----------------------------|----------------------------------|
| 1. CAN'T OPEN COMMAND FILE | Named file is not available |
| 2. CAN'T OPEN CCCNNNXXX. | System error opening named file. |

VI. Figures

Figure 1.: Sample Broadcast Text.

ZCZC CLELAWCLE
TTAA00 KCLE 120600

GREAT LAKES MARINE WEATHER BROADCAST
NATIONAL WEATHER SERVICE CLEVELAND OH
259AM EDT SUN MAY 12 1985

LAKE SUPERIOR

SAULT STE MARIE	110/08	
WHITE FISH PT	180/12	
GRAND MARIAS MI	170/14	
MARQUETTE MI	MMM/MM	
STANNARD ROCK	170/11	
COPPER HARBOR	150/10	
PORTAGE MI	MMM/MM	
ONTONAGON MI	200/12	0 FT WAVES
DEVILS ISLAND	100/09	
DULUTH MN	MMM/MM	
GRAND MARIAS MN	060/07	
ROCK OF AGES	MMM/MM	
PASSAGE IS LT	MMM/MM	
THUNDER BAY ONT	060/09	VSBY 2 MI IN FOG
SLATE IS ONT	MMM/MM	
CARIBOU IS ONT	140/26	7 FT WAVES VSBY 2 MI IN RAIN

SHIPS AND BOUYS

LAT 47.3 LON 90.3	13 NNE OUTER ISLAND LT	140/15	
VSBY 2 MI IN MISSING			
LAT 47.0 LON 86.0	25 SSW CARIBOU ISLAND LT	160/16	5 FT WAVES
LAT 47.0 LON 91.1	25 E TWO HARBOURS RB	030/14	2 FT WAVES
VSBY 2 MI IN MISSING			
LAT 46.9 LON 85.5	21 WNW WHITEFISH POINT	150/19	2 FT WAVES
VSBY 2 MI IN HAZE			
LAT 47.5 LON 88.9	18 N KEWEENAW RB	170/30	5 FT WAVES
VSBY 2 MI IN FOG			
LAT 47.6 LON 89.3	18 E ROCK OF AGES	130/18	0 FT WAVES
VSBY 1 MI IN FOG			
LAT 47.9 LON 84.9	47 NE CARIBOU ISLAND LT	010/30	
VSBY 2 MI IN RAIN			
LAT 46.5 LON 84.3	34 ESE WHITEFISH POINT	100/04	0 FT WAVES

LAKE MICHIGAN

CHARLEVOIX MI	MMM/MM
FRANKFORT	MMM/MM
LUDINGTON	MMM/MM
MUSKEGON	MMM/MM

SAINT JOSEPH	MMM/MM
MICHIGAN CITY	MMM/MM
DUNNE CRIB IL	MMM/MM
KENOSHA WI	180/12
MILWAUKEE WI	MMM/MM
SHEBOYGAN WI	180/22
SHEBOYGAN WI	180/21

BRK WATER LT

TWO RIVERS WI	200/15	2 FT WAVES
STURGEON BAY	MMM/MM	
GILLS ROCK WI	MMM/MM	
GREEN BAY WI	250/05	
ESCANABA MI	140/08	
SEUL-CHOIX PT	160/05	

SHIPS AND BOUYS

BUOY	20 NNW SOUTH MANITOU LT	220/10	
BUOY	40 ESE MILWAUKEE RB	160/12	2 FT WAVES
LAT 45.7 LON 86.0	42 N SOUTH MANITOU LT	190/10	2 FT WAVES
LAT 44.0 LON 87.1	27 W BIG SABLE LIGHT	200/20	7 FT WAVES
USBY 2 MI IN RAIN			
LAT 45.4 LON 85.4	29 SSW WHITE SHOAL LIGHT	160/06	3 FT WAVES
LAT 45.8 LON 85.9	.38 W WHITE SHOAL LIGHT	160/18	

LAKE HURON

ST CLAIR SHORES	200/10
PORT HURON MI	160/10
PORT HOPE MI	150/05
SAGINAW RIVER	230/04
TAWAS MI	180/08
STURGEON POINT	MMM/MM
ALPENA MI	170/06
PRESQUE IS LT	150/07
CHEBOYGAN MI	170/03
DETOUR PASSAGE	000/00
ST IGNACE MI	MMM/MM
GREAT DUCK IS	MMM/MM
GORE BAY ONT	130/07
COVE IS ONT	MMM/MM
WIARTON ONT	180/12
GODERICH ONT	MMM/MM
SARNIA ONT	MMM/MM

SHIPS AND BOUYS

BUOY	28 NE THUNDER BAY IS LT	190/10	2 FT WAVES
BUOY	27 NNE HARBOR BEACH RB	160/12	0 FT WAVES
LAT 44.9 LON 82.9	18 ESE THUNDER BAY IS LT	160/22	3 FT WAVES
LAT 45.1 LON 83.1	10 ENE THUNDER BAY IS LT	170/20	3 FT WAVES
LAT 44.5 LON 82.8	36 N HARBOR BEACH RB	160/18	3 FT WAVES
LAT 44.0 LON 82.6	7 NE HARBOR BEACH RB	170/13	0 FT WAVES
LAT 44.0 LON 82.5	11 ENE HARBOR BEACH RB	180/22	
LAT 44.1 LON 82.5	15 NE HARBOR BEACH RB	150/19	3 FT WAVES

USBY 2 MI IN MISSING

LAT 43.8 LON 82.5	11 ESE HARBOR BEACH RB	170/16	0 FT WAVES
LAT 44.8 LON 83.0	18 SE THUNDER BAY IS LT	170/20	3 FT WAVES
VSBY 2 MI IN MISSING			
LAT 45.1 LON 83.1	10 ENE THUNDER BAY IS LT	160/19	3 FT WAVES
LAT 45.3 LON 80.0	72 E COVE ISLAND LIGHT	120/05	
LAT 45.8 LON 81.2	37 NE COVE ISLAND LIGHT	160/14	0 FT WAVES

LAKE ERIE

BUFFALO	000/00
DUNKIRK BEACH	140/06
ERIE PA	160/10
ASHTABULA	MMM/MM
ASHTABULA LT	160/12
CLE W PIERHD LT	MMM/MM
CLE LK FNT	MMM/MM
LORAIN OH	200/06
MARBLEHEAD OH	180/07
SOUTH BASS IS.	180/11
TOLEDO OH	180/10
BELLE ISLE	180/10
SOUTHEAST SHOAL	170/14
LONG POINT	180/17
PORT COLBORNE	MMM/MM
LONDON ONTARIO	130/05

SHIPS AND BOUYS

BUOY	13 SSE SOUTHEAST SHOAL LT	170/08	0 FT WAVES
LAT 41.9 LON 82.0	22 E SOUTHEAST SHOAL LT	180/14	2 FT WAVES
LAT 42.3 LON 80.4	30 NE ASHTABULA RB	180/16	2 FT WAVES
LAT 42.3 LON 80.6	26 NNE ASHTABULA RB	000/17	2 FT WAVES
LAT 41.8 LON 82.1	19 ESE SOUTHEAST SHOAL LT	170/15	2 FT WAVES
LAT 41.6 LON 82.2	22 SE SOUTHEAST SHOAL LT	200/14	
LAT 41.9 LON 82.9	18 W SOUTHEAST SHOAL LT	170/15	
LAT 42.7 LON 79.4	15 NW DUNKIRK LIGHT	160/11	0 FT WAVES
LAT 42.5 LON 80.0	35 W DUNKIRK LIGHT	200/14	2 FT WAVES
LAT 42.8 LON 79.2	18 E DUNKIRK LIGHT	150/03	0 FT WAVES

LAKE ONTARIO

ALEXANDRIA BAY	000/00
GALLOO ISLAND	100/05
OSWEGO	MMM/MM
ROCHESTER	160/03
NIAGARA	MMM/MM
PORT WELLER ONT	MMM/MM
TORONTO ISLAND	MMM/MM
TRENTON ONT	000/00
MAIN DUCK IS	MMM/MM

SHIPS AND BOUYS

LAT 43.3 LON 78.9	18 WSW THIRTY MILE PT LT	120/07	0 FT WAVES
LAT 43.7 LON 77.2	25 WSW MAIN DUCK ISLAND	100/12	
LAT 43.7 LON 77.2	25 WSW MAIN DUCK ISLAND	150/12	2 FT WAVES

LAT 43.7 LON 78.8 22 NW THIRTY MILE PT LT 050/14 0 FT WAVES
VSBY 2 MI IN MISSING

EXTRA PRESSURES SECTION

SSM 2969	CMX MMMM	DLH 2934	MKG 2970	NDW 2969	GRB 2958	APN 2972
BUF 2990	TOL 2983	SYR 2994	INL 2925	MSP 2933	HON 2930	DIK MMMM
GTF 2988	CYS 2988	GRI 2961	DSM 2943	STL 2970	ABQ 2990	ABI 2975
OKC 2969	MEI 2993	BNA 2993	CVG 2989	DCA 3002	YQT 2945	YZE 2975
YVV 2981	YZR MMMM	YXU 2986	YTR 2992	YUL 3002	YXR MMMM	WYW MMMM
YWG 2936	YXE 2953	CLE 2989				

HIGHEST PRESSURE.. 3002 AT YUL

LOWEST PRESSURE.. 2925 AT INL

NNNN

Figure 2.: Sample Plot File.

```

ZCZC *CLEPLTLAW
CLEPLTLAW0010200020481536285014250097501688150702851519
1212,0720,000002,06Z 05/12/85;
1305,0814,000002,,,8,11008,058,57,,,,,,,,,,,,,40;
1293,0816,000002,,,M,18012,,,,,,,,,,,,,;
1278,0809,000002,,,M,17014,,,,,,,,,,,,,;
1255,0813,000002,,,M,17011,,,,,,,,,,,,,;
1243,0817,000002,,,M,15010,,55,49,,,,,,,,,,,,,;
1202,0796,000002,,,M,10009,949,61,,,,,,,,,,,,,;
1204,0813,000002,,,M,06007,,46,47,,,,,,,,,,,,,;
1215,0831,000102,,,8,06009,976,50,50,10,32,,,,,,,,,;
1275,0824,000202,,,8,14026,,,61,32,,,,,,,,,7,,;
1207,0803,000002,,,0,14015,,50,,32,,,,,,,,,;
1275,0816,000202,,,8,16016,,52,,29,,,,,,,,,5,,34;
1196,0793,000002,,,4,03014,,46,,32,,,,,,,,,2,,39;
1283,0816,000102,,,3,15019,,48,,5,32,,,,,,,,,2,,36;
1227,0813,000102,,,1,17030,,57,,11,32,,,,,,,,,5,,39;
1220,0814,000102,,,3,13018,,50,,12,16,,,,,,,,,0,,41;
1284,0842,000202,,,5,01030,,48,,61,32,,,,,,,,,34;
1304,0814,000002,,,8,10004,,81,,,,,,,,,0,,41;
1280,0706,000002,,,8,18012,047,57,,,,,,,,,,,,,;
1273,0715,000102,,,8,,,968,,5,58,,,,,,,,,44,,4;
1272,0733,000102,,,8,18022,051,54,,5,60,,,,,,,,,45;
1272,0733,000002,,,M,18021,027,51,,,,,,,,,,,,,;
1272,0743,000102,,,8,20015,979,51,,10,57,,,,,,,,,2,,45;
1260,0748,000302,,,8,25005,015,64,54,80,,,,,,,,,;
1285,0792,000002,,,M,16005,058,50,48,,,,,,,,,,,,,;
1283,0776,000002,,,M,22010,045,46,,,,,,,,,,,,,37;
1289,0712,000002,,,M,16012,055,47,,,,,,,,,,,,,2,,38;
1285,0787,000002,,,8,19010,,55,,58,,,,,,,,,2,,43;
1280,0742,000302,,,9,20020,,47,,82,32,,,,,,,,,7,,38;
1296,0783,000002,,,3,16006,,57,,,,,,,,,,,,,3,,36;
1285,0790,000002,,,9,16018,,50,,58,,,,,,,,,43;
1371,0736,000002,,,8,20010,108,72,,,,,,,,,,,,,;
1366,0748,000102,,,4,16010,088,67,,5,58,,,,,,,,,;
1350,0768,000002,,,M,15005,,75,56,,,,,,,,,,,,,;
1337,0753,000002,,,8,23004,091,65,,,,,,,,,,,,,62;
1338,0769,000102,,,4,18008,146,56,,5,60,,,,,,,,,;
1329,0787,000102,,,6,17006,062,62,59,10,58,,,,,,,,,;
1316,0805,000002,,,M,00000,,,,,,,,,,,,,;
1337,0811,000002,,,6,13007,077,57,52,,,,,,,,,,,,,;
1370,0796,000102,,,0,18012,092,70,57,5,60,,,,,,,,,;
1338,0797,000002,,,M,19010,074,43,,,,,,,,,,,,,2,,36;
1353,0777,000002,,,M,16012,088,45,,,,,,,,,,,,,0,,36;
1340,0787,000202,,,4,16022,,48,,42,58,,,,,,,,,3,,37;
1335,0790,000002,,,3,17020,,50,,,,,,,,,,,,,3,,43;
1345,0779,000002,,,7,16018,,54,,58,,,,,,,,,3,,37;
1353,0769,000002,,,3,17013,,54,,58,,,,,,,,,0,,50;
1355,0770,000002,,,9,18022,,54,,,,,,,,,,,,,;
1354,0772,000002,,,5,15019,,50,,32,,,,,,,,,3,,;

```

1358,0765,000002,,,3,17016,,54,,,58,,,,,,0,,43;
1339,0784,000002,,,3,17020,,48,,,32,,,,,,3,,;
1335,0790,000002,,,8,16019,,46,,,58,,,,,,3,,39;
1381,0816,000002,,,3,12005,,64,,,,,,54;
1358,0818,000002,,,1,16014,083,46,,,58,,,,,,0,,42;
1424,0771,000002,,,0,00000,139,65,,,,,,;
1421,0759,000002,,,M,14006,113,71,,,,,,;
1413,0745,000002,,,4,16010,122,74,,,,,,;
1384,0715,000002,,,0,20006,115,71,,,,,,;
1374,0713,000102,,,8,18007,105,72,,5,,,,,,;
1371,0714,000002,,,M,18011,106,69,,,,,,;
1360,0712,000002,,,8,18010,122,71,,,,,,;
1363,0729,000002,,,0,18010,098,72,,,,,,56;
1376,0721,000002,,,8,17014,,,,,58,,,,,,;
1409,0755,000002,,,8,18017,,,,,,;
1386,0757,000002,,,6,13005,111,63,57,,,,,,;
1378,0719,000002,,,M,17008,112,64,,,,,,0,,59;
1383,0726,000002,,,1,18014,,57,,,,,,2,,55;
1405,0747,000102,,,8,18016,,68,,15,,,,,,2,,50;
1402,0745,000002,,,1,00017,,,,,,2,,45;
1382,0723,000002,,,1,17015,,64,,,,,,2,,36;
1382,0718,000002,,,1,20014,,63,,,,,,;
1367,0720,000002,,,1,17015,,66,,,,,,59;
1417,0763,000102,,,4,16011,,59,,15,,,,,,0,,50;
1410,0754,000102,,,3,20014,,66,,15,,,,,,2,,48;
1419,0767,000002,,,7,15003,,61,,,58,,,,,,0,,55;
1451,0812,000002,,,M,10005,133,60,,,,,,;
1441,0789,000102,,,4,16003,122,62,,5,57,,,,,,;
1432,0809,000102,,,3,00000,133,55,52,5,57,,,,,,;
1419,0780,000002,,,5,12007,,63,,,,,,0,,37;
1441,0802,000002,,,0,10012,,54,,,,,,;
1441,0802,000402,,,3,15012,,61,,25,,,,,,2,,43;
1416,0790,000002,,,0,05014,,46,,,32,,,,,,0,,39;
END OF DATA
NNNN

Figure 3.: Sample Graphic

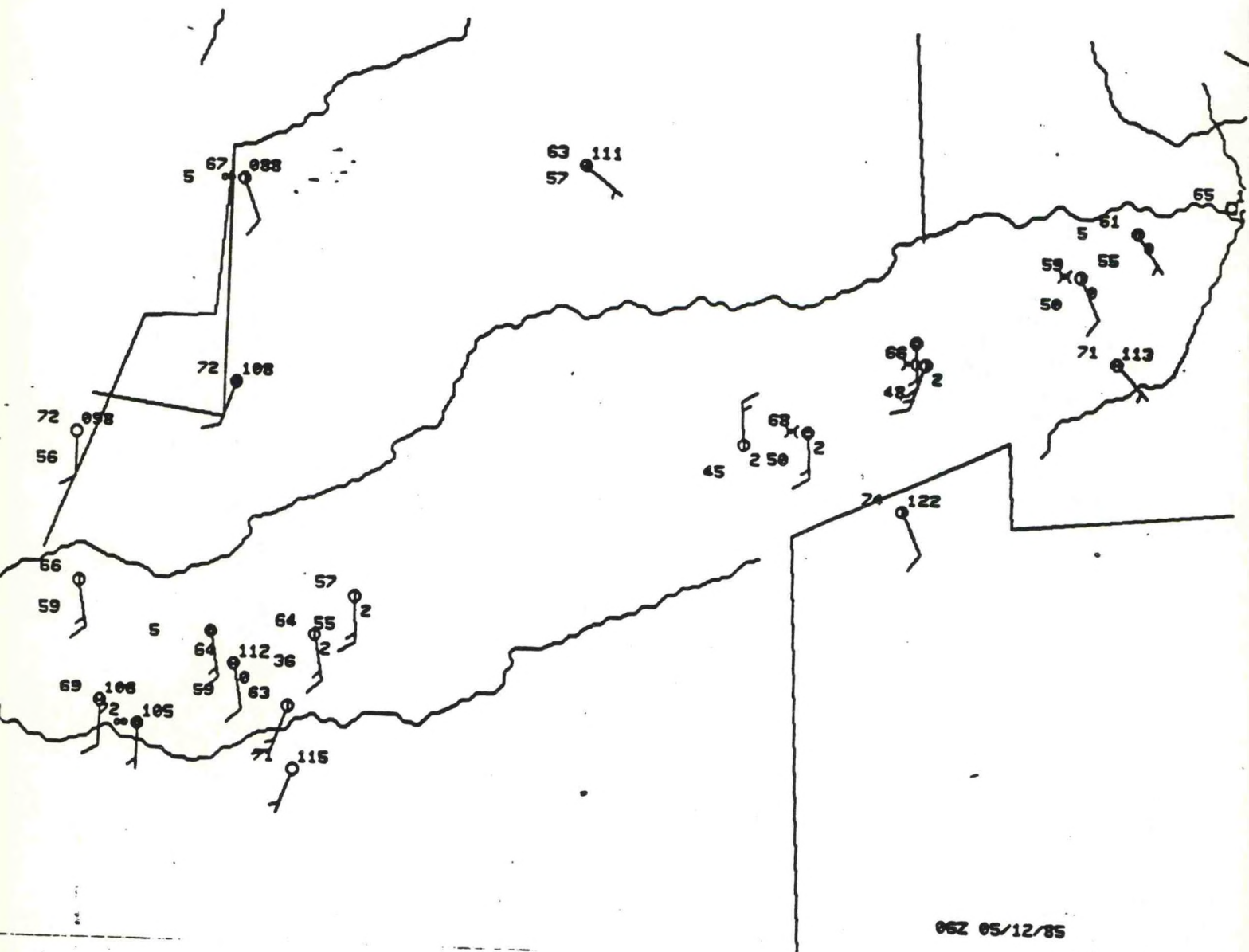


Figure 4.: Station Model from Graphic

	T T		P P P
V V	W W	N	
	TdTd		H H
	TwTw		

Where TT is air temperature (F),
PPP is sea level pressure or altimeter setting (mb),
VV is visibility in miles and fractions,
WW is weather code,
N is cloud cover symbol,
TdTd is dew point (F),
HH is wave height (ft), and
TwTw is water temperature (F).

Figure 5.: Sample Command File

01CLEWRKLAW000
05TTAA00 KCLE

GREAT LAKES MARINE WEATHER BROADCAST
NATIONAL WEATHER SERVICE CLEVELAND OH
06

03CLEPLTLAW000
1612120720
12CLELAWCG9 CLELAWWMI ARBLAWWLC YYZLAWYYZ 0

LAKE SUPERIOR

07SAULT STE MARIE ARBCGRSSM 44Y
15WHITE FISH PT ARBOMRSSM 26Y
15GRAND MARIAS MI ARBOMRSSM 27Y
07MARQUETTE MI ARBCGRSSM 34Y
10STANNARD ROCK CLEBOYCM3 STDM4
08COPPER HARBOR ARBSAOP59 P59
07PORTAGE MI MSPCGRDLH 32Y
15ONTONAGON MI ARBOMRSSM Y64
10DEVILS ISLAND CLEBOYCM3 DISW3
07DULUTH MN MSPCGRDLH 30Y
08GRAND MARIAS MN MSPSAOP61 P61
10ROCK OF AGES CLEBOYCM3 ROAM4
10PASSAGE IS LT CLEBOYCM3 PILM4
08THUNDER BAY ONT YYZSAOYQT YQT
07SLATE IS ONT YYZCGRYYZ WNU
07CARIBOU IS ONT YYZCGRYYZ WDY

SHIPS AND BOUYS

1145001 CLEBOYGL5 CLEBOYGL6 0
1145004 CLEBOYGL5 CLEBOYGL6 0
1145006 CLEBOYGL5 CLEBOYGL6 0
1301

LAKE MICHIGAN

07CHARLEVOIX MI ARBCGRSSM Y09
07FRANKFORT ARBCGRMKG 14C
07LUDINGTON ARBCGRMKG 17C
07MUSKEGON ARBCGRMKG 19C
07SAINT JOSEPH ARBCGRMKG 20C
07MICHIGAN CITY ARBCGRMKG 18C
09DUNNE CRIB IL CHISSMDUK 74465
07KENOSHA WI MKECGRMKE 16C
07MILWAUKEE WI MKECGRMKE 15C
07SHEBOYGAN WI MKECGRMKE 21C
10SHEBOYGAN WI CLEBOYCM3 SGNW3
BRK WATER LT
07TWO RIVERS WI MKECGRMKE C58

07STURGEON BAY	MKECGRMKE	0Y2
15GILLS ROCK WI.	MKEOMRGRB	GSRW3
08GREEN BAY WI	MKESAOGRE	GRB
09ESCANABA MI	ARBSSMESC	72648
08SEUL CHOIX PT	ARBSAOP75	P75

SHIPS AND BOUYS

1145002	CLEBOYGL5	CLEBOYGL6	0
1145007	CLEBOYGL5	CLEBOYGL6	0
1302			

LAKE HURON

07ST CLAIR SHORES	ARBCGRDTW	41Y
07PORT HURON MI	ARBCGRDTW	33G
08PORT HOPE MI	ARBSAOP58	P58
07SAGINAW RIVER	ARBCGRDTW	30G
07TAWAS MI	ARBCGRDTW	39Y
15STURGEON POINT	ARBOMRAPN	SPTM4
08ALPENA MI	ARBSAOPAPN	APN
15PRESQUE IS LT	ARBOMRAPN	PRIM4
15CHEBOYGAN MI	ARBOMRAPN	CYGM4
15DETOUR PASSAGE	ARBOMRSSM	P55
07ST IGNACE MI	ARBCGRSSM	38Y
07GREAT DUCK IS	YYZCGRYYZ	WGG
08GORE BAY ONT	YYZSAOYZE	YZE
07COVE IS ONT	YYZCGRYYZ	WQV
08WIARTON ONT	YYZSAOYVV	YVV
08GODERICH ONT	YYZSAOWGD	WGD
08SARNIA ONT	YYZSAOYZR	YZR

SHIPS AND BOUYS

1145003	CLEBOYGL5	CLEBOYGL6	0
1145008	CLEBOYGL5	CLEBOYGL6	0
1303			

LAKE ERIE

07BUFFALO	BUFCGRBUF	19G
10DUNKIRK BEACH	CLEBOYCM3	DBLN6
07ERIE PA	BUFCGRBUF	25G
07ASHTABULA	BUFCGRBUF	20G
15ASHTABULA LT	CLEOMRCLE	ASH
15CLE W PIERHD LT	CLEOMRCLE	LIX
08CLE LK FNT	CLESAOBKL	BKL
07LORAIN OH	ARBCGRDTW	27G
07MARBLEHEAD OH	ARBCGRDTW	21G
10SOUTH BASS IS.	CLEBOYCM3	SBI01
07TOLEDO OH	ARBCGRDTW	24G
07BELLE ISLE	ARBCGRDTW	31G
07SOUTHEAST SHOAL	YYZCGRYYZ	WOR
07LONG POINT	YYZCGRYYZ	WLX
07PORT COLBORNE	YYZCGRYYZ	WVH
08LONDON ONTARIO	YYZSAOYXU	YXU

SHIPS AND BOUYS
1145005 CLEBOYGL5 CLEBOYGL6 0
1304

LAKE ONTARIO

07ALEXANDRIA BAY BUFCGRBUF 6B3
10GALLOO ISLAND CLEBOYCM3 GLLN6
07OSWEGO BUFCGRBUF 28G
07ROCHESTER BUFCGRBUF 26G
07NIAGARA BUFCGRBUF 13G
07PORT WELLER ONT YYZCGRYYZ WPW
08TORONTO ISLAND YYZSAOYTZ YTZ
08TRENTON ONT YYZSAOYTR YTR
07MAIN DUCK IS YYZCGRYYZ WDK

SHIPS AND BOUYS
1305

EXTRA PRESSURES SECTION

14ARBSAOSSM ARBSAOCMX MSPSAODLH ARBSAOMKG CHISAOMDW MKESAOGRB ARBSAOAPN 0
14BUFSAOBUF CLESAOTOL BUFSASOYR MSPSAOINL MSPSAOMSP FSDSAOHON BISSAODIK 0
14GTFSAGTF CYSSAOCYS OMASAOGRI DSMSAODSM STLSAOSTL ABQSAOABQ FTWSAOABI 0
14OKCSAOKC JANSAMEI MEMSAOBNA CLESAOCVG WBCSAODCA YYZSAOYQT YYZSAOYZE 0
14YYZSAOYVV YYZSAOYZR YYZSAOYXU YYZSAOYTR WULSAOYUL YYZSAOYXR YYZSAOWYW
140
02CLEWRKLA
04CLEPLTLA

Figure 6.: Command File Codes

CODE	FUNCTION	FIELDS USED
01	Open a new broadcast product	9-byte key name begins in column 3 3-byte addressee begins in column 13
02	Close current broadcast product	9-byte key name begins in column 3
03	Open a new plot product	9-byte key name begins in column 3 3-byte addressee begins in column 13
04	Close current plot product	9-byte key name begins in column 3
05	Write WMO header to broadcast product	11-byte code begins in column 3 6-digit date-time group is appended
06	Write 26- or 27-byte mass media header to broadcast product	none
07	Decode Coast Guard (CGR) report and write to plot and/or broadcast products	15-byte name begins in column 3 9-byte key name begins in column 19 3-byte ID begins in column 30
08	Decode Surface Airways (SA) report and write to plot and/or broadcast products	15-byte name begins in column 3 9-byte key name begins in column 19
09	Decode Surface Synoptic (SM) report and write to plot and/or broadcast products	15-byte name begins in column 3 9-byte key name begins in column 19 5-byte ID begins in column 30
10	Decode Coastal Marine (CMAN) report and write to plot and/or broadcast products	15-byte name begins in column 3 9-byte key name begins in column 19 5-byte ID begins in column 30
11	Decode Buoy (BOY) report and write to plot and/or broadcast products	5-byte ID begins in column 3 9-byte key names begin in columns 9, 19, and 69
12	Decode ship (LAW) reports and store them internally	9-byte key names begin in columns 3, 13, and 63

