WSRC-RP--93-778

DE93 010381

WIND SYSTEM DOCUMENTATION (U)

by J. R. Froggatt

Westinghouse Savannah River Company
Savannah River Site
Aiken, South Carolina 29808

Other Authors:

C. P. Tatum
(WSRC)

DISCLAIMER

WIND

SYSTEM

DOCUMENTATION (U)

Authors: J.R. Froggatt

C.P. Tatum

Date: 15 January 1993

CHAPTER 1

** DATA COLLECTION PROGRAMS **

1.1 ARCIV1 (Collect Data Every 1 1/2 Sec. And Summarize Every 15 Minutes)

IDENTIFICATION:

Archiving System TV and SR Tower data collection program

NAME: ARCIV1

TYPE: WIND System data collection program

FUNCTION:

This program executes on both WIND1:: and WIND2::. On WIND2:: it runs in a primary data collection mode and asserts this by obtaining an exclusive lock on resource ARCIV1. On WIND1:: the program attempts to obtain an exclusive lock on resource ARCIV1. If successful (only possible when WIND2:: version is not operating) program executes but checks to see if the WIND2:: version is requesting the lock after every data gathering pass (about every 1.5 seconds). Any time the WIND2:: version requests the lock it is granted and data collection ceases on WIND1::.

When operating in data collection mode, this program reads TV tower, Central Climatology (CL) tower, and the SRP area tower data and accumulates it in global commons ARCOM and ARCOM_CL. New data readings are collected every 1.5 seconds, converted to engineering units, azimuth corrected for winds fluctuating between 0 and 360 degrees, and accumulated for the current 15 minute interval.

At the end of the 15 minute period, maximum values, mean values, and standard deviations are computed for wind speeds, azimuths, elevations and temperatures for the TV, SRP, and CL towers. Also, the CL tower measures a number of additional meterological values. U and V vector components are also accrued during the 15 minute period. This allows the option of later retrieving either vector or scalar derived wind speeds and azimuths. After the above values have been calculated at the end of the 15 minute period, process ARCIV2 is invoked to quality assure the data and write it to disk. Both versions of ARCIV2 (WIND1:: and WIND2::) write data to the same files (located in WD: on SHADOW1:). ARCIV2 is only activated on the system with possession of the exclusive lock on ARCIV1.

INITIATION:

Program is requested for execution in the startup file. Also, program is stopped and restarted manually when a calibration coefficient has been changed by program NEWFIL, NEWFL8, or NEWFL_CL. This is done to allow the new coefficients to be read into global common ARCOM and ARCOM_CL as the program begins execution.

REAL TIME CONSIDERATIONS:

Once program has been initiated, it runs continuously. at a high priority.

AUTHOR: J.R. Froggatt, Process Systems Group, ESD, E. I. DuPont

DATE: 11/4/85
LAST REVISED: 9/15/88

1.2 ARCINI (ARCIV1 SUBR - Program Startup Initialization)

IDENTIFICATION:

ARCIV1 STARTUP INITIALIZATION SUBROUTINE

NAME: ARCINI

TYPE: ARCIV1 SUBROUTINE

FUNCTION:

This subroutine first determines which system (WIND1:: or WIND2::) that the program is operating on. This will allow the Archiving programs to operate properly depending upon whether the other system is in operation or not. The subroutine then initiates the execution of the individual subprocesses that will read from the individual Micromac units and place the data into auxiliary Common ARCOM1. In addition, the subroutine reads and enters into Common ARCOM the coefficients to be used in calculating engineering units from raw voltages. The TV data is read from WD:ARCOEF.DAT and the SR tower data is read from WD:SRCOEF.DAT. The subroutine also reads the quality control flags written during the last execution oF ARCIV2 from WD:ARCQTY.DAT and places them into ARCOM.