- | | | |
|----|--|---|
| 13 | Write ship reports
for a lake | 2-byte lake number begins in column 3
01 = LS, 02 = LM, 03 = LH, 04 = LE,
and 05 = LO |
| 14 | Decode a group of
Surface Airways (SA)
observations and
print them as a line
of the Added Pres-
sures Section | 9-byte key names begin in columns
3, 13, and 63
If no key names are found, the highest
and lowest pressures are printed |
| 15 | Decode Extra Marine
(OMR) report and
write to plot and/or
broadcast products | 15-byte name begins in column 3
9-byte key name begins in column 19
5-byte ID begins in column 30 |
| 16 | Write plot label to
plot product | 4-byte horizontal pixel coordinate
begins in column 3
4-byte vertical pixel coordinate
begins in column 7 |

Figure 7.: Program and Data Flow

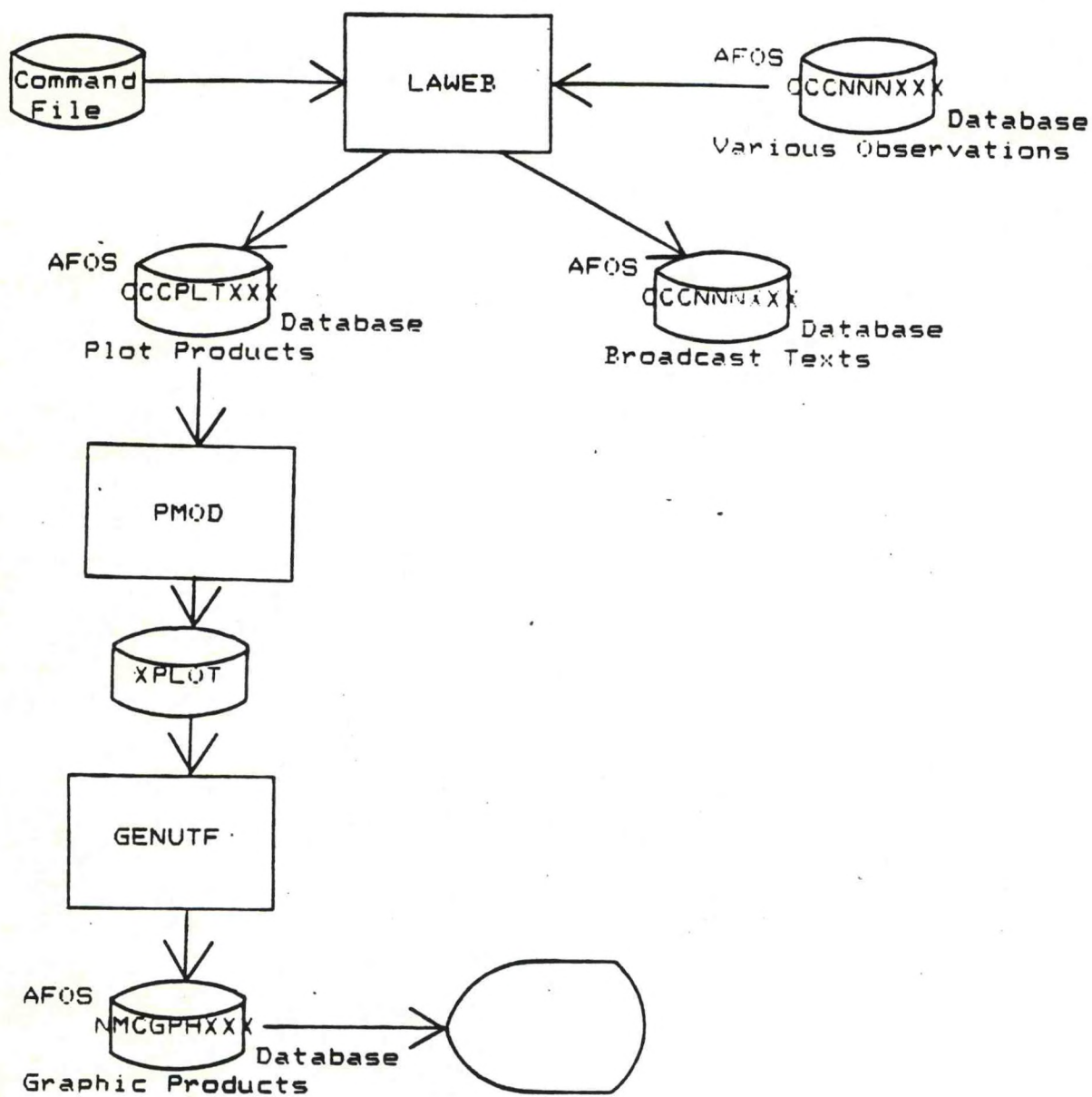


Figure 8.: Software Structure and Load Line

ERH CP 85-???
May, 1985

MAIN PROGRAM
LAWEB

SUBROUTINES

GCOMF

GTIME ----> JDATE

IGCOD

IOPNB

CLSB

IOPNP

CLSP

WMOHD ----> PRNUM ----> IASCL

HDRB

PCGR	----	CGRDEC	----	CGRSR	----	GZLK0
						BACKUP
						MASK
				ICGSKY	----	MASK
				ICGWX	----	MASK
				ICGVS	----	MASK
				MASK		
				ICGWN	----	MASK
				ICGWV	----	MASK
				ICGT	----	MASK
				ICGPR	----	MASK
		CANDEC	----	GETP	----	GZLK0
						BACKUP
						MASK
						OBTIM ----> MASK
				ATEMP	----	MASK
				BOUYWV	----	MASK
				LATLON	----	MASK
				MASK		
				PKWND	----	MASK
				PRES	----	MASK
				SWIND	----	MASK
				WTEMP	----	MASK
				SWX	----	MASK
		NZIWX				
		PRB	----	WLL	----	PRNUM
						SHPLC

```

      PRNUM
      WWSBY ----> PRNUM
      WWX
      GPIX ----> BNSCH
      IWX2N
      PRP ----> PRNUM

PSA ----> SADC ----> GBLK0
      ADXX ----> WWSBYWX
      TEMPDP
      DRSP
      DECVIS
      BLKCHK
      WIND
      INTGR
      SKY
      STRING
      INTALL
      IFAH
      ICEL
      WX
      BFILL

      PRB ----> SCNWX ----> MASK
      WLL ----> PRNUM
      SHPLOC
      PRNUM
      WWSBY ----> PRNUM
      WWX
      GPIX ----> BNSCH
      PRP ----> PRNUM

PSYN ----> CANDEC ----> GETP ----> GBLK0
      BACKUP
      MASK
      OBTIM ----> MASK
      ATEMP ----> MASK
      BOUYWV ----> MASK
      LATLON ----> MASK
      MASK
      PKWND ----> MASK
      PRES ----> MASK
      SWIND ----> MASK
      WTEMP ----> MASK
      SWX ----> MASK
      PRB ----> WLL ----> PRNUM
      SHPLOC
      PRNUM
      WWSBY ----> PRNUM
      WWX
      GPIX ----> BNSCH
      PRP ----> PRNUM

PCMAN ----> CANDEC ----> GETP ----> GBLK0

```



```

      :
      : PRNUM
      : WWSBY ----> PRNUM
      : WWX
      : PRP ----> PRNUM
      :
PPRES ----> :ALTDEC----> MASK
      : PRNUM ----> IASCL
      :
POMR ----> :OMRDEC----> :CGRSR ----> :GBLK0
      :
      : MASK
      : ICGWN ----> MASK
      : ICGWV
      : PRB ----> :WLL ----> :PRNUM
      :
      : SHPLOC
      : PRNUM
      : WWSBY ----> PRNUM
      : WWX
      : GPIX ----> BNSCH
      : PRP ----> PRNUM
      :
PLTLB ----> :PRNUM ----> IASCL

```

Load Line

```
RLDR LAWEB GCOMF GTIME JDATE IGCOD IOPNB CLSB IASCL IOPNP CLSPA
WMOHD PRNUM HDRB PLTLB PRB WLL SHPLOC WWSBY WWX PCMAN CANDECA
GETP OPTIM SWIND ATEMP PKWND PRES MASK GPIX BNSCH IWXZN NZIWX^
PRP PCGR CGRDEC CGRSR BACKUP GBLK0 ICGSKY ICGWX ICGVS ICGWN ICGWA^
ICGT ICGPR POMR OMRDEC LATLON SWX WTEMP BOUYWV PBOY LAKLOC DSHP^
GETPS PSHP PPRES ALTDEC PSYN PSA SADC SCNWX ADXX VSBYWX TEMPDPA
DRSP DECVIS BKLCHK WIND INTGR SKY STRING INTALL IFAH ICEL WX^
BFILL <AG TOP BG UTIL FORT>.LB
```

Program Listings

RLDR LABEL GCOMF GTIME JDATE IGCOD IOPNB CLSB IASCL IOPNP CLSP WPHND+
 PRNUM HDRB PLTLB PRB LLL SHPLOC WVSBY LAK PCMAN CANDEC GETP OBTIM+
 SUIND ATEPH PKUND PRES MASK GPIX BNSCH ILQZN NZILX PRP PCGR CGRDEC+
 CGRSR BACKUP GBLKB ICGSKY ICGLX ICGVS ICGLN ICGLV ICGT ICGPR POMR+
 ORRDEC LATLOM SUX UTEPH BOUYLV PBOY LAKLOC DSHP GETPS PSHP PPRES+
 ALTDEC PSYN PSA SABC SCHUX ADXX VSBYX TEMPD DRSP DECVIS BLKCHK WIND+
 INTER SKY STRING INTALL IFAN ICEL LX BFILL+
 <AG TOP DG UTIL FORT>.LB

READS MARINE DATA FROM A LARGE NUMBER OF DATABASE PRODUCTS IN A WIDE
 VARIETY OF FORMATS. PRODUCES TWO TYPES OF DATABASE PRODUCT AS OUTPUT.
 FIRST, A BROADCAST TEXT WHICH IS READ OVER THE GREAT LAKES VHF RADIO
 SYSTEM, AND, SECOND, A PLOT FILE WHICH IS COMPATIBLE WITH THE PHOD
 SYSTEM.

EXTERNALS

GCOMF	GTIME	IGCOD	IOPNB	CLSB
IOPNP	CLSP	WPHND	HDRB	PCGR
PSA	PSYN	PCMAN	PBOY	DSHP
PSHP	PPRES	POMR	PLTLB	
RDL (UTIL)	STBYT (TOP)	LRS (UTIL)	KLOSE (UTIL)	

CHANNELS/FILES

ALL FILES ARE OPENED ON RDOS CHANNELS.
 ICC COMMAND LINES ARE READ FROM THIS CHANNEL TO CONTROL THE
 PROGRAM'S EXECUTION.
 IBC LINES OF BROADCAST TEXT ARE WRITTEN TO THIS CHANNEL
 IPC LINES OF THE PLOT FILE ARE WRITTEN TO THIS CHANNEL

VARIABLES

MONTH, IDAY, IYEAR VALID DATE OF ALL DATA
 IHR VALID TIME OF ALL DATA
 CTIM "CUT TIME" A REAL VARIABLE CONTAINING THE NUMBER
 OF MINUTES SINCE MIDNIGHT OF JAN. 1 OF THIS YEAR
 (JULIAN TIME). SET TO 30 MINUTES BEFORE THE DATE
 AND TIME IN THE ABOVE VARIABLES.
 ICLIN A LINE FROM THE COMMAND FILE.
 ICODE TAKEN FROM THE FIRST TWO ASCII CHARACTERS OF ICLIN.
 SPECIFIES THE DATA FORMATS TO BE HANDLED.

EXIT ONLY THROUGH THE STOP COMMAND AFTER LINE 9900.

DIMENSION ICLIN(41)

IBC = -1

IPC = -1

CALL GCOMF(ICC,9900)

:OPEN COMMAND FILE.

CALL GTIME(MONTH,IDAY,IYEAR,IHR,CTIM) :COMPUTE TIMES.

10 CALL RDL(ICC,ICLIN,M,IER) :READ A COMMAND LINE.

IF (IER.NE.1) GOTO 9900

ICODE = IGCOD(ICLIN(1)) :LOOK FOR COMMAND CODE.

IF (ICODE.GT.8) GOTO 50

CALL STBYT("<8><12>".ICLIN,M+1) :ECHO LINE.

M = M + 1

IF (IBC.NE.-1) CALL LRS(IBC,ICLIN,M,IER)

GOTO 10

50 GOTO (100, 200, 300, 400, 500, 600, 700, 800, 900,1000,
 + 1100,1200,1300,1400,1500,1600),ICODE

100 IBC = IOPNB(ICLIN)

GOTO 10

200 CALL CLSB(ICLIN,IBC)

GOTO 10

300 IPC = IOPNP(ICLIN)

GOTO 10

400 CALL CLSP(ICLIN,IPC)

GOTO 10

500 CALL WPHND(ICLIN,IDAY,IHR,IBC)

GOTO 10

600 CALL HDRB(IBC)

GOTO 10

700 CALL PCGR(IBC,IPC,ICLIN,IHR,CTIM)

GOTO 10

800 CALL PSA(IBC,IPC,ICLIN,CTIM)

GOTO 10

900 CALL PSYN(IBC,IPC,ICLIN,IHR,CTIM)

GOTO 10

1000 CALL PCMAN(IBC,IPC,ICLIN,IHR,CTIM)

GOTO 10

1100 CALL PBOY(IBC,IPC,ICLIN,IHR,CTIM)

GOTO 10

1200 CALL DSHP(ICLIN,IHR,CTIM)

GOTO 10

1300 CALL PSHP(IBC,IPC,ICLIN)

GOTO 10

1400 CALL PPRES(IBC,ICLIN,IHR,CTIM)

GOTO 10

1500 CALL POMR(IBC,IPC,ICLIN,CTIM)

GOTO 10

1600 CALL PLTLB(IPC,ICLIN,IHR,MONTH,IDAY,IYEAR)

GOTO 10

9900 CALL KLOSE(ICC,IER)

STOP

```

PARAMETER IETX=3          ;END OF TRANSMISSION
PARAMETER IETB=131        ;END OF TRANSMISSION BLOCK
PARAMETER IBLK=32         ;BLANK
PARAMETER INL=10          ;LINE FEED
PARAMETER ISLSH=47        ;SLASH
PARAMETER MINUS=45        ;MINUS SIGN

```

```

SUBROUTINE ADXX(OBS, IOBS, DOBS, JOBS, CAN, NOMIX, IER)

```

```

MARCH 1984    HERM PERROTTI    TDL    ECLIPSE S230
MODIFIED FROM AIRDX MARCH 1985    MATT PEROUTKA USFO CLE
PURPOSE

```

```

THIS SUBROUTINE CONVERTS SURFACE AIRWAYS OBSERVATIONS FROM
UNPACKED ASCII TO COMBINED INTEGER AND ASCII FORMAT
DATA SET USE

```

```

NONE

```

```

VARIABLES

```

```

OBS( ) = UNPACKED ASCII OBSERVATION (INPUT-INTEGER)
IOBS = DIMENSION OF OBS( )
DOBS( ) = DECODED OBSERVATION (OUTPUT-INTEGER)
JOBS = DIMENSION OF DOBS( )
CAN = CANADIAN OBSERVATION FLAG (INPUT-INTEGER)
      1-CANADIAN/0-NOT
NOMIX = MIXING RATIO CALCULATION FLAG (INPUT - LOGICAL)
      .TRUE. - NO CALCULATION/ .FALSE. - DO IT
IER = ERROR CONDITIONS RETURNED (OUTPUT)
      <=0 = FATAL DECODING ERROR
      1 = GOOD DECODING
      >2 = NON-FATAL DECODING ERROR
CP = MOST CURRENT POSITION OF CHARACTER POINTER
      (INTEGER)
OLDCP = INITIAL POSITION OF CHARACTER POINTER
      (INTEGER)
LAST = INDICATED END OF CURRENT OBSERVATION

```

```

GENERAL COMMENTS

```

```

NOTE: SEE TDL CP SERIES FOR MORE DETAILED DISCUSSION OF
FORMATS AND USE OF THIS SUBROUTINE.

```

```

COMMON EQUIVALENCED - NEEDED FOR LPRINT ONLY

```

```

LOGICAL LCHK,CANADA,AMOS,MANNED,MEX,NOMIX,LPRINT,NOSLP
INTEGER OBS(IOBS),DOBS(JOBS),CP,OLDCP,CAN
AMOS=.FALSE.          ;AMOS/AMOS STATION CHECK
LCHK=.FALSE.          ;RETURN FROM SUB STRING
MANNED=.FALSE.        ;CHECK FOR MANNED AMOS
CANADA=CAN.EQ.1       ;CANADIAN STATION CHECK

```

```

BYPASS LMD HEADER

```

```

IBEG=44

```

```

IEND=58

```

```

DO 150 I=IBEG,IEND

```

```

CP=1

```

```

IF(OBS(I).EQ.ICR)GOTO 160      ;CHECKING FOR CARRIAGE RETURN

```

```

150 CONTINUE

```

```

DOBS(3)=2

```

```

IER=DOBS(3)

```

```

RETURN

```

```

;COULDN'T GET PAST LMD HEADER

```

```

SEARCH FOR END OF OBS

```

```

160 CP=CP+1

```

```

IF(OBS(CP).EQ.INL)CP=CP+1

```

```

;LINE FEED? - BUMP POINTER

```

```

IF(OBS(CP).EQ.ICR)CP=CP+1

```

```

;CARR RETURN? - BUMP POINTER.

```

```

IF(OBS(CP).EQ.INL)CP=CP+1

```

```

;LINE FEED? - BUMP POINTER

```

```

DO 170 I=CP,IOBS

```

```

LAST=I

```

```

KOBS=OBS(I)

```

```

IF(KOBS.EQ.IEQL.OR.KOBS.EQ.IETX.OR.KOBS.EQ.IETB)GOTO 175

```

```

170 CONTINUE

```

```

DOBS(3)=3

```

```

IER=DOBS(3)

```

```

RETURN

```

```

;COULDN'T FIND END OF OBS

```

```

175 IF(LAST.GT.IOBS)LAST=IOBS

```

```

INITIALIZE DOBS ARRAY TO ALL MISSING

```

```

CALL INTALL(DOBS,JOBS,IER)

```

```

SUBROUTINE FILTER IS USED TO RETRIEVE THE CITY

```

```

TEMPERATURE FOR THE VERIFICATION PROGRAM - TO

```

```

ACTIVATE IT , AIRDX HAS TO BE RECOMPILED WITH THE

```

```

"X" OPTION

```

```

X CALL FILTER(OBS,IOBS,DOBS,JOBS,LAST,CP,AMOS,$180)

```

```

IF(LAST.LE.20)GOTO 180

```

```

GOTO 185

```

```

RETURN IF NO DATA IN OBS

```

```

180 DOBS(3)=6

```

```

IER=DOBS(3)

```

```

RETURN

```

```

INITIALIZE ERROR RETURNS

```

```

185 DOBS(3)=1

```

```

IER=1

```

```

REMOVE ALL EXTRANIOUS BLANKS, CR, AND LINE FEEDS

```

```

K=1

```

```

DO 188 I=CP,LAST

```

```

ARRAY OVERFLOW/UNDERFLOW CHECK

```

```

KOBS=OBS(I)

```



```

C-----DOBS=DOBS(1)-1
C-----FILTER CR/LF
IF(KOBS.EQ.1CR.OR.KOBS.EQ.(INL)KOBS=1BLK
C-----FILTER BLANK/BLANK
IF(KOBS.EQ.1BLK.AND.LOBS.EQ.1BLK)GOTO 188
C-----FILTER BLANK/SLASH COMBINATION
IF(KOBS.EQ.1BLK.AND.LOBS.EQ.1SLSH)GOTO 188
C-----FILTER SLASH/BLANK COMBINATION
IF(KOBS.EQ.1SLSH.AND.KOBS.EQ.1BLK)GOTO 188
C-----FILTER MINUS/BLANK COMBINATION
IF(KOBS.EQ.MINUS.AND.KOBS.EQ.1BLK)GOTO 188
C-----STORE ONLY GOOD STUFF
189 DOBS(K)=KOBS
K=K+1
188 CONTINUE
C-----ARRAY OBS( ) NOW CONTAINS OBSERVATION WITHOUT HEADER
CP=1
C-----NEW LENGTH OF OBSERVATION
LAST=K
C-----FILL LAST TWO POSITIONS OF OBSERVATION WITH BLANKS
OBS(K-1)="<8> "
OBS(K)="<8> "
C-----CHECK FOR "BAD VERSION"
CALL BLKCHK(OBS, LAST, CP, $988)
CALL STRING(OBS, LAST, "<8>B<8>A<8>D<8> <8>V<8>E<8>R<8>S<8>I<8>O<8>H",
X 11, CP, OLDCP, LCHK)
IF(.NOT.LCHK)GOTO 190
DOBS(3)--1
IER--1
RETURN
C-----CALL SIGN
190 CALL BLKCHK(OBS, LAST, CP, $988)
C-----CHECK FOR BAD CALL SIGN - THIRD CHAR MINUS
CALL STRING(OBS, LAST, "<8>0<8>0<8>0", 3, CP, OLDCP, LCHK)
IF(.NOT.LCHK)GOTO 195
DOBS(3)--4
IER=DOBS(3)
RETURN
195 CALL STRING(OBS, LAST, "<8>0<8>0<8>0<8> ", 4, CP, OLDCP, LCHK)
IF(LCHK)GOTO 200
DOBS(3)--4
IER=DOBS(3)
RETURN
200 CALL PACK(OBS(OLDCP), 4, DOBS(1)) ;COULDN'T FIND CALL SIGN
C-----CHECK FOR "COR"
CALL BLKCHK(OBS, LAST, CP, $988)
CALL STRING(OBS, LAST, "<8>C<8>O<8>R", 3, CP, OLDCP, LCHK)
IF(LCHK)DOBS(5)=6
C-----CHECK FOR "BUOS"
CALL BLKCHK(OBS, LAST, CP, $988)
CALL STRING(OBS, LAST, "<8>B<8>U<8>O<8>S", 4, CP, OLDCP, LCHK)
IF(LCHK)AMOS=.TRUE.
C-----RECORD TYPE
CALL BLKCHK(OBS, LAST, CP, $988)
CALL STRING(OBS, LAST, "<8>S<8>A", 2, CP, OLDCP, LCHK)
IF(LCHK)DOBS(5)=DOBS(5)+1
IF(LCHK)GOTO 205
CALL STRING(OBS, LAST, "<8>R<8>S", 2, CP, OLDCP, LCHK)
IF(LCHK)DOBS(5)=DOBS(5)+2
IF(LCHK)GOTO 205
CALL STRING(OBS, LAST, "<8>S<8>P", 2, CP, OLDCP, LCHK)
IF(LCHK)DOBS(5)=DOBS(5)+3
IF(LCHK)GOTO 205
C-----CHECK FOR MISPELLED RECORD TYPE
CALL STRING(OBS, LAST, "<8>S<8>=", 2, CP, OLDCP, LCHK)
IF(LCHK)DOBS(5)=DOBS(5)+6
CALL STRING(OBS, LAST, "<8>L", 1, CP, OLDCP, LCHK)
IF(LCHK)DOBS(5)=DOBS(5)+7
C-----OBSERVATION TYPE COULD NOT BE DEFINED
IF(DOBS(5).EQ.8)DOBS(3)=4
IER=DOBS(3)
C-----CHECK FOR "COR"
205 CALL BLKCHK(OBS, LAST, CP, $988)
CALL STRING(OBS, LAST, "<8>C<8>O<8>R", 3, CP, OLDCP, LCHK)
IF(LCHK)DOBS(5)=6
C-----CHECK FOR "RTD"
CALL BLKCHK(OBS, LAST, CP, $988)
CALL STRING(OBS, LAST, "<8>R<8>T<8>D", 3, CP, OLDCP, LCHK)
C-----CHECK FOR TIME
CALL BLKCHK(OBS, LAST, CP, $988)
CALL STRING(OBS, LAST, "<8>0<8>0<8>0<8>0", 4, CP, OLDCP, LCHK)
IF(LCHK)CALL INTEG(OBS, LAST, OLDCP, 4, DOBS(4))
C-----OBSERVATION TIME COULD NOT BE IDENTIFIED
IF(DOBS(4).EQ.-99)DOBS(3)=5
IER=DOBS(3)
C-----SAVE TIME FOR LATER TEST FOR 3 OR 6 HOURLIES
ITIME=8
IF(DOBS(4).EQ.-99)GOTO 200
ITIME=DOBS(4)/100
MINUTE=DOBS(4)-ITIME*100
C-----ROUND TO NEAREST WHOLE HOUR

```


PURPOSE

ARGUMENT LIST

EXTERNALS

VARIABLES

30

MARCH 1985
REV 1.00

RANDEL/PEROUTKA

WSFO CLE/FTS 942-4949

PURPOSE

DECODES AIR TEMPERATURE FROM SYNOPTIC OBSERVATIONS. IT SEARCHES ARRAY IUP FOR 10000 GROUP, WHICH INDICATES TEMPERATURE IN TENTHS OF DEGREES, FROM 18GN TO ISTOP. IF SEARCH FAILS IT THEN LOOKS FOR A 1000/ GROUP, INDICATING TEMPERATURE IN WHOLE DEGREES. IT RETURNS TEMPERATURE, ITEMP, IN WHOLE DEGREES FARENHEIT.

ARGUMENT LIST

IUP() - INPUT ARRAY - WITH UNPACKED OBSERVATION.
18GN - INPUT INTEGER - POINTER TO ARRAY IUP WHERE TO BEGIN SEARCH FOR TEMP. SHOULD POINT TO SPACE BEFORE VSBY GROUP
ISTOP - INPUT INTEGER - POINTER TO ARRAY IUP WHERE TO STOP SEARCH FOR TEMP. SHOULD BE MARINE INDICATOR GROUP 222** IF AVAILABLE OR END OF OBSERVATION
ITEMP - OUTPUT INTEGER - TEMPERATURE IN WHOLE DEGEES FARENHEIT.

EXTERNALS

IUADEC - UTIL IFIX - FORT MASK

DIMENSION IUP(512)

ITEMP = -99

L = MASK('<0>101090909',5,IUP,18GN+8,ISTOP) ;TEMP IN TENTHS DEG.
IF (L .EQ. -1) GOTO 100 ;SEARCH FAILED

J = L + 1

TEMP = IUADEC(IUP,J)

;DECODE TEMP

IF(TEMP .GT. 1000.) TEMP = TEMP*(-1.) + 1000. ;TEMP IS NEGATIVE

TEMP = TEMP/10.

;CONVERT TO TENTHS

GOTO 200

L = MASK('<0>1010909<0>',5,IUP,18GN+8,ISTOP);TEMP IN WHOLE DEG.

IF (L .EQ. -1) GOTO 9999

;SEARCH FAILED

J = L + 1

TEMP = IUADEC(IUP,J)

;DECODE TEMP

IF(TEMP .GT. 100.) TEMP = TEMP*(-1.) + 100. ;TEMP IS NEGATIVE

TEMP = (TEMP * 9.) / 5. + 32.

;CONVERTING TO FARENHEIT

IF(TEMP .LT. 0.) TEMP = TEMP - .5

;ROUND OFF ERROR

IF(TEMP .GT. 0.) TEMP = TEMP + .5

;ROUND OFF ERROR

ITEMP = IFIX(TEMP)

;CONVERT TO INTEGER

RETURN

END

SUBROUTINE BACKUP(IUP, LAST, IPT, IER)
DIMENSION IUP(512), IBUF(128)

REV. 1.00

MAY, 1985

FORTRAN IV/ REV 5.20

PEROUTKA/RANDEL

DG ECLIPSE (S230)

WSFO CLE FTS 942-4949

RDOS/ REV. 6.18

MOVES A TWO-BLOCK WINDOW ONE BLOCK LOWER IN THE CURRENT AFOS DATABASE
PRODUCT AND UPDATES POINTERS.

ARGUMENT LIST

IUP I/O WINDOW WITH TWO BLOCKS OF DATABASE DATA.
LAST I/O LAST USABLE LOCATION IN IUP.
IPT I/O POINTER TO ANY LOCATION IN IUP.
IER 0 RETURN CODE. = 1 IF NEW DATA WAS LOADED.

EXTERNALS

WMOVE (UTIL) NXBKF (BG) UNPACK(UTIL)

CHANNELS/FILES

NONE.

VARIABLES

IBUF() TEMPORARY FOR PACKED DATABASE DATA.

CALL WMOVE(IUP(LAST-251), 252, IUP(1)); MOVE DATA HIGHER IN IUP.
IPT = 252 - LAST + IPT ; UPDATE POINTERS.
IF (IPT .LE. 0) IPT = 1
LAST = 252
CALL NXBKF(IBUF, IER) ; GET NEXT BLOCK IF IT'S THERE.
IF (IER.NE.1) RETURN
CALL UNPACK(IBUF(3), 252, IUP(LAST+1)); LOAD NEW BLOCK INTO WINDOW.
LAST = LAST + 252 ; UPDATE POINTER.
IF (IPT .EQ. -1) IPT = 1
RETURN
END

```

C ICHN=CHANNEL WHICH FILE HAS BEEN OPENED TO
C NREC=NUMBER OF RECORDS
C LREC=LENGTH OF EACH RECORD (BYTES)
C ISTAR=BYTE OF FIRST RECORD (0-BEGINNING)
C IFLD=WORD POINTER TO FIELD IN RECORD
C IFLD=LENGTH OF FIELD IN BYTES
C ITEST=ARRAY CONTAINING TEST FIELD
C IAD=RETURNED TWO WORD ARRAY CONTAINING ADDRESS ITEST RECORD
C SHOULD BEGIN AT-
C IC= 1,2,3 IN SECOND WORD INDICATING RECORD WAS FOUND AND
C IS IN ARRAY IC1,IC2, OR IC3
C THOSE THREE ARRAYS SHOULD BE DIMENSIONED LREC/2 WORDS
DIMENSION ITEST(1),IC1(1),IC2(1),IC3(1),IAD(2)
DIMENSION IAD1(2),IAD2(2),IAD3(2)
DIMENSION D1(2),D2(2)
INTEGER D1,D2
IC=0
IAD1(1)=0
IAD1(2)=ISTAR
CALL SPOS(ICHN,IAD1,IER)
CALL ERROR(IER,'I1')
CALL RDS(ICHN,IC1,LREC,IER)
CALL ERROR(IER,'RDS - IC1')
D2(1)=0
D2(2)=LREC
CALL DSUB(D2,D2,IAD1)
CALL DMPY(D1,NREC,LREC)
CALL DSUB(IAD2,D1,D2)
CALL SPOS(ICHN,IAD2,IER)
CALL ERROR(IER,'I2')
CALL RDS(ICHN,IC2,LREC,IER)
CALL ERROR(IER,'RDS-IC2')
CALL BCOMP(IC1(IFLDP),ITEST,IFLD,IER1)
IF(IER1.GT.1)GO TO 100
CALL BCOMP(IC2(IFLDP),ITEST,IFLD,IER2)
IF(IER2.NE.2)GO TO 125
5 CALL DSUB(D1,IAD2,IAD1)
CALL DDVD(INC,IR,D1,LREC)
IF(INC.GE.32767)GO TO 900
IF(INC.LT.1)GO TO 150
INC=(INC-1)/2+1
CALL DMPY(D1,INC,LREC)
CALL DADD(IAD3,IAD1,D1)
CALL SPOS(ICHN,IAD3,IER)
CALL ERROR(IER,'I5')
CALL RDS(ICHN,IC3,LREC,IER)
CALL ERROR(IER,'I6')
CALL BCOMP(IC3(IFLDP),ITEST,IFLD,IER3)
IF(IER3.EQ.1)GO TO 50
IF(IER3.EQ.2)GO TO 60
IF(IER3.NE.3)GO TO 900
IAD(1)=IAD3(1)
IAD(2)=IAD3(2)
IC=3
RETURN
50 IAD1(1)=IAD3(1)
IAD1(2)=IAD3(2)
GO TO 5
60 IAD2(1)=IAD3(1)
IAD2(2)=IAD3(2)
IF(INC.EQ.1)GO TO 150
GO TO 5
100 IAD(1)=IAD1(1)
IAD(2)=IAD1(2)
IF(IER1.NE.3)GO TO 101
IC=1
IAD(1)=IAD1(1)
IAD(2)=IAD1(2)
101 RETURN
125 D1(1)=0
D1(2)=LREC
CALL DADD(IAD,D1,IAD2)
IF(IER2.NE.3)GO TO 126
IAD(1)=IAD2(1)
IAD(2)=IAD2(2)
IC=2
126 RETURN
150 IAD(1)=IAD3(1)
IAD(2)=IAD3(2)
RETURN
900 CALL ERROR(IER3,'IER3')
IER=2
CALL ERROR(IER,'TOO MANY RECORDS IN FILE')
STOP
END

```


SUBROUTINE BOUYWV(IBGN,ISTOP,IUP,IWV)

JUNE 1985
REV 1.00

RANDEL/PEROUTKA

WSFO CLE/FTS 942-4949

PURPOSE

THIS SUBROUTINE DECODES WAVE HEIGHTS IN SYNOPTIC OBSERVATIONS.
IT SEARCHES ARRAY IUP FOR 1**** OR 2**** GROUP FROM IBGN TO ISTOP.
IBGN SHOULD BE POINTING TO THE 222** GROUP. THE SUBROUTINE RETURNS
THE WAVES, IWV, IN WHOLE FEET.

ARGUMENT LIST

IUP() - INPUT ARRAY - WITH UNPACKED OBSERVATION.
IBGN - INPUT INTEGER - POINTER TO ARRAY IUP WHERE
TO BEGIN SEARCH FOR WAVES. SHOULD BE MARINE
DATA GROUP 222**.
ISTOP - INPUT INTEGER - POINTER TO ARRAY IUP WHERE
TO STOP SEARCH FOR WAVES. END OF OBSERVATION.
IWV - OUTPUT INTEGER - WAVES IN WHOLE FEET.

EXTERNALS

IUADEC - UTIL IFIX - FORT MASK

DIMENSION IUP(512)

IWV = -99

SEARCH FOR A 1**** OR 2**** AFTER THE 222** GROUP WHICH
IBGN POINTS TO

IF(ISTOP.GT. IBGN+18) ISTOP = IBGN+18
L = MASK('12<057>9<057>90909',5,IUP,IBGN+6,ISTOP)
IF(L.LE. 0) GOTO 9999 ;SEARCHED FAILED

J = L + 3
WV = IUADEC(IUP,J)

WAVES REPORTED IN HALF METERS, CONVERTING TO FEET.
AND CORRECT FOR ROUND OFF ERRORS

WV = WV * 1.64 +.5
IWV = IFIX(WV)

;CONVERT TO WHOLE NUMBER

9999 RETURN
END

PURPOSE

RETRIEVES AND DECODES SYNOPTIC OBS FROM U.S. LAND STATIONS, BOUYS, CHANS AND CANADIAN COAST GUARDS FROM THE AFOS DATABASE IDENTIFIED BY 'KEY'. VARIOUS SUBROUTINES CALLED BY CANDEC DO THE ACTUAL DECODE. THE PROGRAM WILL DECODE ANY SYNOPTIC OBSERVATION WHERE THE OBSERVATIONS BEGINS WITH A STATION ID AND VISIBILITY IMMEDIATELY FOLLOWS THE ID (CHAN AND LAND SYNOPTIC) OR THE VISIBILITY IMMEDIATELY FOLLOWS THE TIME, LATITUDE AND LONGITUDE GROUPS (CANADIAN CG., AND BOUYS).

ARGUMENT LIST

KEY() - INPUT ARRAY - 9 LETTER AFOS IDENTIFIER
ISTN - INPUT ARRAY - STATION IDENTIFIER
IB - INPUT - NUMBER OF CHARACTERS IN STATION I.D..
IHR - INPUT INTEGER - HOUR OF RUN TIME
CTIM - INPUT REAL NO. -CUTOFF TIME FOR OLD PRODUCTS. IN MINUTES FROM THE BEGINNING OF THE YEAR
ITU - INPUT - DESCRIBES TYPE OF TIME GROUP TO SEARCH FOR IN OBSERVATION. SEE SUBROUTINE OBTIM.FR FOR DESCRIPTION
IWJ - INPUT INTEGER -WIND SPEED UNITS IN OBSERVATION. 1 FOR KNOTS, 2 FOR MPS
LAT - OUTPUT - LATITUDE IN TENTHS OF DEGREES
LON - OUTPUT - LONGITUDE IN TENTHS OF DEGREES
IV - OUTPUT - VISIBILITY IN SIXTEENTHS OF A MILE
ISKY - OUTPUT - SKY COVER IN EIGHTHS
IWIND - OUTPUT - WIND SPEED IN KNOTS
ITEMP - OUTPUT - TEMPERATURE IN DEGREES FARENHEIT
IPRES - OUTPUT - PRESSURE IN TENTHS OF MILLIBARS
ILX - OUTPUT - WEATHER, WMO CODE
IWT - OUTPUT - WATER TEMPERATURE IN DEGREES FARENHEIT
IWV - OUTPUT - WAVE HEIGHTS IN FEET

EXTERNALS

ATEMP BOUYLV LATLON MASK PKLND
PRES SIND WTEMP LK

VARIABLES

IUP() - ARRAY CONTAINING UNPACKED OBSERVATION
J1 - POINTS TO SPACE BEFORE VISIBILITY GROUP
J2 - POINTS TO SPACE BEFORE 222** GROUP, SEA CONDITIONS FOLLOW THIS GROUP
IPT - POINTS TO BEGINNING OF STATION NAME
IEQ - POINTS TO END OF OBSERVATION

DIMENSION IUP(512)

LAT = -99
LON = -99
IV = -99
ISKY = -99
IWIND = -99
ITEMP = -99
IPRES = -99
ILX = -99
IWT = -99
IWV = -99
IPK = -99

CALL GETP(KEY, ISTN, IB, IHR, CTIM, ITU, IUP, IPT, IEQ, IERR)

IF(IERR .NE. 1)RETURN

IPT, IEQ, J1, J2 ARE POINTERS USED IN SEARCHING FOR SPECIFIC GROUPS WITHIN AN OBSERVATION. IF ANY POINTER CAN NOT BE FOUND THE OBSERVATION WILL NOT BE DECODED

J1 = IPT + IB

;VSBY GRP JUST AFTER STN I.D.

IF(ITU .EQ. 1) J1 = IPT+IB+IB

;VSBY GRP FOLLOWS LONGITUDE GROUP

SEARCH FOR MARINE DATA GROUP 222**

J2 = MASK('012'<048)<0>2<0>2<0>2, 4, IUP, J1+12, IEQ); MARINE DATA GROUP

IF(J2 .EQ. -1) J2 = IEQ

;NO 222** GROUP

CALL LATLON(IPT, J1, IUP, LAT, LON)

CALL SWK(J1, J2, IUP, IV, ILX, ISKY)

CALL SWIND(J1, IWJ, IUP, IWIND)

CALL ATEMP(J1, J2, IUP, ITEMP)

CALL PRES(J1, J2, IUP, IPRES)

CALL PKLND(J1, IEQ, IWJ, IUP, IPK)

IF(J2 .EQ. IEQ) GOTO 9999

CALL WTEMP(J2, IEQ, IUP, IWT)

CALL BOUYLV(J2, IEQ, IUP, IWV)

9999 RETURN

END

SUBROUTINE CGRDEC(KEY,ISTN,CTIM,IBF,IBGN,IEND,IER,ISBY,IWIND,IWV,ITW,

+ IT,IPRES)

DIMENSION KEY(5), ISTN(2), IBF(512)

REV. 1.00

MAY, 1985

PEROUTKA/RANDEL
DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

DECODES A COAST GUARD OBSERVATION.

ARGUMENT LIST

KEY I NINE-BYTE KEY OF AFOS PRODUCT TO BE SEARCHED.
ISTN I THREE-BYTE IDENTIFIER OF STATION TO BE DECODED.
CTIM I (REAL) 'CUT TIME' OF REQUIRED OBSERVATION. SEE LAWEB.FR.
ISKY 0 SKY COVER IN OCTAS.
IWX 0 WEATHER IN ASCII FORM. TWO BYTES, SPACE-PADDED, OF ' ', 'T',
'Z', 'ZL', 'R', 'S', 'RS', 'L', 'F', 'D', 'BY', 'H', OR 'K'.
IVSBY 0 VISIBILITY IN SIXTEENTHS OF A MILE.
IWIND 0 WIND VECTOR. INTEGER BETWEEN 0 AND 3699.
IWV 0 WAVE HEIGHT IN FEET.
ITW 0 WATER TEMPERATURE IN DEGREES FAHRENHEIT.
IT 0 AIR TEMPERATURE IN DEGREES FAHRENHEIT.
IPRES 0 SEA LEVEL PRESSURE IN TENTHS OF MILLIBARS.

EXTERNALS

CGRSR	ICGSKY	ICGLW	ICGVS	MASK
ICGLN	ICGLV	ICGT	ICGPR	

CHANNELS/FILES

NONE.

VARIABLES

IBF() BLOCK WITH TWO UNPACKED BLOCKS FROM AFOS DATABASE.
IBGN LOCATION OF STATION IDENTIFIER IN IBF.
IEND LAST USABLE ADDRESS IN IBF.
I TEMPORARY POINTER USED TO KEEP PLACE IN IBF.

ISKY = -99 ;SET DEFAULTS.

IWX = ''

IVSBY = -99

IWIND = -99

IWV = -99

ITW = -99

IT = -99

IPRES = -99

CALL CGRSR(KEY,ISTN,3,CTIM,IBF,IBGN,IEND,IER) ;SET UP BUFFER.

IF (IER.EQ.-1) RETURN

ISKY = ICGSKY(IBF,IBGN+3) ;DECODE SKY, WEATHER, AND VSBY.

IWX = ICGWX(IBF,IBGN+4)

IVSBY = ICGVS(IBF,IBGN+4)

I = MASK('<12><15>',1,IBF,IBGN,IEND);FIND END OF LINE.

IF (I.GT.0) IEND = I

I = MASK('<0>/CW',2,IBF,IBGN+7,IEND) ;FIND WIND GROUP.

IF (I.LT.0) RETURN

IWIND = ICGWN(IBF,I) ;DECODE WIND.

IWV = ICGWV(IBF,I+6) ;WAVES.

ITW = ICGT(IBF,I+11) ;SEA TEMPERATURE.

IT = ICGT(IBF,I+15) ;AIR TEMPERATURE.

IPRES = ICGPR(IBF,I+19) ;PRESSURE.

RETURN

END

REV. 1.00

MAY, 1985
FORTRAN IV/ REV 5.20

PEROUTKA/RANDEL
DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

FINDS THE PROPER VERSION OF A PRODUCT AND SEARCHES FOR A GIVEN STATION IDENTIFIER.

ARGUMENT LIST

KEY I NINE-BYTE KEY OF AFOS PRODUCT TO BE SEARCHED.
ISTN I IDENTIFIER OF STATION TO BE DECODED.
IB I LENGTH OF IDENTIFIER IN BYTES.
CTIM I (REAL) "CUT TIME" OF REQUIRED OBSERVATION. SEE LAWEB.FR.
IUP O ARRAY FILLED WITH TWO BLOCKS OF DATA FROM AFOS PRODUCT.
IPT O LOCATION IN IUP WHERE IDENTIFIER BEGINS.
IEND O LAST USABLE LOCATION IN IUP.
IERR O RETURN CODE. = 1 FOR SUCCESS.

EXTERNALS

UNPACK(UTIL) GBLK0 NXBKF (BG) MASK BACKUP

CHANNELS/FILES

NONE.

VARIABLES

IBUF() TEMPORARY FOR DATABASE DATA.
MSK() UNPACKED STATION IDENTIFIER.
NVER NUMBER OF VERSIONS LEFT IN THE DATABASE TO SEARCH.
J LAST USABLE LOCATION IN IUP.

```
IERR = -1
NVER = -1                                ;GET CURRENT PRODUCT.
CALL UNPACK(ISTN, IB, MSK)                ;SET UP MASK FOR ID SEARCH.
CALL GBLK0(KEY, CTIM, CTIM+120., NVER, IBUF, IER)
IF (IER.NE.1) GOTO 900
NVER1 = NVER + 1
DO 500 K = 1, NVER1                        ;LOOP FOR EACH VERSION.
CALL UNPACK(IBUF, 256, IUP)
J = 256
CALL NXBKF(IBUF, IER)                      ;LOAD A SECOND BLOCK.
IF (IER.NE.1) GOTO 100
CALL UNPACK(IBUF(3), 252, IUP(J+1))
J = J + 252
100 IPT = MASK(MSK, IB, IUP, 1, J)          ;SEARCH FOR STATION ID.
IF (IPT.LT.0) GOTO 300                    ;FOUND THE ID.
IERR = 1
IF (IPT.GT.J-100) CALL BACKUP(IUP, J, IPT, IER)
IEND = MASK("<0><203>", 1, IUP, IPT, J)
IF (IEND.LT.0) IEND = J                  ;FIND END OF PRODUCT.
GOTO 900
300 CALL BACKUP(IUP, J, IPT, IER)          ;DIDN'T FIND STATION ID.
IF (IER.NE.1) GOTO 400
GOTO 100                                ;GO BACK AND KEEP SEARCHING.
400 CALL GBLK0(KEY, CTIM, CTIM+120., NVER, IBUF, IER)
IF (IER.NE.1) GOTO 900                  ;PREVIOUS VERSIONS.
500 CONTINUE
900 RETURN
END
```


REV. 1.00

MAY, 1985

FORTRAM IV/ REV 5.20

PEROUTKA/RANDEL
DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

CLOSES THE BROADCAST TEXT FILE AND STORES IT INTO THE AFOS DATABASE. THE NINE-CHARACTER FILE NAME BEGINS AT ICLIN(2). SENDS AN ALERT TO THE CONSOLE WHICH ORIGINATED THE RUN.

ARGUMENT LIST

ICLIN I PACKED ARRAY WITH THE NINE-LETTER FILE NAME
IBC I RDOS CHANNEL ON WHICH THE FILE IS OPENED

EXTERNALS

WRS (UTIL) KLOSE (UTIL) WMOVE (UTIL) IAND (UTIL) USTOR (TOP)
FORKP (BG)

CHANNELS/FILES

IBC RDOS CHANNEL ON WHICH THE FILE IS OPENED

VARIABLES

IFNAM NULL-TERMINATED NAME OF THE FILE STORE INTO THE DATABASE

N = 1

CALL WRS(IBC,"<203>",N,IER)

;END OF TEXT.

CALL KLOSE(IBC,IER)

;CLOSE FILE.

CALL WMOVE(ICLIN(2),5,IFNAM)

;STORE IN DATABASE.

IFNAM(5) = IAND(IFNAM(5),177400K)

CALL USTOR(IFNAM,0,IER)

CALL FORKP("LAWE",IFNAM,IER)

RETURN

END

DIMENSION ICLIN(41), IFNAM(5)

REV. 1.00

MAY, 1985
FORTRAN IV/ REV 5.20

PEROUTKA/RANDEL
DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

CLOSES THE PLOT FILE AND STORES IT INTO THE AFOS DATABASE. THE NINE-
CHARACTER FILE NAME BEGINS AT ICLIN(2). SENDS AN ALERT TO THE CONSOLE
WHICH ORIGINATED THE RUN.

ARGUMENT LIST

ICLIN I PACKED ARRAY WITH THE NINE-LETTER FILE NAME
IPC I RDOS CHANNEL ON WHICH THE FILE IS OPENED

EXTERNALS

WRS (UTIL) KLOSE (UTIL) WMOVE (UTIL) IAND (UTIL) USTOR (TOP)
FORKP (BG)

CHANNELS/FILES

IPC RDOS CHANNEL ON WHICH THE FILE IS OPENED

VARIABLES

IFNAM NULL-TERMINATED NAME OF THE FILE STORE INTO THE DATABASE

N = 14

CALL WRS(IPC, "END OF DATA<15><12><203>", N, IER)

CALL KLOSE(IPC, IER)

CALL WMOVE(ICLIN(2), 5, IFNAM)

IFNAM(5) = IAND(IFNAM(5), 177400K)

CALL USTORE(IFNAM, 0, IER)

CALL FORKP("LAWEB", IFNAM, IER)

RETURN

END

SUBROUTINE DSHP(ICLIN,IHR,CTIM)
 DIMENSION ICLIN(41)

REV. 1.00

MAY, 1985

FORTRAN IV/ REV 5.20

PEROUTKA/RANDEL
 DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
 RDOS/ REV. 6.18

CONTROLS THE DECODING OF SHIP REPORTS. DECODED REPORTS ARE LOADED INTO
 COMMON BLOCK SHP.

ARGUMENT LIST

ICLIN I COMMAND LINE CONTAINING
 9-BYTE AFOS KEYS STARTING AT ICLIN(2), ICLIN(7), AND
 ICLIN(32).

IHR I VALID TIME OF REQUIRED OBSERVATIONS.

CTIM I (REAL) "CUT TIME" OF REQUIRED OBSERVATIONS. SEE LAWEB.FR

EXTERNALS

ISHFT (FORT) GETPS

CHANNELS/FILES

NONE.

VARIABLES

I LOOP COUNTER AS KEYS ARE SEARCHED.

IC FIRST BYTE OF CURRENT KEY.

DO 100 I = 2, 32, 5

IC = ISHFT(ICLIN(I),-8)

IF (IC.LT."<0>A".OR.IC.GT."<0>Z")

CALL GETPS(ICLIN(I),IHR,CTIM,1)

;LOOP FOR EACH KEY.

;VALID KEYS START WITH A LETTER.

GOTO 900

;CALL SHIP DECODER.

100 CONTINUE

900 RETURN

END

SUBROUTINE GBLK0(KEY,CTIML,CTIMU,NVER,IBUF,IERR)
 DIMENSION KEY(5), KREC(20), IBUF(128)

REV. 1.00

MAY, 1985
 FORTRAN IV/ REV 5.20 •

PEROUTKA/RANDEL
 DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
 RDOS/ REV. 6.18

RETRIEVES THE FIRST BLOCK OF A SPECIFIC VERSION OF A PRODUCT FROM THE
 DATABASE.

ARGUMENT LIST

KEY I KEY OF DATABASE PRODUCT TO BE RETRIEVED.
 CTIML I OLDEST CREATION TIME WHICH IS ACCEPTABLE FOR THE PRODUCT.
 CTIMU I LATEST CREATION TIME WHICH IS ACCEPTABLE FOR THE PRODUCT.
 NVER I/O NUMBER OF VERSIONS REMAINING IN THE DATABASE. UPDATED WITH
 EACH CALL. MUST BE -1 FOR THE FIRST CALL.
 IBUF 0 256-BYTE PACKED BUFFER INTO WHICH THE DESIRED BLOCK IS LOADED.
 IERR 0 RETURN CODE. +1 FOR SUCCESS. -1 FOR FAILURE.

EXTERNALS

KSRCF (BG) IAND (FORT) PRVRF (BG) RDBKF (BG) ISHFT (FORT)

CHANNELS/FILES

NONE

VARIABLES

N TEMPORARY FOR NVER
 KREC() KEY RECORD FOR DATABASE PRODUCT
 IER USED FOR MISCELLANEOUS RETURN CODES
 I LOOP VARIABLE
 PTIM CREATION TIME OF CURRENT PRODUCT

IERR = -1

N = NVER

IF (NVER.GT.0) GOTO 100

CALL KSRCF(KEY,KREC,IER)

;GET KEY.

IF (IER.NE.1) GOTO 900

N = IAND(KREC(9),177K)

100 DO 200 I = 1, N

;LOOP THROUGH THE VERSIONS.

IF (NVER.GT.0) CALL PRVRF(IER)

NVER = N - I

CALL RDBKF(0,IBUF,IER)

;GET BLOCK ZERO.

IF (IER.NE.1) GOTO 900

PTIM = ISHFT(IBUF(9),-8)*16384. + IAND(IBUF(9),377K)*128. +

+ ISHFT(IBUF(10),-8)

;COMPUTE CREATION DATE-TIME.

IF (PTIM.LT.CTIML) GOTO 900

IF (PTIM.GT.CTIMU) GOTO 200

IERR = 1

;SUCCESS!

GOTO 900

200 CONTINUE

900 RETURN

END

SUBROUTINE GCOMF(ICC, IER)
DIMENSION IBF(20)

REV. 1.00

MAY, 1985
FORTRAN IV/ REV 5.20

PEROUTKA/RANDEL
DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

RETRIEVES A FILENAME FROM THE COMMAND LINE AND OPENS THIS FILE ON AN RDOS CHANNEL. MAKES AN ABNORMAL RETURN IF ANY PROBLEMS ARE ENCOUNTERED.

ARGUMENT LIST

ICC RDOS CHANNEL NUMBER ON WHICH NAMED FILE IS OPENED. RETURNED BY SUBROUTINE.
IER LINE NUMBER IN CALLING PROGRAM TO WHICH SUBROUTINE RETURNS IF AN ERROR OCCURS.

EXTERNALS

GCHN (UTIL) OPENR (UTIL) RDL (UTIL) RDS (UTIL) KLOSE (UTIL)
STBYT (TOP) SPCHR (UTIL)

CHANNELS/FILES

ALL FILES ARE OPENED ON RDOS CHANNELS.

IC THE COMMAND LINE FILE, COM.CM IS OPENED AND READ FROM THIS CHANNEL.

ICC THE FILE NAMED IN THE COMMAND LINE IS OPENED ON THIS CHANNEL.

VARIABLES

IBF THE NAME OF THE FILE FOR ICC IS READ INTO THIS BUFFER.