CALLING SEQUENCE: Called by ARCIV1 one time when program begins execution.
AUTHOR: J.R. Froggatt, Process Systems Group, ESD, E.I. DuPont

DATE: 8/10/83
LAST REVISED: 07/08/88

1.3 ARCIBF (ARCIV1 SUBR - Initialization Prior To New 15 Min Data Collection)

IDENTIFICATION:

Subroutine to init common ARCOM, ARCOM_TP, and ARCOM_CL an prior to 15 min data accrual

NAME: ARCIBF

TYPE: ARCIV1 subroutine

FUNCTION:

This subroutine initializes the portions of global commons ARCOM, ARCOM_TP, and ARCOM_CL used for the accumulation of current 15 minute data. All variables initialized are zeroed out with the exception of those used to store 15 minute maximum values. These variables are initialized with artificially low values. All variables handled are contained in global commons ARCOM, ARCOM_TP, ARCOM_CL.

CALLING SEQUENCE:

Subroutine called by ARCIV1 every 15 minutes prior to storing new data for the 15 minute period just begun.

AUTHOR: J.R. Froggatt, On Line Systems Group, ESD, E.I. DuPont

DATE: 11/4/85
LAST MODIFIED: 9/9/88

1.4 ARCNB (ARCIV1 SUBR - Acquire New Data From The Various Met Towers)

IDENTIFICATION:

Subroutine to read new bursts of TV, Climatology, and Area Tower data.

NAME: ARCNB

TYPE: ARCIV1 Subroutine

FUNCTION:

This subroutine reads new bursts of data from the TV, CL, A, C, D, F, H, K, L, and P towers. All data in received from Micro Mac units located at the different towers. Data is in the form of ASCII characters representing voltage signals received from the various meterological instruments. Data is received and processed by ARCIV1 every 1.5 seconds.

CALLING SEQUENCE: Called by ARCIV1 each time pgm ready for data.

REAL TIME CONSIDERATIONS:

Data from the various towers must be received before a 1.49 second timeout or it is ignored for the current pass. The 1.49 second timeout is the time allowed for gathering data since the VAX 8550 requires about 0.01 seconds to process the data each pass and data is required every 1.5 seconds.

AUTHOR: J.R. Froggatt, Process Systems Group, ESD, E.I. DuPont

DATE: 7/31/80
REVISED: 01/10/89

1.5 ARCENU (ARCIV1 SUBR - Calculation Of Engineering Units)

IDENTIFICATION:

Subroutine to convert TV, SR(includes TP), and CL tower voltages to eng units

NAME: ARCENU

TYPE: ARCIV1 subroutine

FUNCTION:

This subroutine converts from raw voltages to engineering units. The equations use emperical coefficients developed while calibrating the field instruments. Any measurement for which the voltage is out of range and any engineering unit value which is obviously incorrect is flagged as an error and is not used any further.

CALLING SEQUENCE:

Called by ARCIV1 after obtaining raw voltages from subroutine ARCNB.

INITIALIZATION:

The calibration coefficients used by the equations in this subroutine are contained in files WD:ARCOEF.DAT, WD:SEVCOE.DAT, WD:COEF_TP, and WD:COEF_CL.DAT for the TV,

SR (plus TP), AND CL towers respectivly. These coefficients are read into global commons ARCOM ARCOM_TP, and ARCOM_CL by subroutine ARCINI when ARCIV1 begins execution. If any of these files is modified, it is necessessary to abort and reactivate ARCIV1 to place the revised data into ARCOM, ARCOM_TP, and ARCOM_CL.

AUTHOR: R.A. Mueller ENVIRONMENTAL TRANSPORT DIVISION
J.R. Froggatt, Process Systems Group, ESD, E.I. DuPont

DATE: 9/26/80
REVISED: 11/06/85
REVISED: 10/20/85
REVISED: 1/22/88

1.6 ARCAZC (ARCIV1 SUBR - Kern/Pendergast Azimuth Correction)

IDENTIFICATION:

Kern/Pendergast Azimuth correction algorithm subroutine

NAME: ARCAZC

TYPE: ARCIV1 SUBROUTINE

FUNCTION:

This subroutine is designed to eliminate the problem encountered when wind directions fluctuate between values near 0 degrees and values near 360 degrees. The basic strategy involves maintaining an average value for each measured azimuth for the current 15 minute interval and comparing it with the currrent reading. Prior to any comparisons, values are biased upward by 720 degrees to avoid any problems that would result from negative numbers. Negative numbers are a possibility even though azimuth values supplied to this subroutine are between 0 - 360 degrees because the algorithm used sometimes subtracts 360 degrees from the original reading.