N GENERAL UTILITY BYTE COUNTER.

```
CALL GCHN(IC, IER)           ;OPEN COMMAND LINE FILE.
CALL OPENR(IC, 'COM.CM', 0, IER)
CALL RDL(IC, IBF, N, IER)     ;READ PROGRAM NAME.
IF (IER.NE.1) RETURN IER
N = 4                         ;READ GLOBAL SWITCHES.
CALL RDS(IC, IBF, N, IER)
CALL RDL(IC, IBF, N, IER)     ;READ FILENAME.
IF (N.EQ.0) RETURN IER
CALL KLOSE(IC, IER)          ;CLOSE COMMAND LINE FILE.
CALL STBYT(0, IBF, N+1)      ;ADD A NULL TO NAME.
CALL GCHN(ICC, IER)          ;OPEN FILE.
CALL OPENR(ICC, IBF, 0, IER)
IF (IER.EQ.1) RETURN
CALL SPCHR('CAN'T OPEN COMMAND FILE.<15>', IER)
RETURN IER
END
```

MARCH 1985
REV 1.00

RANDEL/PEROUTKA

WSFO CLE/FTS 942-4949

PURPOSE

CALLS UP THE CURRENT AFOS PRODUCT "KEY". IT FIRST CHECKS TO SEE IF THE PRODUCT IS CURRENT. IT THEN SEARCHES FOR A SPECIFIC STATION'S OBSERVATION WITHIN THE PRODUCT AND COMPARES THE OB TIME WITH THE RUN TIME. IF NECESSARY THE PROGRAM WILL CALL UP PREVIOUS VERSIONS TO FIND CURRENT OBSERVATION

ARGUMENT LIST

KEY() - INPUT ARRAY - 9 LETTER AFOS STATION IDENTIFIER
ISTN() - INPUT ARRAY - STATION IDENTIFIER
IB - INPUT INTEGER - NUMBER OF CHARACTERS IN STATION I.D..
IHR - INPUT INTEGER - HOUR OF RUN TIME
CTIM - INPUT REAL NO. -CUTOFF TIME FOR OLD PRODUCTS.
ITU - INPUT INTEGER - DESCRIBES TYPE OF TIME GROUP TO SEARCH FOR WITHIN OBSERVATION
IUP() - OUTPUT ARRAY - CONTAINING THE UNPACKED OBSERVATION
IPT - OUTPUT INTEGER - POINTS TO THE STATION I.D.
IEQ - OUTPUT INTEGER - POINTS TO THE END OF OBSERVATION
IERR - OUTPUT INTEGER - ERROR CODE. 1 IF SEARCH FOR OBSERVATION WAS COMPLETED. -1 IF TROUBLE FINDING OBSERVATION.

EXTERNALS

PACK	- UTIL	RDBKF	- BG	IUADEC	- FORT
UNPACK	- UTIL	NXBKF	- BG	BACKUP	-
WMOVE	- UTIL	KSRCF	- BG	MASK	
		PRVRF	- BG	OBTIM	

VARIABLES

KREC() - ARRAY CONTAINING KEY RECORD
IBUF() - ARRAY CONTAINING BLOCK OF DATA
IUP() - ARRAY CONTAINING UNPACKED OBSERVATION
MSK() - ARRAY WITH UNPACKED STATION I.D.
PTIM - DATE TIME POLONOMIAL

DIMENSION KREC(20), IBUF(128), MSK(10), ISTN(5), IUP(512)

IERR = -1

NVER = -1

CALL GBLK0(KEY,CTIM,CTIM+180.,NVER,IBUF,IER) ;RETRIEVE 1ST BLOCK

IF(IER.NE. 1) GOTO 9999

CALL UNPACK(ISTN,IB,MSK) ;UNPACK STATION ID

NVER1 = NVER + 1

DO 500 K = 1, NVER1

IPT = 1

CALL UNPACK(IBUF,256,IUP)

J = 256

CALL NXBKF(IBUF,IER) ;LOAD SECOND BLOCK

IF(IER.NE. 1) GOTO 100

CALL UNPACK(IBUF(3),252,IUP(J+1))

J = J + 252

100 IPT = MASK(MSK,IB,IUP,IPT,J) ;SEARCH FOR STATION ID.

IF(IPT.LT. 0) GOTO 300 ;FOUND ID.

IF (IPT.GT. J-100) CALL BACKUP(IUP,J,IPT,IER);OB. NEAR END OF BLOCK

IF(ITU.EQ. 3) ITIM = IHR

CALL OBTIM(IPT,IB,ITU,IUP,ITIM) ;GET TIME IN OB

IF(ITIM.EQ. -99) GOTO 400 ;ERROR IN OB TIME TRY PREV VER

IF(ITIM - IHR) 9999,150,400 ;CHECK OB TIME

150 IEQ = MASK('<0>='.1,IUP,IPT,J) ;FIND END OF OF

IF(IEQ.EQ. -1) IEQ = J

IF(IEQ.GT. IPT+100) IEQ= IPT + 100

IERR = 1

GOTO 9999

300 CALL BACKUP(IUP,J,IPT,IER) ;CALLS NEXT BLOCK

IF(IER.NE. 1) GOTO 400

GOTO 100

400 CALL GBLK0(KEY,CTIM,CTIM+180.,NVER,IBUF,IER)

IF(IER.NE. 1) GOTO 9999

500 CONTINUE

SUBROUTINE GETPS(KEY, IHR, CTIM, IWJ)
 MARCH 1985
 REV 1.00
 RANDER/PEROUTKA
 WSFO CLE/FTS 942-4949

PURPOSE

RETRIEVES AND DECODES SHIP REPORTS FROM THE AFOS DATA BASE IDENTIFIED BY "KEY". VARIOUS SUBROUTINES CALLED BY GETPS DO THE ACTUALL DECODING. THE DECODED OBS ARE STORED IN THE ARRAY ISHP. PREVIOUS VERSIONS OF "KEY" ARE CALLED UNTIL THE CUTOFF TIME IS REACHED. THE TIME WITHIN EACH SHIP REPORT IS ALSO CHECKED TO MAKE SURE THE REPORT IS CURRENT.

ARGUMENT LIST

KEY() - INPUT ARRAY - 9 LETTER AFOS IDENTIFIER
 IHR - INPUT INTEGER - HOUR OF RUN TIME
 CTIM - INPUT REAL NO. - CUTOFF TIME FOR OLD PRODUCTS.
 IN MINUTES FROM THE BEGINNING OF THE YEAR
 IWJ - INPUT INTEGER - WIND SPEED UNITS IN OBSERVATION
 1 FOR KNOTS, 2 FOR MPS.

EXTERNALS

PACK	- UTIL	NXBKF	- BG	BACKUP	BOUYWV	SWX
RDBKF	- UTIL	PRVRF	- BG	GBLK8	LATLON	WTEMP
UNPACK	- UTIL	IAND	- FORT	MASK	PRES	
KSCRF	- BG	IUADEC	- FORT	ATEMP	SWIND	

VARIABLES

KREC() - ARRAY CONTAINING KEY RECORD
 IBUF() - ARRAY CONTAININGG BLOCK OF DATA
 IUP() - ARRAY CONTAINING UNPACKED OBSERVATION
 ISHP() - ARRAY CONTAINING DECODED SHIP REPORTS
 NSHP - COUNTS THE NUMBER OF SHIP REPORTS DECODED
 PTIM - DATE TIME POLONOMIAL OF PRODUCT

DIMENSION KEY(5), KREC(20), IBUF(128), IUP(512)
 COMMON /SHP/ ISHP(50,11), NSHP
 DATA ISHP/550*-99/, NSHP/0/
 NVER = -1
 CTIML = CTIM - 30.
 CTIMU = CTIM + 180.
 CALL GBLK8(KEY, CTIML, CTIMU, NVER, IBUF, IER) ;LWR LMT TIME FOR CURRENT PRODUCT
 ;UPR LMT TIME FOR CURRENT PRODUCT
 IF (IER.NE.1) GOTO 9999 ;GET 1ST BLOCK OF DATA
 NVER1 = NVER + 1
 DO 1000 K=1, NVER1
 IEND = -1
 CALL UNPACK(IBUF, 256, IUP)
 IPT = 1
 J = 256
 CALL NXBKF(IBUF, IER)
 IF (IER.NE.1) GOTO 100
 CALL UNPACK(IBUF(3), 252, IUP(J+1))
 J = J + 252
 IEND = MASK('<0><203>', 1, IUP, IPT, J) ;AMOUNT OF DATA IN IUP
 ;FIND END OF PRODUCT

```

IF(IEND .NE. -1) J = IEND
IPT = MASK("<0>9<0>9090909",5,IUP,IPT+5,J);POINT TO LAT.
IF(IPT .LT. 0) GOTO 300
IF(IPT .GT. J-100 .AND. IEND .EQ. -1);POINTER NEAR END OF DATA
+ CALL BACKUP(IUP,J,IPT,IER) ;CALL IN NXT BLOCK
L = IPT - 4
ITIM = IUADEC(IUP,L)/10 ;GET SHIP REPORT TIME
IF(ITIM - IHR) 100,400,100 ;COMPARE TIME WITH RUN TIME
300 IF(IEND .NE. -1) GOTO 800
CALL BACKUP(IUP,J,IPT,IER) ;CALL IN NXT BLOCK
IF(IER .NE. 1)GOTO 800
GOTO 100
400 IEQ = MASK("<0>=",1,IUP,IPT,J) ;END OF OBSERVATION
IF(IEQ .EQ. -1) IEQ = J
IF(IEQ .GT. IPT + 100) IEQ = IPT + 100; LIMITS SHIP REPORT TO 100 CHAR.
IP1 = MASK("<012><040>09",2,IUP,IPT+11,IPT+12);POINTER TO VSBY GROUP
IF(IP1 .LE. IPT) GOTO 100
IP2 = MASK("<012><040><0>2<0>2<0>2",4,IUP,IP1+12,IEQ);SEA COND GRP.
IF(IP2 .EQ. -1) IP2 = IEQ ;NO SEA GROUP
ILK = -99
LAT = -99
LON = -99
IV = -99
ISKY = -99 ;INITIALIZE ALL DECODED DATA
IWIND = -99 ;TO -99 MISSING
IAT = -99
ILX = -99
IWT = -99
ILV = -99
CALL LATLON(IPT,IP1,IUP,LAT,LON) ;DECODE LATITUDE LONGITUDE
CALL SWX(IP1,IP2,IUP,IV,ILX,ISKY) ;DECODE VSBY, PRES WX., SKY COVER
CALL SWIND(IP1,ILV,IUP,IWIND) ;DECODE WIND
CALL ATEMP(IP1,IP2,IUP,IAT) ;DECODE AIR TEMP
CALL PRES(IP1,IP2,IUP,IPRES) ;DECODE PRESSURE
IF(IP2 .EQ. IEQ) GOTO 700
CALL WTEMP(IP2,IEQ,IUP,IWT)
CALL BOUYWV(IP2,IEQ,IUP,ILV) ;DECODE WAVES
700 NSHP = NSHP + 1
ISHP(NSHP,1) = LAKLOC(LAT,LON) ;PLACE DECODED DATA INTO
ISHP(NSHP,2) = LAT
ISHP(NSHP,3) = LON
ISHP(NSHP,4) = IV
ISHP(NSHP,5) = ISKY
ISHP(NSHP,6) = IWIND
ISHP(NSHP,7) = IAT
ISHP(NSHP,8) = IPRES
ISHP(NSHP,9) = ILX
ISHP(NSHP,10) = IWT
ISHP(NSHP,11) = ILV
GOTO 100
800 CALL GBLK0(KEY,CTIML,CTIMU,NVER,IBUF,IER); CALL UP PREV. VERSION
IF(IER .NE. 1) GOTO 9999
1000 CONTINUE
9999 RETURN
END

```



```

SUBROUTINE GPIX(ISTN,I,J,IERR)
COMMON /CHNBK/ ICHN, L, KEY(15)
DIMENSION ISTN(3), IB1(20), IB2(20), IB3(20)
DATA ICHN /-1/

```

REV. 1.00

MAY, 1985
 FORTRAN IV/ REV 5.20

PEROUTKA/RANDEL
 DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
 RDOS/ REV. 6.18

USES BNSCH TO RETRIEVE BACKGROUND TWO PIXEL COUNTS FOR A STATION FROM
 STDIR.MS.

ARGUMENT LIST

ISTN() I STATION IDENTIFIER. SIX BYTES. SPACE-PADDED.
 I 0 HORIZONTAL PIXEL COORDINATE.
 J 0 VERTICAL PIXEL COORDINATE.
 IERR 0 ERROR RETURN. EQUALS ONE FOR SUCCESS.

EXTERNALS

GCHN (UTIL) OPENR (UTIL) RDS (UTIL) BNSCH

CHANNELS/FILES

ICHN IS THE RDOS CHANNEL ON WHICH STDIR.MS IS OPENED.

VARIABLES

ICHN CHANNEL ON WHICH STDIR.MS FILE IS OPENED. -1 IF FILE HAS NOT
 BEEN OPENED YET.
 L LENGTH OF A STATION RECORD IN STDIR.MS.
 KEY() FIRST 30 BYTES (KEY RECORD) OF STDIR.MS.
 IB1() BUFFER FOR BNSCH TO USE WHILE SEARCHING.
 IB2() BUFFER FOR BNSCH TO USE WHILE SEARCHING.
 IB3() BUFFER FOR BNSCH TO USE WHILE SEARCHING.
 N BYTE COUNTER USED WITH RDS.

IERR = -1

IF (ICHN.NE.-1) GOTO 100 ;OPEN FILE IF IT'S NOT OPEN ALREADY.

CALL GCHN(ICHN,IER)

CALL OPENR(ICHN,"STDIR.MS",0;IER)

N = 30

CALL RDS(ICHN,KEY,N,IER)

L = (KEY(5) - KEY(4))*2

IF (IER.EQ.1) GOTO 100

ICHN = -1

GOTO 900

100 CALL BNSCH(ICHN,KEY(1),KEY(2),KEY(3),KEY(4),L,ISTN,N,IB1,IB2,

+ IB3,IER) ;SEARCH FOR ID.

IF (IER.EQ.0) GOTO 900

IERR = 1

;SUCCESS!

GOTO (200,300,400),IER

;GET PIXELS FROM BUFFER.

200 I = IB1(KEY(9))

J = IB1(KEY(10))

GOTO 900

300 I = IB2(KEY(9))

J = IB2(KEY(10))

GOTO 900

400 I = IB3(KEY(9))

J = IB3(KEY(10))

900 RETURN

END

SUBROUTINE GTIME(MONTH, IDAY, IYEAR, IHR, CTIM)
DIMENSION IDT(3), ITM(3)

REV. 1.00

MAY, 1985

FORTAN IV/ REV 5.20

PEROUTKA/RANDEL
DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

CHECKS THE SYSTEM CLOCK AND CHOOSES THE VALID DATE AND HOUR FOR THIS RUN.

ARGUMENT LIST

ALL ARGUMENTS ARE OUTPUT BY THE SUBROUTINE

MONTH MONTH OF VALID TIME

IDAY DAY OF VALID TIME

IYEAR YEAR OF VALID TIME

IHR VALID HOUR

CTIM JULIAN MINUTE OF VALID TIME - THIRTY MINUTES

EXTERNALS

DATE (FORT) TIME (FORT) THRU (TOP) JDATE

CHANNELS/FILES

NONE

VARIABLES

IDT

DATE ARRAY RETURNED BY THE FORTAN UTILITY DATE

ITM

TIME ARRAY RETURNED BY THE FORTAN UTILITY TIME

CALL DATE(IDT, IER)

CALL TIME(ITM, IER)

IHR = (ITM(1) + 1)/3*3

IF (IHR.LT.24) GOTO 100

IHR = IHR - 24

CALL THRU(IDT)

100 CTIM = (JDATE(IDT(1), IDT(2), IDT(3)) - 1)*1440. + IHR*60. - 30.

MONTH = IDT(1)

IDAY = IDT(2)

IYEAR = IDT(3)

RETURN

END

SUBROUTINE HDRB(IBC)
DIMENSION IAR(20)

REV. 1.00

MAY, 1985 PEROUTKA/RANDEL
FORTRAN IV/ REV 5.20 . DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

WRITES A MASS MEDIA HEADER ON RDOS CHANNEL IBC.

ARGUMENT LIST

IBC I RDOS CHANNEL TO WHICH HEADER WILL BE WRITTEN

EXTERNALS

MHDR (TOP) WRS (UTIL)

CHANNELS/FILES

IBC IS THE RDOS CHANNEL TO WHICH THE HEADER IS WRITTEN

VARIABLES

IAR() HOLD HEADER WHILE IT IS BEING CREATED

N BYTE COUNTER USED WITH WRS

CALL MHDR(1,IAR,IER) ;HEADER CAN BE 26 OR 27 BYTES LONG.

N = 26

CALL WRS(IBC,IAR(7),N,IER)

N = 1

IF (IAR(20).NE.0) CALL WRS(IBC,IAR(20),N,IER)

N = 2

CALL WRS(IBC,"<15><12>",N,IER)

RETURN

END

INTEGER FUNCTION IASCL(NUM, IPR)

REV. 1.00

MAY, 1985

FORTTRAN IV/ REV 5.20 .

MATTHEW R. PEROUTKA

DG ECLIPSE (S230)

WSFO CLE FTS 942-4949

RDOS/ REV. 6.18

FINDS THE DIGIT IN THE IPR'TH DECIMAL PLACE OF THE VALUE NUM. RETURNS THE ASCII EQUIVALENT OF THAT DIGIT IN THE LEFT BYTE WITH A NULL IN THE RIGHT BYTE.

ARGUMENT LIST

IASCL 0 ONE ASCII DIGIT IN THE LEFT BYTE

NUM I INTEGER VALUE WHERE DIGIT IS FOUND

IPR I A POWER OF TEN WHICH CHOOSES THE DIGIT OF NUM TO BE RETURNED

EXTERNALS

ISHFT (FORT)

CHANNELS/FILES

NONE

VARIABLES

NONE

IASCL = ISHFT(((NUM - (NUM/(10*IPR))*(10*IPR))/IPR + 48),8)
RETURN
END

INTEGER FUNCTION ICGPR(IBF,IPT)
DIMENSION IBF(512)

REV. 1.00

MAY, 1985

FORTRAN IV/ REV 5.20

PEROUTKA/RANDEL
DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

SEARCHES FOR A COAST GUARD PRESSURE GROUP AND DECODES IT.

ARGUMENT LIST

IBF I UNPACKED ARRAY TO BE SEARCHED.

IPT I LOCATION IN IBF WHERE SEARCH BEGINS.

ICGPR 0 PRESSURE (ALTIMETER SETTING) IN TENTHS OF MILLIBARS.

EXTERNALS

MASK IAUDEC(UTIL) IFIX (FORT) FLOAT (FORT)

CHANNELS/FILES

NONE.

VARIABLES

IEND LAST LOCATION IN IBF TO BE SEARCHED.

I LOCATION IN IBF WHERE PRESSURE GROUP BEGINS.

ICGPR = -99

;DEFAULT.

IEND = IPT + 5

I = MASK('09090909',4,IBF,IPT,IEND);FIND NUMBERS.

IF (I.LT.0) GOTO 900

ICGPR = IFIX(FLOAT(IAUDEC(IBF,I))*3.38638 + .5)

900 RETURN

END

INTEGER FUNCTION ICGSKY(IBF,IPT)
 DIMENSION IBF(512)

REV. 1.00

MAY, 1985

FORTRAM IV/ REV 5.20

PEROUTKA/RANDEL
 DG ECLIPSE (S230)

USFO CLE FTS 942-4949
 RDOS/ REV. 6.18

SEARCHES FOR COAST GUARD SKY COVER CONTRACTIONS AND DECODES THEM.

ARGUMENT LIST

IBF I UNPACKED ARRAY TO BE SEARCHED.
 IPT I LOCATION IN IBF WHERE SEARCH BEGINS.
 ICGSKY 0 SKY COVER IN OCTAS.

EXTERNALS

MASK

CHANNELS/FILES

NONE.

VARIABLES

IEND LAST LOCATION IN IBF TO BE SEARCHED.
 I TEMPORARY FOR MASK.

ICGSKY = -99

IEND = IPT + 10

I = MASK('<0> <0>C<0>Y',3,IBF,IPT,IEND)

IF (I.LT.0) GOTO 100 ;CLOUDY.

ICGSKY = 0

GOTO 900

100 I = MASK('<0> <0>P<0>C',3,IBF,IPT,IEND)

IF (I.LT.0) GOTO 200 ;PARTLY CLOUDY.

ICGSKY = 4

GOTO 900

200 I = MASK('<0> <0>C',2,IBF,IPT,IEND)

IF (I.LT.0) GOTO 900

ICGSKY = 0

;CLEAR.

900 RETURN

END


```

      INTEGER FUNCTION ICGT(IBF,IPT)
      DIMENSION IBF(512)

```

```

      REV. 1.00

```

```

      MAY, 1985

```

```

      FORTRAN IV/ REV 5.20

```

```

      PEROUTKA/RANDEL
      DG ECLIPSE (S230)

```

```

      WSFO CLE FTS 942-4949
      RDOS/ REV. 6.18

```

```

      SEARCHES FOR A COAST GUARD TEMPERATURE GROUP AND DECODES IT.

```

```

      ARGUMENT LIST

```

```

      IBF      I      UNPACKED ARRAY TO BE SEARCHED.

```

```

      IPT      I      LOCATION IN IBF WHERE SEARCH BEGINS.

```

```

      ICGT     O      TEMPERATURE (AIR OR SEA) IN DEGREES FAHRENHEIT.

```

```

      EXTERNALS

```

```

      MASK      IAUDEC(UTIL)

```

```

      CHANNELS/FILES

```

```

      NONE.

```

```

      VARIABLES

```

```

      IEND      LAST LOCATION IN IBF TO BE SEARCHED.

```

```

      I         LOCATION IN IBF WHERE TEMPERATURE GROUP BEGINS.

```

```

      ICGT = -99

```

```

      :DEFAULT.

```

```

      IEND = IPT + 4

```

```

      I = MASK('09',1,IBF,IPT,IEND)

```

```

      :FIND NUMBER.

```

```

      IF (I.LT.0) GOTO 100

```

```

      ICGT = IAUDEC(IBF,I)

```

```

      GOTO 900

```

```

100 I = MASK('<0>-',1,IBF,IPT,IEND)

```

```

      IF (I.LT.0) GOTO 900

```

```

      ICGT = IAUDEC(IBF,I)

```

```

900 RETURN

```

```

      END

```

INTEGER FUNCTION ICGVS(IBF,IPT)
DIMENSION IBF(512)

REV. 1.00

MAY, 1985
FORTRAN IV/ REV 5.20

PEROUTKA/RANDEL
DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

SEARCHES FOR COAST GUARD VISIBILITIES AND DECODES THEM.

ARGUMENT LIST

IBF I UNPACKED ARRAY TO BE SEARCHED.
IPT I LOCATION IN IBF WHERE SEARCH BEGINS.
ICGVS 0 VISIBILITY IN SIXTEENTHS OF A MILE.

EXTERNALS
MASK

CHANNELS/FILES
NONE.

VARIABLES

IEND LAST LOCATION IN IBF TO BE SEARCHED.
I TEMPORARY USED FOR MASK.

ICGVS = -99 ;DEFAULT.
IEND = IPT + 10
I = MASK('0909',2,IBF,IPT,IEND) ;WHOLE NUMBERS.
IF (I.LT.0) GOTO 100
ICGVS = (IBF(I) - 48)*160 + (IBF(I+1) - 48)*16
GOTO 900
100 I = MASK('09<0>/09',3,IBF,IPT,IEND);FRACTION.
IF (I.LT.0) GOTO 900
ICGVS = (IBF(I) - 48)*16/(IBF(I+2) - 48)
900 RETURN
END


```

      INTEGER FUNCTION ICGWN(IBF,IPT)
      DIMENSION IBF(512)

```

```

      REV. 1.00

```

```

      MAY. 1985

```

```

      PEROUTKA/RANDEL
      DG ECLIPSE (S230)

```

```

      WSFO CLE FTS 942-4949
      RDOS/ REV. 6.18

```

```

      SEARCHES FOR A COAST GUARD WIND GROUP AND DECODES IT.

```

```

      ARGUMENT LIST

```

```

      IBF      I   UNPACKED ARRAY TO BE SEARCHED.
      IPT      I   LOCATION IN IBF WHERE SEARCH BEGINS.
      ICGWN    0   WIND VECTOR.  INTEGER BETWEEN 0 AND 3699.

```

```

      EXTERNALS

```

```

      MASK

```

```

      CHANNELS/FILES
      NONE.

```

```

      VARIABLES

```

```

      IEND     LAST LOCATION IN IBF TO BE SEARCHED.
      J        NUMBER OF CHARACTERS IN WIND GROUP.
      N        VALUE WHICH A NORTH DIRECTION WILL TAKE (0 OR 360).
      K        LOOP VARIABLE.
      J2       J - 2.
      ID       DIRECTION.

```

```

      ICGWN = -99                                ;DEFAULT.

```

```

      IEND = IPT + 7

```

```

      IF (MASK('<0>C<0>A<0>L<0>M',4,IBF,IPT,IEND).LT.0) GOTO 100

```

```

      ICGWN = 0                                ;CALM.

```

```

      GOTO 900

```

```

100 J = 5                                       ;HOW MANY CHARACTERS WIND GROUP?

```

```

      I = MASK('EWEWEW0909',J,IBF,IPT,IEND)

```

```

      IF (I.GT.0) GOTO 200

```

```

      J = 4

```

```

      I = MASK('EWEW0909',J,IBF,IPT,IEND)

```

```

      IF (I.GT.0) GOTO 200

```

```

      J = 3

```

```

      I = MASK('EW0909',J,IBF,IPT,IEND)

```

```

      IF (I.LT.0) GOTO 900

```

```

200 N = 360

```

```

                                ;CONVERT LETTERS TO NUMBERS.

```

```

      I3 = I + J - 3

```

```

                                ;HOW MANY CHARACTERS IN DIRECTION + I

```

```

      IF((J.EQ. 5 .AND. IBF(I+2).EQ. '<0>E') .OR.

```

```

      +(J.EQ. 4 .AND. IBF(I+1).EQ. '<0>E')) N= 0

```

```

      DO 400 K = 1,I3

```

```

          IF (IBF(K).EQ. '<0>E') IBF(K) = 90

```

```

          IF (IBF(K).EQ. '<0>S') IBF(K) = 180

```

```

          IF (IBF(K).EQ. '<0>W') IBF(K) = 270

```

```

          IF (IBF(K).EQ. '<0>N') IBF(K) = N

```

```

400 CONTINUE

```

```

      J2 = J - 2

```

```

      GOTO (500,600,700), J2

```

```

                                ;COMPUTE DIRECTIONS.

```

```

500 ID = IBF(I)/10

```

```

      ICGWN = ID*100 + (IBF(I+1) - 48)*10 + IBF(I+2) - 48

```

```

      GOTO 900

```

```

600 ID = (IBF(I)/2 + IBF(I+1)/2 + 5)/10

```

```

      ICGWN = ID*100 + (IBF(I+2) - 48)*10 + IBF(I+3) - 48

```

```

      GOTO 900

```

```

700 ID = (IBF(I)/2 + IBF(I+1)/4 + IBF(I+2)/4 + 5)/10

```

```

      ICGWN = ID*100 + (IBF(I+3) - 48)*10 + IBF(I+4) - 48

```

```

900 RETURN

```

```

      END

```

INTEGER FUNCTION ICGWV(IBF,IPT)
 DIMENSION IBF(512)

REV. 1.00

MAY, 1985

FORTRAN IV/ REV 5.20

PEROUTKA/RANDEL
 DG ECLIPSE (S230)

USFO CLE FTS 942-4949
 RDOS/ REV. 6.18

SEARCHES FOR A COAST GUARD WAVE GROUP AND DECODES IT.

ARGUMENT LIST

IBF I UNPACKED ARRAY TO BE SEARCHED.
 IPT I LOCATION IN IBF WHERE SEARCH BEGINS.
 ICGWV 0 WAVE HEIGHT IN FEET.

EXTERNALS

MASK

CHANNELS/FILES

NONE.

VARIABLES

IEND LAST LOCATION IN IBF TO BE SEARCHED.

I TEMPORARY USED IN MASK.

ICGWV = -99

;DEFAULTS.

IEND = IPT + 5

IF (MASK('<0>C<0>A<0>L<0>M',4,IBF,IPT,IEND).LT.0) GOTO 100

ICGWV = 0

;CALM.

GOTO 900

100 I = MASK('0909',2,IBF,IPT,IEND)

IF (I.LT.0) GOTO 900

ICGWV = (IBF(I) - 48)*10 + IBF(I+1) - 48

900 RETURN

END


```

      INTEGER FUNCTION ICGWX(IBF,IPT)
      DIMENSION IBF(512)

C
C
C      REV. 1.00
C
C
C      MAY, 1985          *      PEROUTKA/RANDEL          WSFO CLE FTS 942-4949
C      FORTRAN IV/ REV 5.20      DG ECLIPSE (S238)          RDOS/ REV. 6.18
C
C      SEARCHES FOR COAST GUARD WEATHER CONTRACTIONS AND DECODES THEM.
C
C      ARGUMENT LIST
C      IBF      I      UNPACKED ARRAY TO BE SEARCHED.
C      IPT      I      LOCATION IN IBF WHERE SEARCH BEGINS.
C      ICGSKY  O      WEATHER CODE IN ASCII. TWO BYTES, SPACE PADDED, OF "", "T",
C                      "ZR", "ZL", "R", "S", "RS", "L", "F", "H", "BY", OR "K".
C
C      EXTERNALS
C      MASK
C
C      CHANNELS/FILES
C      NONE.
C
C      VARIABLES
C      IEND      LAST LOCATION IN IBF TO BE SEARCHED.
C
      ICGWX = ""                                ;DEFAULT.
      IEND = IPT + 7
      IF (MASK("<0>T<0>S<0>T<0>M",4,IBF,IPT,IEND).LT.0) GOTO 100
      ICGWX = "T "                                ;THUNDERSTORM.
      GOTO 900
100 IF (MASK("<0>Z<0>R",2,IBF,IPT,IEND).LT.0) GOTO 150
      ICGWX = "ZR"                                ;FREEZING RAIN.
      GOTO 900
150 IF (MASK("<0>Z<0>L",2,IBF,IPT,IEND).LT.0) GOTO 200
      ICGWX = "ZL"                                ;FREEZING DRIZZLE.
      GOTO 900
200 IF (MASK("<0>S<0>R",2,IBF,IPT,IEND).LT.0) GOTO 250
      ICGWX = "RS"                                ;RAIN AND SNOW.
      GOTO 900
250 IF (MASK("<0>R",1,IBF,IPT,IEND).LT.0) GOTO 300
      ICGWX = "R "                                ;RAIN.
      GOTO 900
300 IF (MASK("<0>S",1,IBF,IPT,IEND).LT.0) GOTO 350
      ICGWX = "S "                                ;SNOW.
      GOTO 900
350 IF (MASK("<0>L",1,IBF,IPT,IEND).LT.0) GOTO 400
      ICGWX = "L "                                ;DRIZZLE.
      GOTO 900
400 IF (MASK("<0>F",1,IBF,IPT,IEND).LT.0) GOTO 450
      ICGWX = "F "                                ;FOG.
      GOTO 900
450 IF (MASK("<0>H",1,IBF,IPT,IEND).LT.0) GOTO 500
      ICGWX = "H "                                ;HAZE.
      GOTO 900
500 IF (MASK("<0>K",1,IBF,IPT,IEND).LT.0) GOTO 900
      ICGWX = "K "
900 RETURN
      END

```

INTEGER FUNCTION IGCOD(I)

REV. 1.00

MAY, 1985

PEROUTKA/RANDEL
FORTRAN IV/ REV 5.20 . DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

EXAMINES THE PACKED ASCII CHARACTERS IN I AND RETURNS THEIR DECIMAL
VALUE IF THEY ARE DIGITS. MINUS ONE IS RETURNED IF THE CHARACTERS ARE
NOT DECIMAL DIGITS.

ARGUMENT LIST

IGCOD 0 DECIMAL VALUE
I I HOLDS TWO DECIMAL DIGITS

EXTERNALS

ISHFT (FORT) IAND (FORT)

CHANNELS/FILES

NONE

VARIABLES

J1 VALUE OF DIGIT IN LEFT BYTE
J2 VALUE OF DIGIT IN RIGHT BYTE

IGCOD = -1

J1 = ISHFT(I,-8) - 60K

IF (J1.LT.0.OR.J1.GT.9) RETURN

J2 = IAND(I,377K) - 60K

IF (J2.LT.0.OR.J2.GT.9) RETURN

IGCOD = J1*10 + J2

RETURN

END


```

      INTEGER FUNCTION IOPNB(ICLIN)
      DIMENSION ICLIN(41), IFNAM(5)

```

REV. 1.00

MAY, 1985

FORTRAM IV/ REV 5.20

PEROUTKA/RANDEL
DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

CREATES AND OPENS A FILE NAMED BY THE NINE BYTES BEGINNING WITH ICLIN(2).
HEADING DATA ARE WRITTEN TO THE FILE SO IT WILL STORE IN THE AFOS DATABASE
AS A BROADCAST TEXT WITH THE SAME NINE-CHARACTER ID. THREE BYTES OF
ADDRESSEE FOLLOW THE NAME IN ICLIN. THE RDOS CHANNEL ON WHICH THE FILE
WAS OPENED IS RETURNED. MINUS ONE IS RETURNED FOR FAILURE.

ARGUMENT LIST

IOPNB 0 RDOS CHANNEL OF FILE OPENED BY FUNCTION
ICLIN 1 COMMAND LINE CONTAINING FILE NAME AND ADDRESSEE

EXTERNALS

WMOVE (UTIL) IAND (FORT) DFILW (UTIL) CRAND (UTIL) GCHN (UTIL)
OPNE (UTIL) WRS (UTIL) SPCHR (UTIL)

CHANNELS/FILES

IOPNB RETURNS THE RDOS CHANNEL ON WHICH THE FILE WAS OPENED

VARIABLES

IFNAM NULL-TERMINATED NAME OF THE FILE TO OPEN

```

      CALL WMOVE(ICLIN(2),5,IFNAM)           ;GET FILENAME FROM COMMAND LINE.
      IFNAM(5) = IAND(IFNAM(5),177400K)      ;NULL PAD.
      CALL DFILW(IFNAM,IER)                  ;OPEN FILE.
      CALL CRAND(IFNAM,IER)
      CALL GCHN(IOPNB,IER)
      CALL OPNE(IOPNB,IFNAM,0,IER)
      N = 12
      CALL WRS(IOPNB,ICLIN(2),N,IER)          ;WRITE HEADERS TO FILE.
      IF (IER.EQ.1) GOTO 100                 ;ERROR MESSAGE.
      CALL SPCHR('CAN'T OPEN ',IER)
      CALL SPCHR(IFNAM,IER)
      CALL SPCHR('.<15>',IER)
      IOPNB = -1
      RETURN
100 N = 0
      CALL WRS(IOPNB,'<377><377><377><377>30<305><200>',N,IER)
      RETURN
      END

```

REV. 1.00

MAY, 1985

FORTRAN IV/ REV 5.20

PEROUTKA/RANDEL
DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

CREATES AND OPENS A FILE NAMED BY THE NINE BYTES BEGINNING WITH ICLIN(2).
HEADING DATA ARE WRITTEN TO THE FILE SO IT WILL STORE IN THE AFOS DATABASE
AS A PLOT FILE WITH THE SAME NINE-CHARACTER ID. THREE BYTES OF ADDRESSEE
FOLLOW THE NAME IN ICLIN. THE RDOS CHANNEL ON WHICH THE FILE WAS OPENED
IS RETURNED. MINUS ONE IS RETURNED FOR FAILURE.

ARGUMENT LIST

IOPNP 0 RDOS CHANNEL OF FILE OPENED BY FUNCTION
ICLIN 1 COMMAND LINE CONTAINING FILE NAME AND ADDRESSEE

EXTERNALS

WMOVE (UTIL) IAND (FORT) DFILW (UTIL) CRAND (UTIL) GCHN (UTIL)
OPNE (UTIL) WRS (UTIL) SPCHR (UTIL)

CHANNELS/FILES

IOPNP RETURNS THE RDOS CHANNEL ON WHICH THE FILE WAS OPENED

VARIABLES

IFNAM NULL-TERMINATED NAME OF THE FILE TO OPEN

```

CALL WMOVE(ICLIN(2),5,IFNAM)          ;GET FILENAME.
IFNAM(5) = IAND(IFNAM(5),177400K)
CALL DFILW(IFNAM,IER)                  ;OPEN FILE.
CALL CRAND(IFNAM,IER)
CALL GCHN(IOPNP,IER)
CALL OPNE(IOPNP,IFNAM,0,IER)
N = 12                                ;WRITE HEADINGS.
CALL WRS(IOPNP,ICLIN(2),N,IER)
IF (IER.EQ.1) GOTO 100                 ;ERROR HANDLING.
CALL SPCHR('CAN'T OPEN ',IER)
CALL SPCHR(IFNAM,IER)
CALL SPCHR('<15>',IER)
100 N = 8                              ;MORE HEADINGS.
CALL WRS(IOPNP,'<377><377><377><377>50<305><200>',N,IER)
N = 9
CALL WRS(IOPNP,IFNAM,N,IER)
N = 46
CALL WRS(IOPNP,'0010200020481536205014250097501600150702051519',
+      N,IER)
N = 2
CALL WRS(IOPNP,'<15><12>',N,IER)
RETURN
END

```



```

      INTEGER FUNCTION IWX2N(IWX)
      COMMON /WBLK/ IWC(12), IWTEXT(6,13), NWC(12)
      DATA NWC /95, 66, 56, 61, 71, 68, 51, 18, 6, 7, 5, 4/

```

REV. 1.88

MAY, 1985
FORTRAN IV/ REV 5.28

PEROUTKA/RANDEL
DG ECLIPSE (S238)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

CONVERTS TWO-LETTER WEATHER CODES TO WMO WEATHER CODE NUMBERS.

ARGUMENT LIST

IWX I WEATHER IN ASCII FORM. TWO BYTES, SPACE-PADDED OF " ", "T",
 "ZR", "ZL", "R", "S", "RS", "L", "F", "D", "BY", "H", OR "K".
IWX2N O WMO CODE NUMBER. (8 - 99). -99 FOR FAILURE.

EXTERNALS

IAND (FORT)

CHANNELS/FILES
NONE.

VARIABLES

IW HOLDS TWO-LETTER WEATHER CODE.
IWC() LIST OF WEATHER CODES.
NWC() LIST OF WMO WEATHER CODES.
I LOOP COUNTER DURING SEARCH THROUGH LIST.
J TEMPORARY TO HOLD TWO-LETTER WEATHER CODE.

```

      IW = IWX
      IF (IAND(IW,377K).EQ.0) IW = IOR(IW,"<8> ")
      DO 100 I = 1, 12
         J = I
         IF (IW.EQ.IWC(I)) GOTO 200
100  CONTINUE
         IWX2N = -99
         RETURN
200  IWX2N = NWC(J)
      RETURN
      END

```

;USE A SPACE PAD.
;SEARCH FOR CODE.
;SEARCH FAILED.

```

INTEGER FUNCTION JDATE(MONTH, IDAY, IYEAR)
COMMON /DJULQ/ MLIST(12)
DATA MLIST/31,28,31,30,31,30,31,31,30,31,30,31/

```

REV. 1.00

MAY, 1985
FORTRAN IV/ REV 5.20

MATTHEW R. PEROUTKA
DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

RETURNS THE NUMBER OF DAYS SINCE JANUARY 1 FOR DATE INPUT.

ARGUMENT LIST

JDATE	O	JULIAN DATE
MONTH	I	MONTH
IDAY	I	DAY
IYEAR	I	YEAR

EXTERNALS
NONE

CHANNELS/FILES
NONE

VARIABLES

MLIST LIST WITH NUMBER OF DAYS IN EACH MONTH.

```

JDATE = 0 ;CHECK INPUT DATA.
IF (MONTH.LT.1.OR.MONTH.GT.12) RETURN
IF (IYEAR.LT.1) RETURN
IF (IDAY.LT.1) RETURN
MLIST(2) = 28
I = IYEAR/4*4 ;IS THIS A LEAP YEAR?
IF (I.EQ.IYEAR) MLIST(2) = 29
I = IYEAR/100*100
IF (I.EQ.IYEAR) MLIST(2) = 28
I = IYEAR/400*400
IF (I.EQ.IYEAR) MLIST(2) = 29
IF (IDAY.GT.MLIST(MONTH)) RETURN
DO 100 I = 1,MONTH ;COMPUTE JULIAN DATE.
100 JDATE = JDATE + MLIST(I)
JDATE = JDATE - MLIST(MONTH) + IDAY
RETURN
END

```


INTEGER FUNCTION LAKLOC(LAT,LON)

REV. 1.00

MAY, 1985

FORTRAN IV/ REV 5.20

PEROUTKA/RANDEL

DG ECLIPSE (S238)

WSFO CLE FTS 942-4949

RDOS/ REV. 6.18

IDENTIFIES THE LAKE WHERE A POINT IS LOCATED.

ARGUMENT LIST

LAT I LATITUDE IN TENTHS OF DEGREES NORTH.

LON I LONGITUDE IN TENTHS OF DEGREES WEST.

LAKLOC 0 LAKE NUMBER. 1 = LS, 2 = LM, 3 = LH, 4 = LE, 5 = LO.

EXTERNALS

NONE.

CHANNELS/FILES

NONE.

VARIABLES

NONE.

LAKLOC = -1

IF (LAT.LT.464.OR.LAT.GT.490.OR. ;SUPERIOR

+ LON.LT.840.OR.LON.GT.921) GOTO 100

LAKLOC = 1

GOTO 900

100 IF (LAT.LT.413.OR.LAT.GT.463.OR. ;MICHIGAN

+ LON.LT.847.OR.LON.GT.885) GOTO 200

LAKLOC = 2

GOTO 900

200 IF (LAT.LT.430.OR.LAT.GT.463.OR. ;HURON

+ LON.LT.799.OR.LON.GT.846) GOTO 300

LAKLOC = 3

GOTO 900

300 IF (LAT.LT.413.OR.LAT.GT.429.OR. ;ERIE

+ LON.LT.788.OR.LON.GT.835) GOTO 400

LAKLOC = 4

GOTO 900

400 IF (LAT.LT.430.OR.LAT.GT.444.OR. ;ONTARIO

+ LON.LT.760.OR.LON.GT.798) GOTO 900

LAKLOC = 5

900 RETURN

END

SUBROUTINE LATLON(IBGN,ISTOP,IUP,LAT,LON)

MARCH 1985
REV 1.00

RANDEL/PEROUTKA

WSFO CLE/FTS 942-4949

PURPOSE

DECODES LATITUDE AND LONGITUDE FROM SYNOPTIC OBSERVATIONS. IT SEARCHES ARRAY IUP FOR A 99*** GROUP FOR THE LATITUDE. IT THEN SEARCHES FOR A 7 GROUP IMMEDIATELY AFTER THE 99 GROUP FOR THE LONGITUDE. IT SEARCHES THE ARRAY FROM IBGN TO ISTOP. IT EITHER THE LAT. OR LON. CAN NOT BE FOUND A -99 IS RETURNED FOR BOTH

ARGUMENT LIST

IUP() - INPUT ARRAY - WITH UNPACKED OBSERVATION.
IBGN - INPUT INTEGER - POINTER TO ARRAY IUP WHERE
TO BEGIN SEARCH
ISTOP - INPUT INTEGER - POINTER TO ARRAY IUP WHERE
TO STOP SEARCH
LAT - OUTPUT INTEGER - LATITUDE IN TENTHS OF DEGREES.
LON - OUTPUT INTEGER - LONGITUDE IN TENTHS OF DEGREES.

EXTERNALS

IUADEC - UTIL IFIX - FORT MASK

DIMENSION IUP(512)

LAT = -99

LON = -99

SEARCHING FOR LATITUDE

L = MASK('<0>9<0>9090909',5,IUP,IBGN,ISTOP);SEARCH FOR 99*** GRP
IF(L.EQ. -1) GOTO 9999 ;FOUND LAT GRP

J = L + 2

LAT = IUADEC(IUP,J)

;DECODE LAT

IF(LAT.LT. 0 .OR. LAT.GT. 900) GOTO 1000 ;LAT OUT OF BOUNDS

SEARCHING FOR LONGITUDE, 7*** GROUP RIGHT AFTER LAT GROUP

L = MASK('<0>709090909',5,IUP,L,L+12) ;SEARCH FOR 7 GROU
IF(L.EQ. -1) GOTO 1000 ;FOUND LON GRP

J = L + 2

LON = IUADEC(IUP,J)

;DECODE LON

IF(LON.GE. 0 .OR. LON.LE. 18000) GOTO 9999;LON WITHIN CORRECT VALUES
;PROBLEM FINDING LAT

LAT = -99

LON = -99

RETURN

END


```

FUNCTION MASK(MSK,LMSK,IUP,IBGN,ISTOP)
DIMENSION MSK(1), IUP(1)

```

THIS FUNCTION SEARCHES IUP (AN UNPACKED ARRAY OF ASCII CHARACTERS) FROM IBGN TO ISTOP FOR A STRING WHICH MATCHES THE MASK MSK. MSK IS LMSK WORDS IN LENGTH AND HAS THE FOLLOWING CHARACTERISTICS:

1. IF THE LEFT BYTE OF A WORD IS 0, THE RIGHT BYTE NEEDS AN EXACT MATCH.
2. IF THE LEFT BYTE IS NON-ZERO, THEN A MATCH MUST BE GREATER THAN OR EQUAL TO THE LEFT BYTE; AND LESS THEN OR EQUAL TO THE RIGHT BYTE. '09' OR 'AZ', E. G.
3. IF A WORD EQUALS -1, IT WILL MATCH ANY BYTE (A WILD CARD).

THE FUNCTION RETURNS THE LOCATION IN IUP OF THE BEGINNING OF THE SUBSTRING. MASK RETURNS -1 IF THE SEARCH FAILS.

```

LIMIT = ISTOP - LMSK + 1
DO 600 MASK = IBGN, LIMIT
DO 500 I = 1, LMSK
  M = MSK(I)
  IF (M.EQ.-1) GOTO 500           ;WILD CARD.
  ML = ISHFT(M,-8)
  MR = IAND(M,377K)
  L = IUP(MASK+I-1)
  IF (ML.NE.0) GOTO 200           ;NEED AN EXACT MATCH.
  IF (MR.EQ.L) GOTO 500
  GOTO 600
200  IF (L.LT.ML.OR.L.GT.MR) GOTO 600 ;RANGE FOR MATCH.
500  CONTINUE
     GOTO 700
600  CONTINUE
     MASK = -1
700  RETURN
     END

```

;UNSUCCESSFUL SEARCH.

INTEGER FUNCTION N2ILX(N)

REV. 1.00

MAY, 1985

FORTRAN IV/ REV 5.20

PEROUTKA/RANDEL

DG ECLIPSE (S230)

WSFO CLE FTS 942-4949

RDOS/ REV. 6.18

CONVERTS WMO WEATHER CODE TO ASCII FORM.

ARGUMENT LIST

N I WMO CODE NUMBER (0 - 99).

N2ILX 0 WEATHER IN ASCII FORM. TWO BYTES, SPACE-PADDED OF " ", "T",
"ZR", "ZL", "R", "S", "RS", "L", "F", "D", "BY", "H", OR "K".

EXTERNALS

NONE.

CHANNELS/FILES

NONE.

VARIABLES

NONE.

N2ILX = "T "

IF (N.GT.90.OR.N.EQ.29) GOTO 900

IF (N.GT.16.AND.N.LT.20) GOTO 900

N2ILX = "ZR"

IF (N.EQ.24.OR.N.EQ.66.OR.N.EQ.67)

N2ILX = "ZL"

IF (N.EQ.56.OR.N.EQ.57) GOTO 900

N2ILX = "R "

IF (N.EQ.21.OR.N.EQ.25.OR.N.EQ.27)

IF (N.GT.57.AND.N.LT.66) GOTO 900

IF (N.GT.79.AND.N.LT.83) GOTO 900

N2ILX = "S "

IF (N.EQ.22.OR.N.EQ.26) GOTO 900

IF (N.GT.69.AND.N.LT.80) GOTO 900

IF (N.GT.84.AND.N.LT.91) GOTO 900

N2ILX = "RS"

IF (N.EQ.23.OR.N.EQ.68.OR.N.EQ.69)

IF (N.EQ.83.OR.N.EQ.84) GOTO 900

N2ILX = "L "

IF (N.EQ.20) GOTO 900

IF (N.GT.49.AND.N.LT.56) GOTO 900

N2ILX = "F "

IF (N.EQ.28) GOTO 900

IF (N.GT.9.AND.N.LT.13) GOTO 900

IF (N.GT.39.AND.N.LT.50) GOTO 900

N2ILX = "D "

IF (N.EQ.6.OR.N.EQ.8.OR.N.EQ.9) GOTO 900

IF (N.GT.29.AND.N.LT.36) GOTO 900

N2ILX = "Y "

IF (N.EQ.7) GOTO 900

N2ILX = "H "

IF (N.EQ.5) GOTO 900

N2ILX = "K "

IF (N.EQ.4) GOTO 900

N2ILX = ""

900 RETURN

END

;THUNDERSTORM CODES 17-19, 29,
;AND 91-99.

;FREEZING RAIN CODES 24, 66, AND
GOTO 900;67
;FREEZING DRIZZLE CODES 56 AND 57.

;RAIN CODES 21, 25, 27, 58-65, AND
GOTO 900;80-82.

;SNOW CODES 22, 26, 70-79, AND
;85-90.

;RAIN/SNOW CODES 23, 68, 69, 83,
GOTO 900;AND 84.

;DRIZZLE CODES 20 AND 50-55.

;FOG CODES 10-12, 20, AND 40-49.

;DUST CODES 6, 8, 9, AND 30-35.

;BLOWING SPRAY CODE 7.

;HAZE CODE 5.

;SMOKE CODE 4.

;ALL OTHERS RETURN NULL.

SUBROUTINE OBTIM(IPT,IB,ITU,IUP,ITIM)

MARCH 1985
REV 1.00

RANDEL/PEROUTKA

WSFO CLE/FTS 942-4949

PURPOSE

DECODES THE TIME GROUP WITHIN DIFFERENT TYPE SYNOPTIC OBSERVATIONS, I.E. LAND VERSUS SHIP. THE VARIABLE ITU DESCRIBES THE TYPE OF TIME GROUP THAT THE PROGRAM SEARCHES FOR. IUT = 1 INDICATES THE TIME GROUP IS JUST AFTER THE STATION ID. THE STATION ID IS "IB" BYTES IN LENGTH. ITU = 2 INDICATES THAT THE TIME GROUP IS IN A 9**** GROUP FOLLOWING A 555 GROUP. ITU = 3 INDICATES NO TIME GROUP WITHIN THE OBSERVATION, PRODUCT CONSIDERED CURRENT BASED ON THE DATE TIME POLYNOMIAL. THE SUBROUTINE RETURNS THE DECODED TIME, ITIM, TO THE CALLING PROGRAM IN HOURS.

ARGUMENT LIST

IUP() - INPUT ARRAY - WITH UNPACKED OBSERVATION.
IPT - INPUT INTEGER - POINTER TO AN ELEMENT WITHIN THE ARRAY IUP
IB - INPUT INTEGER - NUMBER OF BYTES IN THE STATION IDENTIFIER.
ITU - INPUT INTEGER - DESCRIBES TYPE OF TIME GROUP TO SEARCH FOR. 1 = TIME GROUP RIGHT AFTER STATION IDENTIFIER. ITU = 2, TIME IN 9**** GROUP FOLLOWING THE 555 GROUP. 3 = NO TIME GROUP IN OBSERVATION, DO NOT ATTEMPT TO DECODE TIME.

ARGUMENTS

IUADEC - UTIL IFIX - FORT MASK

DIMENSION IUP(512)

```

C      IF (ITU .EQ. 3) GOTO 9999          ;DO NOT DECODE TIME GROUP.
C      ITIM = -99
C      GOTO(100,200), ITU                ;DETERMINE VALUE OF ITU
100    J = IPT + IB + 3                  ;ITU = 1, JUMP TO TIME GROUP
C      I = IUADEC(IUP,J)                  ;DECODE TIME GROUP
C      ITIM = I/10                        ;CALCULATE HOUR
C      GOTO 300
200    J = MASK("<012><040><0>5<0>5<0>5<012><040><0>909090909",10,IUP,
+IPT+25,IPT+100)                        ;ITU = 2, SEARCH FOR " 555 9****"
C      IF (J .EQ. -1) GOTO 300            ;SEARCH FAILED
C      J = J + 6
C      ITIM = IUADEC(IUP,J)
C      ITIM = ITIM/100
300    IF (ITIM .LT. 0 .OR. ITIM .GT. 23) ITIM = -99
9999   RETURN
      END
```

```
SUBROUTINE OMRDEC(KEY,ISTN,CTIM,IWND,IWV)
  DIMENSION KEY(5), ISTN(5), IBF(512)
```

```
REV. 1.00
```

```
MAY, 1985
```

```
FORTRAN IV/ REV 5.20
```

```
PEROUTKA/RANDEL
DG ECLIPSE (S238)
```

```
WSFO CLE FTS 942-4949
RDOS/ REV. 6.18
```

```
DECODES AN OMR OBSERVATION.
```

``` ARGUMENT LIST ```

```
KEY      I  NINE-BYTE KEY OF AFOS PRODUCT TO BE SEARCHED.
ISTN     I  THREE-BYTE IDENTIFIER OF STATION TO BE DECODED.
CTIM     I  (REAL) "CUT TIME" OF REQUIRED OBSERVATION.  SEE LAWEB.FR.
IWND     0  WIND VECTOR.  INTEGER BETWEEN 0 AND 3699.
IWV      0  WAVE HEIGHT IN FEET.
```

``` EXTERNALS ```

```
IAND (FORT)  CGRSR          MASK          IUADEC(UTIL)  ICGLN
ICGLV
```

``` CHANNELS/FILES ```

```
NONE.
```

``` VARIABLES ```

```
IBF( )  BLOCK WITH TWO UNPACKED BLOCKS FROM AFOS DATABASE.
IB       LENGTH OF STATION ID IN BYTES.
IBGN     LOCATION OF STATION IDENTIFIER IN IBF.
IEND     LAST USABLE ADDRESS IN IBF.
IPT      POINTER TO CURRENT PLACE IN IBF.
I        TEMPORARY USED FOR MASK SEARCHES.
```

```
IWND = -99
```

```
;SET DEFAULTS.
```

```
IWV = -99
```

```
IB = 3
```

```
;HOW LONG IS ID?
```

```
IF (IAND(ISTN(2),377K).GT."<0>0") IB = 5
```

```
CALL CGRSR(KEY,ISTN,IB,CTIM,IBF,IBGN,IEND,IER) ;SET UP BUFFER.
```

```
IF (IER.EQ.-1) GOTO 900
```

```
I = MASK("<12><15>",1,IBF,IBGN,IEND);FIND END OF LINE.
```

```
IF (I.GT.0) IEND = I
```

```
IPT = IBGN + 6
```

```
;MOVE POINTER PAST ID.
```

```
I = MASK("09090909",4,IBF,IPT,IPT+10);LOOK FOR FOUR-DIGIT WIND GROUP.
```

```
IF (I.LT.0) GOTO 100
```

```
IWND = IUADEC(IBF,IPT)
```

```
GOTO 200
```

```
100 IWND = ICGLN(IBF,IPT)
```

```
;LOOK FOR CGR-TYPE WIND GROUP.
```

```
IF (IWND.NE.-99) GOTO 200
```

```
I = MASK("EW09",2,IBF,IPT,IEND)
```

```
;LOOK FOR CGR-TYPE WITH ONLY ONE
```

```
IF (I.LT.0) GOTO 200
```

```
;FORCE DIGIT.
```

```
IBF(I+2) = IBF(I+1)
```

```
;ADD A ZERO AND TRY AGAIN.
```

```
IBF(I+1) = "<0>0"
```

```
IWND = ICGLN(IBF,IPT)
```

```
200 I = MASK("<0>W<0>A<0>V<0>E",4,IBF,IPT,IEND);WAVES.
```

```
IF (I.LT.0) GOTO 900
```

```
IPT = I
```

```
I = MASK("<0> ",1,IBF,IPT,IEND)
```

```
IF (I.LT.0) GOTO 900
```

```
IWV = ICGLV(IBF,I)
```

```
;WAVES.
```

```
900 RETURN
```

```
END
```


SUBROUTINE PBOY(IBC, IPC, ICLIN, IHR, CTIM)
 DIMENSION ICLIN(41)

REV. 1.00

MAY, 1985

FORTRAN IV/ REV 5.20

PEROUTKA/RANDEL
 • DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
 RDOS/ REV. 6.18

CONTROLS DECODING AND PRINTING OF A BUOY OBSERVATION.