Three comparisons are made involving the current 15 minute average and the current reading.
They are:

1) Average vs Current
2) Average vs (Current + 360)
3) Average vs (Current - 360)

If 1) results in the smallest difference no further adjustment is made beyond the original biasing of 720.

If 2) results in the smallest difference, 360 is added to the original value and if 3) results in the smallest difference, 360 is subtracted from the original value. These values are subsequently accumulated in sum x and sum x**2 values by subroutine ARCADD. The mean and standard deviation are later calculated by subroutine ARCUPD at the end of the 15 minute period and the bias of 720 is backed out from the mean value.

The algorithm basically holds fluctuations between 0 and 360 degrees to the vicinity of either 0 (720) or 360 (1080) depending upon where the average happens to be. This eliminates the problem of distorted mean values and standard deviations.

Mean values are also computed using U and V vector components and these values are most often used by WIND system data retrieval programs. However, the standard deviations computed using the sum x and sum x**2 values resulting from this subroutine are the ones used by the data retrieval programs.

CALLING SEQUENCE:

Called by ARCIV1 after ARCIV1 calls ARCENU

INITIALIZATION:

At beginning of a new 15 minute period, new average azimuth values are begun.

AUTHOR: J.R. Froggatt, Process Systems Group, ESD, E.I. DuPont

DATE: 8/1/80
REVISED: 11/06/85
REVISED: 1/22/88

1.7 ARCADD (ARCIV1 SUBR - Update Of Running Totals With Latest Readings)

IDENTIFICATION:

Subroutine to add latest values to accumulating 15 min values

NAME: ARCADD

TYPE: ARCIV1 SUBROUTINE

FUNCTION:

This subroutine adds the latest TV, SR, and CL tower data into accumulating variables located in global commons ARCOM, ARCOM_TP,

and ARCOM_TP. The number of good readings is incremented (if appropriate), and the value is added to sum X and the square of the value is added to sum X**2 for each good reading. In addition, the current maximum for each variable is collected by comparing current values with previous maximums.

This subroutine also accrues totals for U and V vector components. This allows a more accurate determination of wind speeds and azimuths. However, the scalar values for speed and direction are also maintained.
the formulas used are:

U = - (SPEED * COS(AZM))
V = - (SPEED * SIN(AZM))

CALLING SEQUENCE: Called by ARCIV1 after ARCAZC.

INITIALIZATION:

Accumulating variables in global commons ARCOM and ARCOM_CL are zeroed out at beginning of 15 minute period by subroutine ARCIBF.

AUTHOR: J.R. Froggatt, Process Systems group, ESD, E.I. DuPont

DATE: 9/26/80
REVISED: 11/06/85
REVISED: 10/20/87
REVISED: 1/22/88

1.8 ARCUBF (ARCIV1 SUBR - End Of 15 Minute Data Collection Period Processing)

IDENTIFICATION:

Subroutine to update 15 min buffer prior to writing to disk

NAME: ARCUBF

TYPE: ARCIV1 subroutine

FUNCTION:

This subroutine transfers data collected during the 15 minute period just ended to a holding area. Means and standard deviations are computed from the SUM X and SUM X**2 values accumulated during the period.
In addition, some preliminary quality assurance is performed on the mean values computed and the appropriate quality assurance flags are set.