ARGUMENT LIST

IBC I RDOS CHANNEL TO WHICH BROADCAST TEXT WILL BE WRITTEN.
 IPC I RDOS CHANNEL TO WHICH PLOT DATA WILL BE WRITTEN.
 ICLIN() I COMMAND LINE CONTAINING
 5-BYTE STATION IDENTIFIER STARTING AT ICLIN(2),
 9-BYTE AFOS KEY STARTING AT ICLIN(5), ICLIN(10), AND
 ICLIN(35).
 IHR I VALID TIME OF REQUIRED OBSERVATION.
 CTIM I (REAL) "CUT TIME" OF REQUIRED OBSERVATION. SEE LAWE.B.FR.

EXTERNALS

ISHFT (FORT) CANDEC LAKLOC PRB CART (AG)
 FLOAT (FORT) PRP

CHANNELS/FILES

IBC IS THE RDOS CHANNEL TO WHICH BROADCAST TEXT IS WRITTEN.
 IPC IS THE RDOS CHANNEL TO WHICH PLOT DATA IS WRITTEN.

VARIABLES

K LOOP COUNTER WHILE SEARCHING FOR KEYS.
 IC THE FIRST BYTE OF A POTENTIAL KEY NAME.
 LAT LATITUDE IN TENTHS OF A DEGREE NORTH.
 LON LONGITUDE IN TENTHS OF A DEGREE WEST.
 IV DUMMY FOR CANDEC.
 ISKY DUMMY FOR CANDEC.
 ILND WIND VECTOR. INTEGER BETWEEN 0 AND 3699.
 IT AIR TEMPERATURE IN DEGREES FAHRENHEIT.
 IPRES SEA LEVEL PRESSURE IN TENTHS OF MILIBARS.
 ILX DUMMY FOR CANDEC.
 IWT WATER TEMPERATURE IN DEGREES FAHRENHEIT.
 ILV WAVE HEIGHT IN FEET.
 IPK PEAK WIND SPEED IN KNOTS.
 LAKE LAKE NUMBER. 1 = LS, 2 = LM, 3 = LH, 4 = LE, 5 = LO.
 I HORIZONTAL PIXEL COORDINATE.
 J VERTICAL PIXEL COORDINATE.

DO 100 K = 5, 35, 5

;GET PRODUCT KEYS.

IC = ISHFT(ICLIN(K), -8)

IF (IC.LT. "<0>A".OR.IC.GT. "<0>Z") GOTO 100

CALL CANDEC(ICLIN(K), ICLIN(2), 5, IHR, CTIM, 1, 2, LAT, LON, IV, ISKY,
 + ILND, IT, IPRES, ILX, IWT, ILV, IPK)

IF (LAT.EQ. -99) GOTO 100

;KEEP LOOKING.

LAKE = LAKLOC(LAT, LON)

;GET LAKE NUMBER.

IF (IBC.NE. -1) CALL PRB(IBC, "", 1, LAKE, LAT, LON, ILND, ILV, -99, 0)

IF (IPC.EQ. -1) GOTO 900

CALL CART(FLOAT(LAT)/10., FLOAT(LON)/10., 2, I, J, IER)

CALL PRP(IPC, I/2, J/2, -99, ILND, IPRES, IT, -99, -99, -99, IWT, ILV)

GOTO 900

100 CONTINUE

900 RETURN

END

SUBROUTINE PCGR(IBC,IPC,ICLIN,IHR,CTIM)
 DIMENSION ICLIN(41), ISTD(3)

REV. 1.00

MAY, 1985

FORTAN IV/ REV 5.20

PEROUTKA/RANDEL
 DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
 RDOS/ REV. 6.18

CONTROLS DECODING AND PRINTING OF A COAST GUARD OBSERVATION.

ARGUMENT LIST

IBC I RDOS CHANNEL TO WHICH BROADCAST TEXT WILL BE WRITTEN.
 IPC I RDOS CHANNEL TO WHICH PLOT DATA WILL BE WRITTEN.
 ICLIN() I COMMAND LINE CONTAINING
 15-BYTE STATION NAME STARTING AT ICLIN(2),
 9-BYTE AFOS KEY STARTING AT ICLIN(10), AND
 3-BYTE STATION IDENTIFIER STARTING AT ICLIN(15).
 IHR I VALID TIME OF REQUIRED OBSERVATION.
 CTIM I (REAL) "CUT TIME" OF REQUIRED OBSERVATION. SEE LAWEB.FR.

EXTERNALS

ISHFT (FORT)	CGRDEC	CANDEC	H2ILX	PRB
GPIX	ILW2H	PRP	IOR (FORT)	

CHANNELS/FILES

IBC IS THE RDOS CHANNEL TO WHICH BROADCAST TEXT IS WRITTEN.
 IPC IS THE RDOS CHANNEL TO WHICH PLOT DATA IS WRITTEN.

VARIABLES

I TEMPORARY USED WHILE DECIDING IF STATION IS AMERICAN OR
 CANADIAN.
 ISKY SKY COVER IN OCTAS.
 ILX WEATHER IN ASCII FORM.
 IVSBY VISIBILITY IN SIXTEENTHS OF A MILE.
 ILND WIND VECTOR. INTEGER BETWEEN 0 AND 3699.
 ILV WAVE HEIGHT IN FEET.
 ITW WATER TEMPERATURE IN DEGREES FAHRENHEIT.
 IT AIR TEMPERATURE IN DEGREES FAHRENHEIT.
 IPRES SEA LEVEL PRESSURE IN TENTHS OF MILIBARS.
 LAT DUMMY FOR CANDEC.
 LON DUMMY FOR CANDEC.
 IW WMO WEATHER CODE (0 - 99).
 IPK DUMMY FOR CANDEC.
 ISTD() SIX-BYTE, SPACE PADDED STATION ID FOR GPIX.

```

  I = ISHFT(ICLIN(10),-8) ;CANADIAN OR AMERICAN?
  IF (I.EQ."<0>W".OR.I.EQ."<0>Y") GOTO 100
  CALL CGRDEC(ICLIN(10),ICLIN(15),CTIM,ISKY,ILX,IVSBY,ILND,ILV,
+           ITW,IT,IPRES) ;AMERICAN.
  GOTO 200
100 CALL CANDEC(ICLIN(10),ICLIN(15),3,IHR,CTIM,1,1,LAT,LON,IVSBY,
+           ISKY,ILND,IT,IPRES,IW,ITW,ILV,IPK)
  ILX = H2ILX(ILV)
200 IF (IBC.NE.-1) CALL PRB(IBC,ICLIN(2),0,0,0,0,ILND,ILV,IVSBY,ILD)
  IF (IPC.EQ.-1) GOTO 900
  IF (ISKY.EQ.-99.AND.ILND.EQ.-99) GOTO 900
  ISTD(1) = ICLIN(15)
  ISTD(2) = IOR(IAND(ICLIN(16),177400K),"<0> ")
  ISTD(3) = " "
  CALL GPIX(ISTD,I,J,IER)
  IF (IER.NE.1) GOTO 900
  IF (I.NE."<0>W".AND.I.NE."<0>Y") IW = ILW2H(ILX)
  CALL PRP(IPC,I,J,ISKY,ILND,IPRES,IT,-99,IW,IVSBY,ITW,ILV)
900 RETURN
END

```


SUBROUTINE PCMAN(IBC,IPC,ICLIN,IHR,CTIM)
 DIMENSION ICLIN(41)

REV. 1.00

MAY, 1985

FORTRAN IV/ REV 5.20

PEROUTKA/RANDEL

DG ECLIPSE (S230)

WSFO CLE FTS 942-4949

RDOS/ REV. 6.18

CONTROLS DECODING AND PRINTING OF A CMAN OBSERVATION.

ARGUMENT LIST

IBC I RDOS CHANNEL TO WHICH BROADCAST TEXT WILL BE WRITTEN.
 IPC I RDOS CHANNEL TO WHICH PLOT DATA WILL BE WRITTEN.
 ICLIN() I COMMAND LINE CONTAINING
 15-BYTE STATION NAME STARTING AT ICLIN(2),
 9-BYTE AFOS KEY STARTING AT ICLIN(10), AND
 5-BYTE STATION IDENTIFIER STARTING AT ICLIN(15).
 IHR I VALID TIME OF REQUIRED OBSERVATION.
 CTIM I (REAL) 'CUT TIME' OF REQUIRED OBSERVATION. SEE LAWEB.FR.

EXTERNALS

CANDEC PRB IOR (FORT) IAND (FORT) GPIX
 PRP

CHANNELS/FILES

IBC IS THE RDOS CHANNEL TO WHICH BROADCAST TEXT IS WRITTEN.
 IPC IS THE RDOS CHANNEL TO WHICH PLOT DATA IS WRITTEN.

VARIABLES

LAT DUMMY FOR CANDEC.
 LON DUMMY FOR CANDEC.
 IV DUMMY FOR CANDEC.
 ISKY DUMMY FOR CANDEC.
 ILND WIND VECTOR. INTEGER BETWEEN 0 AND 3699.
 IT AIR TEMPERATURE. DEGREES FAHRENHEIT.
 IPRES SEA LEVEL PRESSURE. TENTHS OF MILLIBARS.
 ILX DUMMY FOR CANDEC.
 IWT DUMMY FOR CANDEC.
 ILV DUMMY FOR CANDEC.
 IPK PEAK WIND SPEED IN KNOTS.
 I HORIZONTAL PIXEL COORDINATE.
 J VERTICAL PIXEL COORDINATE.

CALL CANDEC(ICLIN(10),ICLIN(15),5,IHR,CTIM,2,1,LAT,LON,IV,ISKY,
 +ILND,IT,IPRES,ILX,IWT,ILV,IPK)
 IF (IBC.NE.-1) CALL PRB(IBC,ICLIN(2),0,0,0,0,ILND,-99,-99,0)
 IF (IPC.EQ.-1) GOTO 100
 IF (ILND.EQ.-99.AND.IT.EQ.-99.AND.IPRES.EQ.-99)
 + GOTO 100
 ICLIN(17) = IOR(IAND(ICLIN(17),177400K),'<0>')
 CALL GPIX(ICLIN(15),I,J,IER)
 IF (IER.NE.1) GOTO 100
 CALL PRP(IPC,I,J,-99,ILND,IPRES,IT,-99,-99,-99,-99,-99)
 100 RETURN
 END

SUBROUTINE PKWND (IBGN, ISTOP, IUNIT, IUP, IPK)

MARCH 1985
REV 1.00

RANDELEROUTKA

WSFO CLE/FTS 942-4949

PURPOSE

DECODES PEAK WIND FROM SYNOPTIC OBSERVATIONS. SEARCHES THE ARRAY
IUP FOR A "333 921" GROUP FROM IBGN TO ISTOP. THE PEAK WIND
IS RETURNED TO CALLING PROGRAM IN WHOLE KNOTS.

ARGUMENT LIST

IUP () - INPUT ARRAY - WITH UNPACKED OBSERVATION.
IBGN - INPUT INTEGER - POINTER TO ARRAY IUP WHERE
TO BEGIN SEARCH FOR PEAK WIND
ISTOP - INPUT INTEGER - POINTER TO ARRAY IUP WHERE
TO STOP SEARCH FOR PEAK WIND
IUNIT - INPUT INTEGER - 1 IF OBSERVED WINDS IN KNOTS
OR 2 IF OBSERVED WIND IN MPS.
IWIND - OUTPUT INTEGER - PEAK WIND IN WHOLE KNOTS.

EXTERNALS

IUADEC - UTIL IFIX - FORT MASK

DIMENSION IUP(512)

IPK = -99

L = MASK(' <0>3<0>3<0>3<012><040><0>9<0>2<0>10909', 9, IUP, IBGN,
+ISTOP)

IF (L .LE. 0) GOTO 9999 ;CAN'T FIND 333 921

J = L + 7

IPK = IUADEC(IUP, J) ;DECODE WIND SPEED.

IF (IPK .LT. 0 .OR. IPK .GT. 99) IPK = -99 ;ERROR IN PK WND

IF (IPK .NE. 99) GOTO 100

IF IPK = 99 MUST LOOK FOR 2ND PK WND GROUP

L = MASK(' <0>9<0>2<0>10909', 4, IUP, L, ISTOP) ;SEARCH FOR 2ND 921 GRP

IF (L .LE. 0) GOTO 100 ;NO 2ND WIND GRP IPK = 99

J = L + 3

IPK = IUADEC(IUP, J)

IF (IPK .LT. 0 .OR. IPK .GT. 50) IPK = -99 ;ERROR IN PK WND OR WND

> 150 KNOTS - TOO LARGE

IF (IPK .NE. -99) IPK = IPK + 100

100 IF (IUNIT .EQ. 1) GOTO 9999 ;PK WIND ALREADY IN KNOTS

CONVERTING MPS TO KNOTS

SPD = IPK * 1.94 + .5

IPK = IFIX (SPD)

;CONVERT TO INTEGER

9999 RETURN

END

SUBROUTINE PLTLB(IPC, ICLIN, IHR, MONTH, IDAY, IYEAR)
DIMENSION ICLIN(41)

REV. 1.00

MAY, 1985

FORTRAN IV/ REV 5.20

PEROUTKA/RANDEL
DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.10

WRITES A LABEL RECORD TO THE PLOT FILE OPEN ON RDOS CHANNEL IPC. THE
FIRST EIGHT BYTES BEGINNING AT ICLIN(2) ARE TAKEN TO BE THE I AND J
PIXEL COORDINATES.

ARGUMENT LIST

IPC	I	RDOS CHANNEL TO WHICH HEADER WILL BE WRITTEN
ICLIN	I	COMMAND LINE FROM COMMAND FILE
IHR	I	VALID HOUR OF RUN
MONTH	I	VALID MONTH OF RUN
IDAY	I	VALID DAY OF MONTH OF RUN
IYEAR	I	VALID YEAR OF RUN

EXTERNALS

WRS (UTIL) PRNUM

CHANNELS/FILES

IPC IS THE RDOS CHANNEL TO WHICH THE LABEL IS WRITTEN

VARIABLES

N BYTE COUNTER FOR WRS

N = 4		:GET COORDINATES.
CALL WRS(IPC, ICLIN(2), N, IER)		
N = 1		
CALL WRS(IPC, " ", N, IER)		
N = 4		
CALL WRS(IPC, ICLIN(4), N, IER)		
N = 8		:LABEL INDICATOR.
CALL WRS(IPC, "00000Z", N, IER)		
CALL PRNUM(IPC, IHR, 10, 2, 0)		:HOUR
N = 2		
CALL WRS(IPC, "Z ", N, IER)		
CALL PRNUM(IPC, MONTH, 10, 2, 0)		:MONTH
N = 1		
CALL WRS(IPC, " / ", N, IER)		
CALL PRNUM(IPC, IDAY, 10, 2, 0)		:DAY
N = 1		
CALL WRS(IPC, " / ", N, IER)		
CALL PRNUM(IPC, IYEAR, 10, 2, 2)		:YEAR
RETURN		
END		

SUBROUTINE POMR(IBC,IPC,ICLIN,CTIM)
 DIMENSION ICLIN(41), ISTN(3)

REV. 1.00

MAY, 1985

FORTAN IV/ REV 5.20

PEROUTKA/RANDEL
 DG ECLIPSE (S230)

USFO CLE FTS 942-4949
 RDOS/ REV. 6.18

CONTROLS DECODING AND PRINTING OF AN OMR OBSERVATION.

ARGUMENT LIST

IBC I RDOS CHANNEL TO WHICH BROADCAST TEXT WILL BE WRITTEN.
 IPC I RDOS CHANNEL TO WHICH PLOT DATA WILL BE WRITTEN.
 ICLIN() I COMMAND LINE CONTAINING
 15-BYTE STATION NAME STARTING AT ICLIN(2),
 9-BYTE AFOS KEY STARTING AT ICLIN(10), AND
 3-BYTE STATION IDENTIFIER STARTING AT ICLIN(15).
 CTIM I (REAL) "CUT TIME" OF REQUIRED OBSERVATION. SEE LAWEB.FR.

EXTERNALS

OMRDEC PRB IAND (FORT) IOR (FORT) GPIX
 PRP

CHANNELS/FILES

IBC IS THE RDOS CHANNEL TO WHICH BROADCAST TEXT IS WRITTEN.
 IPC IS THE RDOS CHANNEL TO WHICH PLOT DATA IS WRITTEN.

VARIABLES

IWIND WIND VECTOR. INTEGER BETWEEN 0 AND 3699.
 IWV WAVE HEIGHT IN FEET.
 ISTN() SIX-BYTE, SPACE PADDED STATION ID FOR GPIX.
 I HORIZONTAL PIXEL COORDINATE.
 J VERTICAL PIXEL COORDINATE.

```

CALL OMRDEC(ICLIN(10),ICLIN(15),CTIM,IWIND,IWV);DECODE OB.
IF (IBC.EQ.-1) GOTO 100
CALL PRB(IBC,ICLIN(2),0,0,0,0,IWIND,IWV,-99,0);PRINT BROADCAST TEXT.
100 IF (IPC.EQ.-1) GOTO 900
IF (IWIND.EQ.-99.AND.IWV.EQ.-99) GOTO 900
ISTN(1) = ICLIN(15) ;ADD SPACE PADS TO STATION ID.
IF (IAND(ICLIN(16),377K).GT."<0>0") GOTO 200
ISTN(2) = IOR(IAND(ICLIN(16),177400K),"<0> ")
ISTN(3) = " "
GOTO 300
200 ISTN(2) = ICLIN(16)
ISTN(3) = IOR(IAND(ICLIN(17),177400K),"<0> ")
300 CALL GPIX(ISTN,I,J,IER)
IF (IER.NE.1) GOTO 900
CALL PRP(IPC,I,J,-99,IWIND,-99,-99,-99,-99,-99,-99,IWV)
900 RETURN
END

```



```

SUBROUTINE PPRES(IBC, ICLIN, IHR, CTIM)
DIMENSION ICLIN(41)
COMMON /PHILO/ IHI(3), LOW(3)
DATA IHI /-9999, 'MSG ' / LOW /9999, 'MSG ' /

C
C REV. 1.00
C
C MAY, 1985 PEROUTKA/RANDEL WSFO CLE FTS 942-4949
C FORTRAN IV/ REV 5.20 DG ECLIPSE (S230) RDOS/ REV. 6.10
C
C CONTROLS PRINTING OF ADDED PRESSURE SECTION.
C
C ARGUMENT LIST
C IBC I RDOS CHANNEL TO WHICH TEXT WILL BE WRITTEN.
C ICLIN() I COMMAND LINE CONTAINING 9-BYTE AFOS KEYS STARTING AT ICLIN(2),
C ICLIN(7), ..., AND ICLIN(32). IF NO KEYS ARE FOUND, THE HIGH
C AND LOW PRESSURES ARE PRINTED.
C IHR I VALID TIME OF REQUIRED OBSERVATIONS.
C CTIM I (REAL) "CUT TIME" OF REQUIRED OBSERVATION. SEE LAWE.B.FR.
C
C EXTERNALS
C ISHFT (FORT) ALTDEC LRS (UTIL) PRNUM
C
C CHANNELS/FILES
C IBC IS THE RDOS CHANNEL TO WHICH TEXT IS WRITTEN.
C
C VARIABLES
C I LOOP VARIABLE FOR AFOS KEYS.
C ISAVE HOLDS THE LAST VALUE OF I AFTER EXITING THE LOOP.
C I3 I + 3.
C I4 I + 4.
C IC THE FIRST BYTE OF THE CURRENT KEY.
C IALT ALTIMETER SETTING IN HUNDRETHS OF INCHES OF MERCURY.
C N BYTE COUNTER FOR LRS.
C IHI() HIGHEST PRESSURE IN IHI(1), 3-BYTE IDENTIFIER OF STATION
C BEGINNING AT IHI(2).
C LOW() LOWEST PRESSURE IN LOW(1), 3-BYTE IDENTIFIER OF STATION BEGIN-
C ING AT LOW(2).
C
DO 300 I = 2, 32, 5
  ISAVE = I
  I3 = I + 3
  I4 = I + 4
  IC = ISHFT(ICLIN(I), -8) ;CHECK FIRST CHARACTER OF KEY
  IF (IC.LT."<0>A".OR.IC.GT."<0>Z") GOTO 400
  CALL ALTDEC(ICLIN(I), IHR, CTIM, IALT) ;GET ALTIMETER
  N = 4 ;WRITE TO FILE.
  CALL LRS(IBC, ICLIN(I3), N, IER)
  IF (IALT.EQ.-99) GOTO 200
  CALL PRNUM(IBC, IALT, -1, 0, 0)
  N = 2
  CALL LRS(IBC, " ", N, IER)
  IF (IALT.GT.LOW(1)) GOTO 100 ;UPDATE COMMON BLOCK.
  LOW(1) = IALT
  LOW(2) = ICLIN(I3)
  LOW(3) = ICLIN(I4)
100 IF (IALT.LT.IHI(1)) GOTO 300
  IHI(1) = IALT
  IHI(2) = ICLIN(I3)
  IHI(3) = ICLIN(I4)
  GOTO 300
200 N = 6 ;ALTIMETER MISSING.
  CALL LRS(IBC, "### ", N, IER)
300 CONTINUE
400 IF (ISAVE.EQ.2) GOTO 500 ;END OF A NORMAL LINE.
  N = 2
  CALL LRS(IBC, "<15><12>", N, IER)
  GOTO 900
500 N = 19 ;NO KEYS. WRITE HIGH AND LOW.
  CALL LRS(IBC, "HIGHEST PRESSURE.. ", N, IER)
  CALL PRNUM(IBC, IHI(1), -1, 0, 0)
  N = 4
  CALL LRS(IBC, " AT ", N, IER)
  N = 4
  CALL LRS(IBC, IHI(2), N, IER)
  N = 2
  CALL LRS(IBC, "<15><12>", N, IER)
  N = 18
  CALL LRS(IBC, "LOWEST PRESSURE.. ", N, IER)
  CALL PRNUM(IBC, LOW(1), -1, 0, 0)
  N = 4
  CALL LRS(IBC, " AT ", N, IER)
  N = 4
  CALL LRS(IBC, LOW(2), N, IER)
  N = 2
  CALL LRS(IBC, "<15><12>", N, IER)
900 RETURN
END

```

```

C REV. 1.00
C
C MAY, 1985 PEROUTKA/RANDEL WSFO CLE FTS 942-4949
C FORTRAN IV/ REV 5.20 DG ECLIPSE (S238) RDOS/ REV. 6.18
C
C WRITES A LINE OF BROADCAST TEXT TO RDOS CHANNEL IBC.
C
C ARGUMENT LIST
C IBC I RDQS CHANNEL TO WHICH TEXT WILL BE WRITTEN
C ISTN I 15-BYTE STRING CONTAINING STATION NAME. (NULL FOR SHIPS
C AND BUOYS.)
C IBS I DETERMINES TYPE OF REPORT.
C 0 FOR LAND STATION
C 1 FOR BUOY
C 2 FOR SHIP
C LAKE I (SHIPS AND BUOYS ONLY) LAKE WHERE REPORT ORIGINATED. (VALUES
C DEFINED IN LAKLOC.FR)
C LAT I (SHIPS AND BUOYS ONLY) LATITUDE IN TENTHS OF A DEGREE NORTH.
C LON I (SHIPS AND BUOYS ONLY) LONGITUDE IN TENTHS OF A DEGREE WEST.
C IWND I WIND VECTOR. INTEGER BETWEEN 0 AND 3699.
C ILV I WAVE HEIGHT IN FEET.
C IVSBY I VISIBILITY IN SIXTEENTHS OF A MILE.
C ILX I WEATHER IN ASCII FORM. TWO BYTES, SPACE-PADDED OF " ", "T",
C "2R", "2L", "R", "S", "RS", "L", "F", "D", "BY", "H", OR "K".
C
C EXTERNALS
C LRS (UTIL) LLL PRNUM LVSBY LLX
C
C CHANNELS/FILES
C IBC IS THE RDOS CHANNEL TO WHICH THE LINE IS WRITTEN.
C
C VARIABLES
C N BYTE COUNTER FOR LRS
C IDIR WIND DIRECTION IN DEGREES
C ISP WIND SPEED IN KNOTS
C
C IF (IBS.NE.0) GOTO 100 ;LAND STATION
C N = 15 ;STATION NAME
C CALL LRS(IBC,ISTN,N,IER)
C N = 1
C CALL LRS(IBC," ",N,IER)
C GOTO 200
C 100 CALL LLL(IBC,IBS,LAKE,LAT,LON) ;LAT/LON FOR BUOY AND SHIP
C 200 IF (IWND.NE.-99) GOTO 300 ;WIND GROUP
C N = 6
C CALL LRS(IBC,"TTT/TT",N,IER)
C GOTO 400
C 300 IDIR = IWND/100*10 ;FIND DIRECTION AND SPEED.
C ISP = IWND - IDIR*10
C IF (IDIR.LE.360) GOTO 325 ;SPEED OF 100 KNOTS OR GREATER.
C IDIR = IDIR - 500
C ISP = ISP + 100
C 325 IF (IDIR.NE.490) GOTO 350 ;SPECIAL CASE: VARIABLE DIRECTION.
C N = 4
C CALL LRS(IBC,"VRB/",N,IER)
C GOTO 375
C 350 CALL PRNUM(IBC,IDIR,100,2,0) ;WRITE DIRECTION.
C N = 1
C CALL LRS(IBC,"/",N,IER)
C 375 N = 10 ;WRITE SPEED.
C IF (ISP.GE.100) N = 100
C CALL PRNUM(IBC,ISP,N,2,0)
C 400 IF (ILV.EQ.-99) GOTO 500 ;WAVE GROUP
C N = 2
C CALL LRS(IBC," ",N,IER)
C CALL PRNUM(IBC,ILV,100,1,0)
C N = 9
C CALL LRS(IBC," FT WAVES",N,IER)
C 500 IF (IVSBY.EQ.-99.OR.IVSBY.GE.64) GOTO 700 ;VISIBILITY...
C IF (IBS.EQ.0) GOTO 600 ;PUT VSBY AND LX ON NEXT LINE IF
C N = 2 ;SHIP OR BUOY.
C CALL LRS(IBC,"<15><12>",N,IER)
C 600 N = 8
C CALL LRS(IBC," VSBY ",N,IER)
C CALL LVSBY(IBC,IVSBY)
C N = 7
C CALL LRS(IBC," MI IN ",N,IER)
C CALL LLX(IBC,IWD0) ;...AND WEATHER
C 700 N = 2
C CALL LRS(IBC,"<15><12>",N,IER)
C RETURN
C END

```


SUBROUTINE PRES(IBGN,ISTOP,IUP,IPRES)

MARCH 1985
REV 1.00

RANDEL/PEROUTKA

WSFO CLE/FTS 942-4949

PURPOSE

DECODES PRESSURE FROM SYNOPTIC OBSERVATIONS. SEARCHES THE ARRAY IUP FOR A ~~40000~~ GROUP FROM IBGN+8 TO ISTOP. IBGN MUST POINT TO THE SPACE BEFORE VSBY GROUP. ISTOP SHOULD POINT TO THE MARINE DATA GROUP 222**.

ARGUMENT LIST

IUP() - INPUT ARRAY - WITH UNPACKED OBSERVATION.
IBGN - INPUT INTEGER - POINTER TO ARRAY IUP WHERE
TO BEGIN SEARCH FOR PRES. SHOULD POINT TO
SPACE BEFORE VSBY GROUP
ISTOP - INPUT INTEGER - POINTER TO ARRAY IUP WHERE
TO STOP SEARCH. SHOULD BE MARINE DATA INDICATOR
222** OR END OF OB.
IPRES - OUTPUT INTEGER - PRESSURE IN TENTHS OF DEGREES

EXTERNALS

IUADEC - UTIL IFIX - FORT MASK

DIMENSION IUP(512)

IPRES = -99

L = MASK('<0>409090909',5,IUP,IBGN+8,ISTOP);POINT TO PRES GROUP
IF(L.LE.0) GOTO 9999 ;CAN'T FIND PRES GROUP
J = L +1

IPRES = IUADEC(IUP,J) ;DECODE PRESSURE
IF(IPRES.LE.1000) IPRES = IPRES +10000 ;PRES > 1000MB

CHECK TO SEE IN PRESSURE IS TOO LARGE OR SMALL

IF(IPRES.GT.11000.OR.IPRES.LT.8500) IPRES = -99 ;

9999 RETURN
END

COMMON /PSOLDT/ ILDT(99)
 DATA ILDT /3*0.5*1.2,1.9*1.2,4*2.3*4.3*2,1.2,2.1,2.4*1.2,
 +1.2,2.1,1.2,2.3*1.2,1.4,3.4,3*1.3,1.2,1.4,3.4,3*1.3,1,
 +2.1,4.3,4,4*1.3,18*3,4*2,5*3/
 REV. 1.00

MAY, 1985 PEROUTKA/RANDEL WSFO CLE FTS 942-4949
 FORTRAN IV/ REV 5.20 DG ECLIPSE (S238) RDOS/ REV. 6.18

WRITES A LINE OF PLOT DATA TO RDOS CHANNEL IBC.

ARGUMENT LIST

IPC I RDOS CHANNEL TO WHICH DATA WILL BE WRITTEN
 I I HORIZONTAL PIXEL COORDINATE.
 J I VERTICAL PIXEL COORDINATE.
 ISKY I SKY COVER IN OCTAS.
 ILND I WIND VECTOR. INTEGER BETWEEN 0 AND 3699.
 IPRES I SEA LEVEL PRESSURE IN TENTHS OF MILLIBARS.
 IT I AIR TEMPERATURE IN DEGREES FAHRENHEIT.
 ITD I DEW POINT IN DEGREES FAHRENHEIT.
 ILX I WMO WEATHER CODE (0 - 99).
 IVSBY I VISIBILITY IN SIXTEENTHS OF A MILE.
 ITU I SEA SURFACE TEMPERATURE IN DEGREES FAHRENHEIT.
 ILV I WAVE HEIGHT IN FEET.

EXTERNALS

PRNUM LRS (UTIL)

CHANNELS/FILES

IPC IS THE RDOS CHANNEL TO WHICH THE LINE IS WRITTEN.

VARIABLES

ILDT() FONT SIZE NEEDED TO PLOT A GIVEN WEATHER CODE.
 = 0 FOR NO WEATHER CODE TO PLOT.
 = 1 FOR NORMAL FONT SIZE.
 = 2 FOR DOUBLE-WIDTH FONT.
 = 3 FOR DOUBLE-HEIGHT FONT, AND
 = 4 FOR DOUBLE- HEIGHT AND WIDTH FONT.
 N BYTE COUNTER FOR LRS.
 I GENERAL-PURPOSE TEMPORARY VARIABLE.

CALL PRNUM(IPC,I,1000,2,1) ;COORDINATES
 CALL PRNUM(IPC,J,1000,2,1)
 N = 3
 CALL LRS(IPC,'000',N,IER) ;PSO FROM PSOLDT
 I = 0
 IF (ILX.GT.0.AND.ILX.LT.100) I = ILDT(ILX)
 CALL PRNUM(IPC,I,1,2,0) ;W FROM PSOLDT
 N = 5
 CALL LRS(IPC,'02...',N,IER) ;DT FROM PSOLDT
 IF (ISKY.EQ.-99) GOTO 100 ;SKY COVER
 IF (ISKY.EQ.2) ISKY = 3 ;!!!FIX GDM BUG!!!
 CALL PRNUM(IPC,ISKY,1,0,1)
 GOTO 200
 100 N = 2
 CALL LRS(IPC,'M',N,IER)
 200 IF (ILND.NE.-99) GOTO 300 ;WIND
 CALL PRNUM(IPC,ILND,0,0,1)
 GOTO 400
 300 ID = ILND/100
 IS = ILND - 100*ID
 IF (ID.LE.36) GOTO 350 ;SPEEDS GREATER THAN 100 KNOTS.
 ID = ID - 50
 IS = IS + 100
 350 CALL PRNUM(IPC,ID,10,2,0)
 CALL PRNUM(IPC,IS,100,2,1)
 400 CALL PRNUM(IPC,IPRES,100,2,1) ;SEA LEVEL PRESSURE
 CALL PRNUM(IPC,IT,0,0,1) ;AIR TEMPERATURE
 CALL PRNUM(IPC,ITD,0,0,1) ;DEWPOINT TEMPERATURE
 I = ILX ;WEATHER CODE
 IF (I.LT.4.OR.I.GT.99) I = -99
 CALL PRNUM(IPC,I,0,0,1)
 I = IVSBY ;VISIBILITY
 IF (I.EQ.64) I = 57 ;FOUR MILES
 IF (I.EQ.80) I = 50 ;FIVE MILES
 IF (I.EQ.96) I = 60 ;SIX MILES
 IF (I.GT.96) I = -99 ;SEVEN OR GREATER
 CALL PRNUM(IPC,I,0,0,1)
 N = 8 ;EXTRA FIELDS.
 CALL LRS(IPC,'.....',N,IER)
 CALL PRNUM(IPC,ILV,0,0,1) ;WAVE
 CALL PRNUM(IPC,-99,0,0,1) ;SWELL
 CALL PRNUM(IPC,ITU,0,0,2) ;WATER TEMPERATURE
 RETURN
 END

SUBROUTINE PSA(IBC,IPC,ICLIN,CTIM)
DIMENSION ICLIN(41)

REV..1.00

MAY, 1985
FORTRAN IV/ REV 5.28

PEROUTKA/RANDEL
DG ECLIPSE (S238)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

CONTROLS THE DECODING AND PRINTING OF A SURFACE AIRWAYS OBSERVATION.

ARGUMENT LIST

IBC I RDOS CHANNEL OF BROADCAST FILE.
IPC I RDOS CHANNEL OF PLOT FILE.
ICLIN I LINE FROM COMMAND FILE.
CTIM I FLOATING-POINT "CUT TIME" OF OBSERVATION TO BE TAKEN.

EXTERNALS

SADC N2ILX PRB IOR (FORT) GPIX
PRP

CHANNELS/FILES

IBC RDOS CHANNEL ON WHICH BROADCAST TEXT FILE IS OPENED.
IPC RDOS CHANNEL ON WHICH PLOT FILE IS OPENED.

VARIABLES

IV VISIBILITY.
ISKY SKY COVER.
IWND WIND VECTOR.
IT TEMPERATURE.
ITD DEW POINT.
IPRES SEA LEVEL PRESSURE.
IW WMO WEATHER CODE.
ILX TWO-BYTE ASCII WEATHER CODE.
I HORIZONTAL PIXEL COORDINATE.
J VERTICAL PIXEL COORDINATE.

CALL SADC(ICLIN(18),CTIM,IV,ISKY,IWND,IT,ITD,IPRES,IW)
ILX = N2ILX(IW)

IF (IBC.NE.-1) CALL PRB(IBC,ICLIN(2),0.0,0.0,IWND,-99,IV,ILX)
IF (IPC.EQ.-1) GOTO 100

IF (IWND.EQ.-99.AND.IT.EQ.-99.AND.IPRES.EQ.-99)

+ GOTO 100

ICLIN(14) = IOR(IAND(ICLIN(14),177480K),'<0>')

ICLIN(15) = .

CALL GPIX(ICLIN(13),I,J,IER)

IF (IER.NE.1) GOTO 100

CALL PRP(IPC,I,J,ISKY,IWND,IPRES,IT,ITD,IW,-99,-99)

100 RETURN

END


```

SUBROUTINE PSHP(IBC,IPC,ICLIN)
DIMENSION ICLIN(41)
COMMON /SHP/ ISHP(50,11), NSHP

```

REV. 1.00

MAY, 1985

FORTTRAN IV/ REV 5.20

PEROUTKA/RANDEL
DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

CONTROLS PRINTING OF SHIP REPORTS FOR A LAKE.

ARGUMENT LIST

IBC I RDOS CHANNEL TO WHICH BROADCAST TEXT WILL BE WRITTEN.
IPC I RDOS CHANNEL TO WHICH PLOT DATA WILL BE WRITTEN.
ICLIN() I COMMAND LINE CONTAINING
LAKE NUMBER IN TWO ASCII BYTES STARTING AT ICLIN(2).

EXTERNALS

IAND (FORT) N2ILX PRB CART (AG) FLOAT (FORT)
PRP

CHANNELS/FILES

IBC IS THE RDOS CHANNEL TO WHICH BROADCAST TEXT IS WRITTEN.
IPC IS THE RDOS CHANNEL TO WHICH PLOT DATA IS WRITTEN.

VARIABLES

ISHP() ARRAY OF DECODED SHIP OBSERVATIONS CONTAINING
1 LAKE NUMBER,
2 LATITUDE IN TENTHS OF DEGREES NORTH,
3 LONGITUDE IN TENTHS OF DEGREES WEST,
4 VISIBILITY IN SIXTEENTHS OF A MILE,
5 SKY COVER IN OCTAS,
6 WIND VECTOR (INTEGER BETWEEN 0 AND 3699),
7 AIR TEMPERATURE IN DEGREES FAHRENHEIT,
8 SEA LEVEL PRESSURE IN TENTHS OF MILLIBARS,
9 WMO WEATHER CODE (0 - 99),
10 WATER TEMPERATURE IN DEGREES FAHRENHEIT, AND
11 WAVE HEIGHT IN FEET.
NSHP NUMBER OF OBSERVATIONS IN ISHP.
LAKE LAKE NUMBER. 1 = LS, 2 = LM, 3 = LH, 4 = LE, 5 = LO.
I LOOP COUNTER FOR SHIP OBSERVATIONS.
ILX WEATHER CODE IN ASCII FORM.
J HORIZONTAL PIXEL COORDINATE.
K VERTICAL PIXEL COORDINATE.

LAKE = IAND(ICLIN(2),377K) - 48 ;GET LAKE NUMBER.

IF (LAKE.LT.1.OR.LAKE.GT.5) GOTO 900

DO 200 I = 1, NSHP ;LOOP FOR EACH SHIP.

IF (ISHP(I,1).NE.LAKE) GOTO 200 ;RIGHT LAKE?

IF (IBC.EQ.-1) GOTO 100

ILX = N2ILX(ISHP(I,9)) ;BROADCAST SECTION.

CALL PRB(IBC,"",2,LAKE,ISHP(I,2),ISHP(I,3),ISHP(I,6),
ISHP(I,11),ISHP(I,4),ILX)

100 IF (IPC.EQ.-1) GOTO 200 ;PLOT SECTION.

CALL CART(FLOAT(ISHP(I,2))/10.,FLOAT(ISHP(I,3))/10.,2,J,K,IER)

CALL PRP(IPC,J/2,K/2,ISHP(I,5),ISHP(I,6),ISHP(I,8),ISHP(I,7),
-99,ISHP(I,9),ISHP(I,4),ISHP(I,10),ISHP(I,11))

200 CONTINUE

900 RETURN

END

SUBROUTINE PSYN(IBC,IPC,ICLIN,IHR,CTIM)
DIMENSION ICLIN(41)

REV. 1.00

MAY, 1985

FORTAN IV/ REV 5.20

PEROUTKA/RANDEL
DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

CONTROLS DECODING AND PRINTING OF A SURFACE SYNOPTIC OBSERVATION.

ARGUMENT LIST

IBC I RDOS CHANNEL TO WHICH BROADCAST TEXT WILL BE WRITTEN.
IPC I RDOS CHANNEL TO WHICH PLOT DATA WILL BE WRITTEN.
ICLIN() I COMMAND LINE CONTAINING
15-BYTE STATION NAME STARTING AT ICLIN(2),
9-BYTE AFOS KEY STARTING AT ICLIN(10), AND
5-BYTE STATION ID STARTING AT ICLIN(15).
IHR I VALID TIME OF REQUIRED OBSERVATION.
CTIM I (REAL) "CUT TIME" OF REQUIRED OBSERVATION. SEE LAWEB.FR.

EXTERNALS

CANDEC PRB CART (AG) IOR (FORT) GPIX
PRP

CHANNELS/FILES

IBC IS THE RDOS CHANNEL TO WHICH BROADCAST TEXT IS WRITTEN.
IPC IS THE RDOS CHANNEL TO WHICH PLOT DATA IS WRITTEN.

VARIABLES

LAT DUMMY FOR CANDEC.
LON DUMMY FOR CANDEC.
IV VISIBILITY.
ISKY SKY COVER.
IWND WIND VECTOR. INTEGER BETWEEN 0 AND 3699.
IT AIR TEMPERATURE IN DEGREES FAHRENHEIT.
IPRES SEA LEVEL PRESSURE IN TENTHS OF MILIBARS.
IWX WMO WEATHER CODE (0-99).
IWT WATER TEMPERATURE IN DEGREES FAHRENHEIT.
IWV WAVE HEIGHT IN FEET.
IPK DUMMY FOR CANDEC.
I HORIZONTAL PIXEL COORDINATE.
J VERTICAL PIXEL COORDINATE.

CALL CANDEC(ICLIN(10),ICLIN(15),5,IHR,CTIM,3,1,LAT,LON,IV,ISKY,
+IWND,IT,IPRES,IWX,IWT,IWV,IPK)

IF (IBC.NE.-1) CALL PRB(IBC,ICLIN(2),0,0,0,0,IWND,-99,-99,0)

IF (IPC.EQ.-1) GOTO 100

IF (IWND.EQ.-99.AND.IT.EQ.-99.AND.IPRES.EQ.-99)

+ GOTO 100

ICLIN(17) = IOR(IAND(ICLIN(17),177400K), "<0> ")

CALL GPIX(ICLIN(15),I,J,IER)

IF (IER.NE.1) GOTO 100

CALL PRP(IPC,I,J,-99,IWND,IPRES,IT,-99,-99,-99,-99)

100 RETURN

END

SUBROUTINE SADC(KEY,CTIM,IVSBY,ISKY,IWIND,IT,ITD,IPRES,IWDO
 DIMENSION KEY(5), KREC(20), IBUF(128), ITDAT(256)

REV. 1.00

MAY, 1985

PEROUTKA/RANDEL
 DG ECLIPSE (S238)

WSFO CLE FTS 942-4949
 RDOS/ REV. 6.18

PROVIDES AN INTERFACE FOR TO SURFACE AIRWAYS DECODER AIRDX.

ARGUMENT LIST

KEY() I AFOS KEY OF PRODUCT TO BE DECODED.
 CTIM I FLOATING POINT 'CUT TIME' OF DESIRED OBSERVATION.
 IVSBY 0 VISIBILITY IN SIXTEENTHS OF A MILE.
 ISKY 0 SKY COVER IN OCTAS.
 IWIND 0 INTEGER REPRESENTING WIND VECTOR.
 IT 0 TEMPERATURE.
 ITD 0 DEW POINT.
 IPRES 0 SEA LEVEL PRESSURE IN TENTHS OF MILLIBARS.
 IWDO 0 WMO WEATHER CODE (8-99)

EXTERNALS

IAND (FORT) GBLK8 UNPACK(UTIL) PRVRF (BG) ADXX
 IFIX (FORT) SCHWX

CHANNELS/FILES

NONE

VARIABLES

K1 FIRST LETTER OF THE KEY. USED TO DETERMINE IF CANADIAN.
 ICAN CANADIAN/US INDICATOR.
 KREC() KEY RECORD.
 NVER NUMBER OF VERSIONS AVAILABLE.
 IBUF() BUFFER FOR BLOCK FROM DATABASE.
 ITDAT() BUFFER FOR UNPACKED BLOCK FROM DATABASE.
 IV1 TEMPORARY FOR VISIBILITY CONVERSION.
 IV2 TEMPORARY FOR VISIBILITY CONVERSION.

```

      IVSBY = -99
      ISKY = -99
      IWIND = -99
      IT = -99
      ITD = -99
      IPRES = -99
      IWDO = ''
      K1 = IAND(KEY(1),177400K)
      ICAN = 0
      IF (K1.EQ.'Y'.OR.K1.EQ.'U') ICAN = 1
      NVER = 0
100 CALL GBLK8(KEY,CTIM*15.,CTIM*70.,NVER,IBUF,IER)      ;SEARCH DATABASE.
      IF (IER.NE.1) GOTO 900
      CALL UNPACK(IBUF,256,ITDAT)
      CALL ADXX(ITDAT,256,IBUF,128,ICAN,.TRUE.,IER)      ;DECODE OBSERVATION.
      IF (IBUF(5).EQ.3) GOTO 100
      IVSBY = IBUF(21)      ;CONVERT VSBY TO 1/16THS OF A MILE.
      IF (IVSBY.EQ.-99) GOTO 300
      IF (IVSBY.GT.8) IVSBY = IVSBY*16
      IF (IVSBY.GT.8) GOTO 300
      IV1 = -IVSBY/1000
      IV2 = -IVSBY - IV1*1000
      IVSBY = IV1*16 + IV2*16/1000
300 ISKY = IBUF(9)      ;CONVERT SKY COVER TO OCTAS.
      IF (ISKY.EQ.'<0> ') ISKY = -99
      IF (ISKY.EQ.'<0>C') ISKY = 0
      IF (ISKY.EQ.'<0>S') ISKY = 3
      IF (ISKY.EQ.'<0>8') ISKY = 6
      IF (ISKY.EQ.'<0>0') ISKY = 8
      IF (ISKY.EQ.'<0>X') ISKY = 9
      IF (IBUF(32).EQ.-99.OR.IBUF(33).EQ.-99) GOTO 400;WIND.
      IWIND = IBUF(32)*10 + IBUF(33)
      GOTO 500
400 IWIND = -99
500 IT = IBUF(29)      ;TEMPERATURES.
      ITD = IBUF(30)
      IPRES = IBUF(28)      ;TAKE EITHER PRESSURE FOR IPRES.
      IF (IPRES.EQ.-99.AND.IBUF(35).NE.-99)
      + IPRES = IFIX(FLOAT(IBUF(35))*3.38638 + .5)
      CALL SCHWX(IBUF,IWDO)      ;GET WMO CODE NUMBER FROM WEATHER
900 RETURN      ;STRING.
      END

```

```

SUBROUTINE SCWX(IBUF,ILX)
  DIMENSION MSK(6), IBUF(128), IUP(10)
  COMMON /SAWM/ ISAWM(4,27)
  DATA ISAWM /-1,-1,"<0>T",95,"<0>S<0>W<0>",-1,85,-1,"<0>S<0>W",86,
+ "<0>R<0>W<0>+",82,"<0>R<0>W<0>",-1,80,-1,"<0>R<0>W",81,
+1,"<0>I<0>P",79,-1,"<0>S<0>+",75,-1,"<0>S<0>",-1,71,
+ "<0>S<0>B<0>S",73,-1,"<0>B<0>S",38,-1,-1,"<0>S",73,
+ "<0>Z<0>R<0>",-1,66,-1,"<0>Z<0>R",67,-1,"<0>R<0>+",65,
+1,"<0>R<0>",-1,61,-1,-1,"<0>R",63,"<0>Z<0>L<0>",-1,56,
+1,"<0>Z<0>L",57,-1,"<0>L<0>+",55,-1,"<0>L<0>",-1,51,
+1,-1,"<0>L",53,-1,"<0>BDN",32,-1,-1,"<0>F",10,-1,-1,"<0>D",6,
+1,-1,"<0>H",5,-1,-1,"<0>K",4/

```

REV. 1.00

MAY, 1985

PEROUTKA/RANDEL
DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

SCANS DECODED OBSERVATION ARRAY FOR WEATHER AND OBSTRUCTIONS TO
VISIBILITY. RETURNS APPROPRIATE WMO CODE NUMBER.

ARGUMENT LIST

IBUF I DECODED OBSERVATION ARRAY FROM AIRDX.
ILX 0 WMO CODE NUMBER. -99 FOR ERROR OR NO WEATHER.

EXTERNALS

UNPACK(UTIL) MASK

CHANNELS/FILES
NONE

VARIABLES

IUP() UNPACKED WEATHER FROM IBUF.
I LOOP VARIABLE.
ISAWM() DATA ARRAY. EACH ENTRY HAS SIX BYTES OF MASK AND TWO BYTES
OF WMO CODE NUMBER.

```

ILX = -99                                ;DEFAULT.
IF (IBUF(27).EQ." ") GOTO 900           ;NO WEATHER.
CALL UNPACK(IBUF(23),10,IUP)            ;UNPACK WEATHER.
DO 100 I = 1, 27                        ;LOOP FOR EACH WEATHER TYPE.
  ILX = ISAWM(4,I)
  IF (MASK(ISAWM(1,I),3,IUP,1,10).NE.-1) GOTO 200
100 CONTINUE
  GOTO 900
200 IF (ILX.NE.85.AND.ILX.NE.86) GOTO 300;CHECK FOR MIXTURES.
  IF (MASK("<0>S",1,IUP,1,10).NE.-1.AND.
+ MASK("<0>R",1,IUP,1,10).NE.-1) ILX = 68
300 IF (MASK("<0>S<0>W",2,IUP,1,10).NE.-1.AND.
+ MASK("<0>R<0>W",2,IUP,1,10).NE.-1) ILX = 83
900 RETURN
END

```



```

C.....PURPOSE OF SUBROUTINE IS TO FIND DISDANCE AND ASMUTH
C.....FROM GIVEN SOURCE LOCATION
SUBROUTINE SHPLOC(LAKE,LATSH,LONSH,HDIS,IDEGRE,JLAT)
COMMON /DISLOC/SOLAT(25),SOLON(25),IBB(5),IEE(5)
REAL LATSH,LONSH,LATSO,LONSO,LATSHR
ARCCOS(X)=ATAN(SQRT(1-X**2)/X)
C...LATSH =LAT OF SHIP ;LONSH = LON OF SHIP ,LATSO = LAT OF SOURCE
C...LONSO = LON OF SOURCE
C... HDIS = CLOSES DISDANCE OF SHIP TO SOURCE
TWOPHI=6.283185
IB=IBB(LAKE)
IE=IEE(LAKE)
CONDEG=TWOPHI/360.
C...IB = FIRST LOC OF LAKE ;IE = LAST LOCATION OF LAKE
HDIS=1000
RADADJ=57.295779
C...RADADJ = CONVERSION FACTOR OF RADIANDS TO DEGREES
J=0
C ....FOLLOWING DO LOOP DETERMINES THE CLOSEST DISTANCE TO SOURCE STATION
DO 10 I=IB,IE
LATSO = SOLAT(I)*CONDEG
LONSO=SOLON(I)*CONDEG
DIFF=(LONSH-SOLON(I))*CONDEG
LATSHR=LATSH*CONDEG
D=SIN(LATSO)*SIN(LATSHR)+COS(LATSO)*COS(LATSHR)*COS(DIFF)
IF (D .GE. 0.9999999)GO TO 20 .
D=ARCCOS(D)*RADADJ*60.
IF(D .GT. HDIS)GO TO 10
HDIS=D
J=I
10 CONTINUE
ANGLE=0
IF(LONSH-SOLON(J) .EQ. 0.)GO TO 12
LATSO=SOLAT(J)*CONDEG
LONSO=SOLON(J)*CONDEG
DIFF=(LONSH-SOLON(J))*CONDEG
XDIS=SIN(LATSO)**2 +COS(LATSO)**2*COS(DIFF)
IF(XDIS .GT. 1.)XDIS=1.
XDIS=ARCCOS(XDIS)*RADADJ*60.
IF(XDIS .GT. HDIS)XDIS=HDIS
IF(XDIS .EQ. 0)GO TO 12
ANGLE=ARCCOS(XDIS/HDIS)*RADADJ
12 QUAD=90
SGN=1
LONSO=SOLON(J)
LATSO=SOLAT(J)
IF(LONSO .LT. LONSH)QUAD=270
IF((LATSO .LT. LATSH) .AND. (QUAD .EQ. 90.))SGN=-1
IF((LATSO .GT. LATSH) .AND. (QUAD .EQ. 270.))SGN=-1
IF((LONSO .EQ. LONSH) .AND. (LATSO .LT. LATSH)) QUAD=0 ;CORR
IF((LONSO .EQ. LONSH) .AND. (LATSO .GT. LATSH)) QUAD=180 ;CORR

DEGRE=QUAD + SGN*ANGLE
JLAT=J
IDEGRE=(DEGRE+ 5.)/10.
IF(IDEGRE .EQ. 0)IDEGRE=36
GO TO 30
C ....FOLLOWING FOR WHEN SOURCE AND SHIP AT THE SAME LOCATION
20 JLAT=I
IDEGRE=37
HDIS=0
30 RETURN
END

```

SUBROUTINE SWIND(IBGN,IUNIT,IUP,IWIND)

JUNE 1985
REV 1.00

RANDEL/PEROUTKA

WSFO CLE/FTS 942-4949

PURPOSE

DECODES THE WIND GROUP FROM SYNOPTIC OBSERVATIONS. SEARCHES ARRAY IUP FOR THE WIND 8 ELEMENTS FROM IBGN. IBGN MUST POINT TO SPACE BEFORE VSBY GROUP. IUNIT DESCRIBES THE OBSERVED WIND SPEED UNIT. 1 FOR KNOTS, AND 2 FOR MPS.

ARGUMENT LIST

IUP() - INPUT ARRAY - WITH UNPACKED OBSERVATION.
IBGN - INPUT INTEGER - POINTER TO ARRAY IUP WHERE TO BEGIN SEARCH FOR WIND. MUST POINT TO SPACE BEFORE VSBY GROUP.
IUNIT - INPUT INTEGER - 1 IF OBSERVED WINDS IN KNOTS OR 2 IF OBSERVED WINDS IN MPS.
IWIND - OUTPUT INTEGER - WIND DIRECTION AND SEED.

EXTERNALS

IUADEC - UTIL MASK
IFIX - FORT

DIMENSION IUP(512)

IWIND = -99

L = MASK('89898989<12><40>',5,IUP,IBGN+8,IBGN+12) ;FIND WIND GROUP
IF (L.LE.0) GOTO 9999
IWIND = IUADEC(IUP,L) ;DECODE WIND GROUP

CHECK FOR LEGAL WIND VALUE. 0000 - 3699 NOMRAL WIND,
5000 - 8699 WIND SPEED GREATER THAN 100, 9900 - 9999
WIND DIRECTION INDETERMINATE

IF((IWIND .LT. 0 .OR. IWIND .GT. 9999) .OR. (IWIND .GT. 3699
+.AND. IWIND .LT. 5000) .OR. (IWIND .GT. 8699 .AND. IWIND .LT.
+9900)) IWIND = -99

IF(IUNIT .EQ. 1 .OR. IWIND .EQ. -99)GOTO 9999 ;WIND SPEED IN KNOTS
;NO CONVERSION NEEDED

CONVERTING FROM MPS TO KNOTS

ISPD = IWIND - (IWIND/100 *100) ;DECODE WIND SPEED
SPD = ISPD * 1.94 + .5 ;CONVERTING FROM MPS TO KNOTS
ISPD = IFIX(SPD) ;CONVERT TO INTEGER
IWIND = (IWIND/100 * 100) + ISPD ;COMBINE SPEED AND DIRECTION

9999 RETURN
END


```

SUBROUTINE SLX(IGNH,ISTOP,IUP,IV,ILX,ISKY)
C
C MARCH 1985          RANDEL/PEROUTKA      WSFO CLE/FTS 942-4949
C REV 1.00
C
C PURPOSE
C THIS SUBROUTINE DECODES THE PRESENT WEATHER, VISIBILITY AND SKY
C COVER IN SYNOPTIC OBSERVATIONS. IT SEARCHES THE ARRAY IUP
C STARTING AT 'IGNH' FOR THE VARIOUS ELEMENTS. IGNH MUST POINT TO THE
C SPACE BEFORE THE VISIBILITY GROUP. THE SUBROUTINE RETURNS THE
C PRESENT WEATHER, ILX; VISIBILITY, IV; AND AMOUNT OF SKY COVER, ISKY;
C TO THE CALLING PROGRAM.
C
C ARGUMENT LIST
C IUP( ) - INPUT ARRAY - WITH UNPACKED OBSERVATION.
C IGNH - INPUT INTER - POINTER TO ARRAY IUP WHERE TO BEGIN
C SEARCH. MUST POINT TO SPACE BEFORE VSBY GROUP
C ISTOP - INPUT INTEGER - POINTER TO ARRAY IUP WHERE TO STOP
C SEARCH. SHOULD BE MARINE DATA GROUP 222*** OR END OF
C OBSERVATION
C IV - OUTPUT INTEGER - VISIBILITY IN SIXTEENTHS OF A MILE
C ILX - OUTPUT INTEGER - CODED PRESENT WEATHER.
C ISKY - OUTPUT INTEGER - AMOUNT OF SKY COVER IN OKTAS
C
C EXTERNALS
C IUADEC - UTIL      IFIX - FORT      MASK
C
C DIMENSION IUP(512)
C ILX = -99
C IV = -99
C ISKY = -99
C
C SEARCH FOR VISIBILITY
C
C L = IGNH + 4 ;POINT TO VSBY
C IV = IUADEC(IUP,L) ;DECODE VISIBILITY
C IF(IV .LT. 98 .OR. IV .GT. 99) IV = -99 ;VSBY OUT OF BOUNDS
C IF(IV .EQ. -99) GOTO 1800
C IV = IV - 89
C GOTO(98,91,92,93,94,95,96,97,98,99),IV
90 IV = 8 ; 8 < 1/16 MILE
C GOTO 1800
91 IV = 1 ; 1/16 < 1/8
C GOTO 1800
92 IV = 2 ; 1/8 < 1/4
C GOTO 1800
93 IV = 4 ; 1/4 < 1/2
C GOTO 1800
94 IV = 8 ; 1/2 < 1
C GOTO 1800
95 IV = 16 ; 1 < 2 MILES
C GOTO 1800
96 IV = 32 ; 2 < 5
C GOTO 1800
97 IV = 80 ; 5 < 11 MILES
C GOTO 1800
98 IV = 176 ; 11 < 26 MILES
C GOTO 1800
99 IV = 432 ; > 26 MILES
C
C SEARCH FOR SKY COVER - IGNH MUST BE POINTING TO THE SPACE
C BEFORE THE VISIBILITY GROUP.
C
1800 ISKY = IUP(IGNH + 7) - 48 ;JUMP TO SKY CODE AND
C ;DECODE IT
C IF(ISKY .LT. 0 .OR. ISKY .GT. 9) ISKY = -99 ;INVALID SKY CODE
C
C SEARCH FOR PRESENT WEATHER
C
C L = MASK('8>78989<857>9<857>9',5,IUP,IGNH+8,ISTOP);FIND PRESENT WX GRP
C IF(L .EQ. -1) GOTO 9999 ;SEARCH FAILED
C
C ILX = (IUP(L+1) - 48)*10 + (IUP(L+2) - 48) ;DECODE WEATHER
C IF(ILX .LT. 0 .OR. ILX .GT. 99) ILX = -99 ;CHECK ILX VALUE
9999 RETURN
END

```

```

SUBROUTINE LLL(IBC,IBS,LAKE,LAT,LON)
COMMON /DISLOC/ SOLAT(25), SOLON(25), IBB(5), IEE(5),
+ISOUR(9,25), ICPTS(2,37)
DATA SOLAT /47.8,47.1,47.2,47.4,47.4,48.6,46.8,47.9,45.8,45.8,
+ 44.1,43.8,43.8,41.6,44.8,46.8,45.8,45.3,43.9,43.1,41.9,41.9,
+ 42.5,43.4,43.9/
DATA SOLON /91.7,90.4,88.8,87.6,85.8,87.8,85.8,89.3,85.8,86.1,
+ 86.5,86.2,87.9,87.3,83.9,83.3,81.7,82.7,82.4,82.5,88.8,
+ 79.2,78.5,76.7/
DATA IBB /1.9,16,21,24/ IEE /8,15,20,23,25/
DATA ISOUR/'TLD HARBOURS RB OUTER ISLAND LT ',
+'KEWEENAW RB MANITOU ISLAND CARIBOU ISLAND LT ',
+'SLATE ISLAND WHITEFISH POINT ROCK OF AGES ',
+'WHITE SHOAL LIGHT SOUTH MANITOU LT BIG SABLE LIGHT ',
+'GRAND HAVEN RB MILWAUKEE RB GARY BREAKWATER LT ',
+'STURGEON BAY RB DETOUR REEF LIGHT THUNDER BAY IS LT ',
+'COVE ISLAND LIGHT HARBOR BEACH RB SRN LK MURON BUOY ',
+'SOUTHEAST SHOAL LTASHTABULA RB DUNKIRK LIGHT ',
+'THIRTY MILE PT LT MAIN DUCK ISLAND '
DATA ICPTS /' N NNE NNE NE NE ENE ENE E E E ESE ESE',
+' SE SE SSE SSE S S S SSU SSU SU SU USW USW U U ',
+' W WNW WNW NW NW NNW NNW N N N '

```

REV. 1.00

MAY, 1985
FORTRAN IV/ REV 5.20

PEROUTKA/RANDEL
DG ECLIPSE (S238)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.10

PRINTS THE LATITUDE, LONGITUDE, AND LOCATION FOR A SHIP OR BUOY REPORT.

ARGUMENT LIST

IBC I RDOS CHANNEL TO WHICH TEXT WILL BE WRITTEN
 IBS I DETERMINES TYPE OF REPORT.
 1 FOR BUOY
 2 FOR SHIP
 LAKE I (SHIPS AND BUOYS ONLY) LAKE WHERE REPORT ORIGINATED. (VALUES
 DEFINED IN LAKLOC.FR)
 LAT I (SHIPS AND BUOYS ONLY) LATITUDE IN TENTHS OF A DEGREE NORTH.
 LON I (SHIPS AND BUOYS ONLY) LONGITUDE IN TENTHS OF A DEGREE WEST.

EXTERNALS

LRS (UTIL) PRNUM SHPLOC FLOAT (FORT) IFIX (FORT)

CHANNELS/FILES

IBC IS THE RDOS CHANNEL TO WHICH THE DATA IS WRITTEN.

VARIABLES

SOLAT() LATITUDES FOR THE VARIOUS SOURCE POINTS
 SOLON() LONGITUDES FOR THE VARIOUS SOURCE POINTS
 IBB() USED TO ASSOCIATE A SOURCE POINT WITH A GIVEN LAKE
 IEE() USED TO ASSOCIATE A SOURCE POINT WITH A GIVEN LAKE
 ISOUR() NAMES OF THE VARIOUS SOURCE POINTS
 ICPTS() USED TO CONVERT DEGREES OF THE COMPASS TO 16-POINT COMPASS
 READINGS.
 HDIS DISTANCE OF A SHIP FROM THE CLOSEST SOURCE POINT
 IDEG AZIMUTH OF A SHIP FROM THE CLOSEST SOURCE POINT
 J NUMBER DESIGNATING CLOSEST SOURCE POINT

```

GOTO (100,200) IBS
100 N = 20
CALL LRS(IBC,'BUOY
GOTO 300
200 N = 4
CALL LRS(IBC,'LAT ',N,IER)
CALL PRNUM(IBC,LAT/10,-1,1,0)
N = 1
CALL LRS(IBC,' ',N,IER)
CALL PRNUM(IBC,LAT,1,2,0)
N = 5
CALL LRS(IBC,' LON ',N,IER)
CALL PRNUM(IBC,LON/10,-1,1,0)
N = 1
CALL LRS(IBC,' ',N,IER)
CALL PRNUM(IBC,LON,1,2,0)
N = 3
CALL LRS(IBC,' ',N,IER)
300 CALL SHPLOC(LAKE,FLOAT(LAT)/10.,FLOAT(LON)/10.,HDIS,IDEG,J)
CALL PRNUM(IBC,IFIX(HDIS+.5),10,1,0);NOW WRITE LOCATION.
N = 4
CALL LRS(IBC,ICPTS(1,IDEG),N,IER)
N = 1
CALL LRS(IBC,' ',N,IER)
N = 10
CALL LRS(IBC,ISOUR(1,J),N,IER)
N = 3
CALL LRS(IBC,' ',N,IER)
RETURN
END

```



```

SUBROUTINE WMOHD(ICLIN, IDAY, IHR, IBC)
  DIMENSION ICLIN(41), LINE(6)

```

```

REV. 1.00

```

```

MAY, 1985

```

```

FORTRAN IV/ REV 5.20

```

```

• PEROUTKA/RANDEL
  DG ECLIPSE (S230)

```

```

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

```

```

WRITES A WMO HEADER TO RDOS CHANNEL IBC. THE HEADING CONSISTS OF THE
11 BYTES BEGINNING AT ICLIN(2) AND A SIX-DIGIT DATE-TIME GROUP COMPUTED
FROM IDAY AND IHR.

```

ARGUMENT LIST

```

ICLIN  I  COMMAND LINE FROM COMMAND FILE
IDAY   I  VALID DATE OF RUN
IHR    I  VALID HOUR OF RUN
IBC    I  RDOS CHANNEL ON WHICH TO WRITE HEADER

```

EXTERNALS

```

WMOVE (UTIL)  IAND (FORT)  IOR (FORT)  WRS (UTIL)  PRNUM

```

CHANNELS/FILES

```

IBC IS THE RDOS CHANNEL NUMBER TO WHICH THE HEADER IS WRITTEN

```

VARIABLES

```

LINE().  HOLDS THE FIRST PART OF THE HEADER TEMPORARILY

```

```

CALL WMOVE(ICLIN(2),6,LINE)      ;GET MOST OF HEADER FROM ICLIN
LINE(6) = IAND(LINE(6),177400K)  ;ADD A SPACE
LINE(6) = IOR(LINE(6),'<0> ')
N = 12
CALL WRS(IBC,LINE,N,IHR)
CALL PRNUM(IBC,IDAY,10,2,0)      ;DATE-TIME GROUP
CALL PRNUM(IBC,IHR*100,1000,2,3)
RETURN
END

```

SUBROUTINE WTEMP(IBGN, ISTOP, IUP, IWT)

MARCH 1985
REV 1.00

RANDEL/PEROUTKA

WSFO CLE/FTS 942-4949

PURPOSE

DECODES WATER TEMP. FROM SYNOPTIC OBSERVATIONS. SEARCH THE ARRAY IUP FOR 0**** GROUP, WHICH INDICATES TEMPERATURES ARE IN TENTHS OF DEGREES, FROM IBGN TO ISTOP. IF FIRST SEARCH FAILS LOOK FOR 0****/ GROUP, INDICATING TEMPERATURE IN WHOLE DEGREES. RETRUNS THE WATER TEMPERATURE IN WHOLE DEGREES CELCIUS. IBGN SHOULD POINT TO THE MARINE DATA GROUP 222**

ARGUMENT LIST

IUP() - INPUT ARRAY - WITH UNPACKED OBSERVATION.
IBGN - INPUT INTEGER - POINTER TO ARRAY IUP WHERE
TO BEGIN SEARCH FOR WATER TEMP. SHOULD BE
MARINE DATA GROUP 222**
ISTOP - INPUT INTEGER - POINTER TO ARRAY IUP WHERE
TO STOP SEARCH FOR TEMP.
IWT - OUTPUT INTEGER - WATER TEMPERATURE IN WHOLE
DEGREES FARENHEIT.

EXTERNALS

IUADEC - UTIL IFIX - FORT MASK

DIMENSION IUP(512)

IWT = -99

L = MASK('<0>001090909',5,IUP,IBGN,ISTOP) ;TEMP GRP IN TENTHS
IF (L .LE. IBGN) GOTO 100 ;SEARCH FAILED

J = L + 1

TEMP = IUADEC(IUP,J)

;DECODE TEMP

IF(TEMP .GT. 1000.) TEMP = TEMP*(-1.) + 1000. ;TEMP IS NEGATIVE
TEMP = TEMP/10. ;CONVERT TO TENTHS

GOTO 200

100 L = MASK('<0>0010909<0>/',5,IUP,IBGN,ISTOP) ;TEMP IN WHOLE DEG.
IF (L .LE. IBGN) GOTO 9999 ;SEARCH FAILED

J = L + 1

TEMP = IUADEC(IUP,J)

;DECODE TEMP

IF(TEMP .GT. 100.) TEMP = TEMP*(-1.) + 100. ;TEMP IS NEGATIVE

200 TEMP = (TEMP * 9.)/5. + 32. ;CONVERTING TO FARENHEIT

IF(TEMP .LT. 0.) TEMP = TEMP - .5 ;ROUND OFF ERROR

IF(TEMP .GT. 0.) TEMP = TEMP + .5 ;ROUND OFF ERROR

IWT = IFIX(TEMP)

;CONVER TO INTEGER

9999 RETURN

END

SUBROUTINE WWSBY(IBC,IVSBY)

REV. 1.00

MAY, 1985

FORTRAN IV/ REV 5.20

PEROUTKA/RANDEL
DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

WRITES VISIBILITY TO AN RDOS CHANNEL AS A WHOLE NUMBER AND/OR FRACTION.

ARGUMENT LIST

IBC I RDOS CHANNEL TO WHICH TEXT WILL BE WRITTEN

IVSBY I VISIBILITY (IN SIXTEENTHS OF A MILE) TO BE PRINTED

EXTERNALS

WRS (UTIL) MOD (FORT) PRNUM

CHANNELS/FILES

IBC IS THE RDOS CHANNEL TO WHICH THE VISIBILITY IS WRITTEN.

VARIABLES

N BYTE COUNTER USED FOR WRS

NWHOL WHOLE PORTION OF VISIBILITY TO BE PRINTED

NFRAC FRACTIONAL PORTION OF VISIBILITY TO BE PRINTED

I LOOP VARIABLE WHILE SEARCHING FOR A DENOMINATOR

IDEN TRIAL DENOMINATOR

J TEMPORARY USED IN CHOOSING DENOMINATOR

NUM NUMERATOR

IF (IVSBY.NE.0) GOTO 100

;ZERO IS A SPECIAL CASE

N = 1

CALL WRS(IBC,"0",N,IER)

GOTO 900

100 NWHOL = IVSBY/16

;SPLIT IVSBY INTO WHOLE AND

NFRAC = MOD(IVSBY,16)

;FRACTIONAL PARTS.

IF (NWHOL.NE.0) CALL PRNUM(IBC,NWHOL,-1,0,0);PRINT WHOLE.

IF (NFRAC.EQ.0) GOTO 900

;PRINT FRACTIONAL.

DO 200 I = 1, 4

IDEN = 2**I

;POTENTIAL DENOMINATOR

J = 16/IDEN

IF (MOD(NFRAC,J).NE.0) GOTO 200

NUM = NFRAC/J

CALL PRNUM(IBC,NUM,10,0,0)

;PRINT VISIBILITY

N = 1

CALL WRS(IBC,"/",N,IER)

CALL PRNUM(IBC,IDEN,10,0,0)

GOTO 900

200 CONTINUE

900 RETURN

END

```

SUBROUTINE WUX(IBC,IWX)
COMMON /WBLK/ IWC(12), IWTEXT(6,13), NWC(12)
DATA IWC /'T ZRZLR S RSL F D Y H K ' /
DATA IWTEXT /'THUNDERSTORMFREEZG RAIN FREEZG DRZL RAIN
+ *SNOW      RAIN/SNOW  DRIZZLE  FOG      DUST
+ *BLOWG SPRAY HAZE    SMOKE    MISSING  ' /

```

REV. 1.00

MAY, 1985
FORTRAN IV/ REV 5.20

PEROUTKA/RANDEL
DG ECLIPSE (S230)

WSFO CLE FTS 942-4949
RDOS/ REV. 6.18

WRITES WEATHER TEXT TO AN RDOS CHANNEL GIVEN ONE OF THE CODES IN IWC.

ARGUMENT LIST

IBC I RDOS CHANNEL TO WHICH TEXT WILL BE WRITTEN
IWC I WEATHER CODE IN ASCII. TWO BYTES, SPACE- OR NULL-PADDED.
 GENERALLY, ONE OF THE CODES IN IWC.

EXTERNALS

IAND (FORT) IOR (FORT) WRS (UTIL)

CHANNELS/FILES

IBC IS THE RDOS CHANNEL TO WHICH THE TEXT IS WRITTEN.

VARIABLES

IW HOLD WEATHER CODE
I LOOP COUNTER WHEN CHECKING CODE AGAINST LIST.
IWC() LIST OF WEATHER CODES
N BYTE COUNTER USED FOR WRS
IWTEXT() TEXT WHICH CORRESPONDS TO CODES IN IWX

```

      IW = IWX
      IF (IAND(IW,377K).EQ.0) IW = IOR(IW,'<0> ') ;USE SPACE PAD.
      DO 100 I = 1, 12                           ;COMPARE CODE LIST TO CODE
      J = I                                       ;RECEIVED.
      IF (IW.EQ.IWC(I)) GOTO 200
100 CONTINUE
      N = 12                                     ;NO MATCH.
      CALL WRS(IBC,IWTEXT(1,13),N,IER)
      RETURN
200 N = 12
      CALL WRS(IBC,IWTEXT(1,J),N,IER)           ;PRINT WEATHER.
      RETURN
      END

```


LAWEB: DATA PROCESSING FOR THE GREAT LAKES

PART A: INFORMATION AND INSTALLATION

PROGRAM NAME: LAWEB

AAL ID:

REVISION NO.: 2.00

PURPOSE: Reads marine data from a large number of database products in a wide variety of formats. Produces two types of database products. First, a broadcast text which is read over the Great Lakes VHF Radio System, and, second, a plot file which is compatible with the PMOD system.

PROGRAM INFORMATION:

Development Programmer(s):

William C. Randel

Matthew R. Peroutka

Location: WSFO Cleveland, OH

Phone: (FTS) 942-4949

Language: FORTRAN/Rev. 5.57

Save File Creation Date: February 28, 1986

Running Time: about four minutes for all available data

Disk Space:

Program Files 113 RDOS blocks

Data Files 20 RDOS blocks

Maintenance Programmer(s):

William C. Randel

Matthew R. Peroutka

Location: WSFO Cleveland, OH

Phone: (FTS) 942-4949

Type: Standard

PROGRAM REQUIREMENTS

Program Files:

Name

LAWEB.SV

Comments

Main program

Data Files:

Name

(various)

DP Location

DPØ

Read/Write

Read

Comments

Command File

Name specified in command
line

cccnxxx

DPØ

Write

Temporary output for database
products
Names specified in command
file

AFOS Products:

<u>ID</u>	<u>Action</u>	<u>Comments</u>
(various)	Input and output products are specified by Command File.	

LOAD LINE

```
RLDR LAWEB GCOMF GTIME JDATE IGCOD IOPNB CLSB IASCL IOPNP CLSP^
WMOHD PRNUM HDRB PLTLB PRB WLL SHPLOC WWSBY WWX PCMAN CANDEC^
GETP OBTIM SWIND ATEMP PKWND PRES MASK GPIX BNSCH IWX2N N2IWX^
PRP PCGR CGRDEC CGRSR BACKUP GBLK0 ICGSKY ICGWX ICGVS ICGWN ICGWV^
ICGT ICGPR POMR OMRDEC LATLON SWX WTEMP BOUYWV PBOY LAKLOC DSHP^
GETPS PSHP PPRES ALTDEC PSYN PSA SADC SCNWX ADXX VSBYWX TEMPDP^
DRSP DECVIS BKLCHK WIND INTGR SKY STRING INSTALL IFAH ICEL WX^
BFILL <AG TOP BG UTIL FORT>.LB
```

PROGRAM INSTALLATION

1. Move LAWEB.SV to APPL1 on DZ0 with a link to DP0.
2. Create a command file using the codes in Figure 6.
3. Most sites create a number of command files for various purposes and run the program through macros. This allows one command to run the LAWEB program followed by various combinations of PMOD, HCOPI and GENUTF.

LAWEB: DATA PROCESSING FOR THE GREAT LAKES

PART B: EXECUTION AND ERROR CONDITIONS

PROGRAM NAME: LAWEB

AAL ID:
REVISION NO.: 1.00

PROGRAM EXECUTION:

1. RUN:LAWEB (COMMANDFILE) where COMMANDFILE is the name of the appropriate Command File.

ERROR CONDITIONS

Messages from ADM

Meaning

1. JOB LAWEB COMPLETE PRODUCT CCCNNNXXX STORED

Dasher Messages

Meaning

- | | |
|----------------------------|----------------------------------|
| 1. CAN'T OPEN COMMAND FILE | Named file is not available |
| 2. CAN'T OPEN CCCNNNXXX. | System error opening named file. |

Eastern Region Computer Programs and Problems (Continued)

- 19 Verification of Asynchronous Transmissions. Lawrence Cedrone, March 1984. (PB84 189885)
- 20 AFOS Hurricane Plotter. Charles Little, May 1984. (PB84 199629)
- 21 WARN - A Warning Formatter. Gerald G. Rigdon, June 1984. (PB84 204551)
- 22 Plotting TDL Coastal Wind Forecasts, Paula Severe, June 1984 (Revised) (PB84-220789)
- 23 Severe Weather Statistics STADTS Decoder (SWX) and Plotter (SWY), Hugh M. Stone, June 1984. (PB84-213693)
- 24 WXR, Harold Opitz, August 1984. (PB84-23722) (Revised August 1985, PB86 100815/AS)
- 25 FTASUM: Aviation Forecast Summaries, Matthew Peroutka, August 1984. (PB85-112977)
- 26 SAOSUM: A Short Summary of Observations. Matthew Pernutka, October 1984. (PB85-120384)
- 27 TRAJ - Single Station Trajectory Plot, Tom Nizioł, December 1984. (PB85-135002)
- 28 VIDTEX, Gerald G. Rigdon, February 1985 (PB85-175669/AS)
- 29 ISENTROPIC PLOTTER, Charles D. Little, February 1985 (PB85-175651/AS)
- 30 CERR: An Aviation Verification Program, M. Peroutka, April 1985. (PB85-204824/AS)
- 31 Correlation and Regression Equation Program - REGRS, H. Stone, May 1985. (PB85-213353/AS)
- 32 Scatter Diagram and Histogram Program - SCATR, H. Stone, May 1985. (PB85-213346/AS)
- 33 TIMCHEK, Gerald G. Rigdon, June 1985. (PB85-221257/AS)
- 34 A MOS Temperature - PoP Forecast Plot, William C. Randel, October 1985 (PB86-120029/AS).
- 35 ROTODRAW, Thomas Nizioł, November 1985 (PB86-131828/AS)

NOAA SCIENTIFIC AND TECHNICAL PUBLICATIONS

The National Oceanic and Atmospheric Administration was established as part of the Department of Commerce on October 3, 1970. The mission responsibilities of NOAA are to assess the socioeconomic impact of natural and technological changes in the environment and to monitor and predict the state of the solid Earth, the oceans and their living resources, the atmosphere, and the space environment of the Earth.

The major components of NOAA regularly produce various types of scientific and technical information in the following kinds of publications:

PROFESSIONAL PAPERS — Important definitive research results, major techniques, and special investigations.

CONTRACT AND GRANT REPORTS — Reports prepared by contractors or grantees under NOAA sponsorship.

ATLAS — Presentation of analyzed data generally in the form of maps showing distribution of rainfall, chemical and physical conditions of oceans and atmosphere, distribution of fishes and marine mammals, ionospheric conditions, etc.

TECHNICAL SERVICE PUBLICATIONS — Reports containing data, observations, instructions, etc. A partial listing includes data serials; prediction and outlook periodicals; technical manuals, training papers, planning reports, and information serials; and miscellaneous technical publications.

TECHNICAL REPORTS — Journal quality with extensive details, mathematical developments, or data listings.

TECHNICAL MEMORANDUMS — Reports of preliminary, partial, or negative research or technology results, interim instructions, and the like.



Information on availability of NOAA publications can be obtained from:

**ENVIRONMENTAL SCIENCE INFORMATION CENTER (D822)
ENVIRONMENTAL DATA AND INFORMATION SERVICE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE**

**6009 Executive Boulevard
Rockville, MD 20852**