CALLING SEQUENCE: Called by ARCIV1 at end of 15 minute period

REAL TIME CONSIDERATIONS:

Runs just prior to ARCIV1 request for execution of program ARCIV2

AUTHOR: J.R. Froggatt, Process Systems Group, ESD, E.I. DuPont

DATE: 11/4/85

1.9 TMPPUT (ARCIV1 SUBR - Place Latest 15 Minute Data On Disk)

IDENTIFICATION:

Subroutine to update 15 min buffer prior to writing to disk

NAME: ARCUBF

TYPE: ARCIV1 subroutine

FUNCTION:

This subroutine transfers data collected during the 15 minute period just ended to a holding area. Means and standard deviations are computed from the SUM X and SUM X**2 values accumulated during the period. In addition, some preliminary quality assurance is performed on the mean values computed and the appropriate quality assurance flags are set.

CALLING SEQUENCE: Called by ARCIV1 at end of 15 minute period

REAL TIME CONSIDERATIONS:

Runs just prior to ARCIV1 request for execution of program ARCIV2

AUTHOR: J.R. Froggatt, Process Systems Group, ESD, E.I. DuPont

DATE: 11/4/85

1.10 ARCRD (ARCIV1 SUB PROCESS PGM - Read Met Twr UMAC Data)

IDENTIFICATION:

Pgm to read data from a specific met twr depending upon subprocess name

NAME: ARCRD

TYPE: ARCIV1 Subprocess

FUNCTION:

This program inputs new data from either the TV tower or one of the SR towers depending on the name of the subprocess the program is running under. Currently, the TV tower, CL tower and A, C, D, F, H, K, L, and P Area towers have individual Micro-Mac units supplying data. The program is signaled from detached process ARCIV1 when data is desired using an event flag. This program signals ARCIV1 that data is ready also thru the use of an event flag.

CALLING SEQUENCE: Signaled by detached process ARCIV1 each time it requires new data.

REAL TIME CONSIDERATIONS:

While this routine is waiting for I/O, other pgms in the system can execute. ARCIV1 continuously loops around processing the latest raw data supplied by this program in it's current subprocess as well as other copies of this program executing in subprocesses devoted to other Area towers.

AUTHOR: J.R. Froggatt, On Line Systems Group, ESD, E.I. DuPont

DATE: 11/85
LAST MODIFIED: 10/87

1.11 ARCIV2 (Quality Assure 15 Minute Average Data And Write To Disk)

IDENTIFICATION:

Quality Assurance Program

NAME: ARCIV2

INSTALLED NAMES: ARCIV2

TYPE: WIND System Data Collection Program

FUNCTION:

This program is called at the end of every 15 minute period by data collection program ARCIV1. The program's purpose is to QUALITY Assure the data and then write it to disk using subroutine ARCPUT. Erroneous data is replaced with appropriate estimates. The Quality Assurance Algorithm implemented in the WIND System operates in two phases. The first phase consists of three schemes for determining invalid data, while the second phase is concerned with replacing the invalid data with estimated values.

Data is flagged as being invalid if an instrument is known to be out of service (done manually using pgm flags). In addition, if less than 10% of the expected number of readings for a 15 minute period for a particular instrument are within the known ranges of the instrumentation, the data for that period and instrument is flagged (performed by ARCIV1 subroutine ARCUBF). The data that pass this initial screening is further checked against the average of all similar data items (e.g., all SRS Tower Azimuths). Any observation that differs by more than a specific amount from the mean is flagged, and a new mean is calculated from the remaining points. This procedure continues until an acceptable mean is computed that involves at least two instruments, or all of the observations are flagged.

The allowed deviation from the mean is a specified number of estimated standard deviations of the variable in question. The estimated standard deviations are calculated with empirical relationships based on meteorological conditions and statistical paramters, such as Atmosperic Stability and Sampling Period.
An additional consideration is required in Quality Controlling the data collected on the WJBF-TV Tower. Here the individual data points are compared to the mean values of like data items at different tower heights. In order to make meaningful comparisons, all data are adjusted to a common reference height. Speed, Siga, and Sige are adjusted using a power law relationship while Temperature and Wind Direction are adjusted using a linear relationship. In all cases, equation parameters are used that have been empirically developed as a function of Atmospheric Stability.

The second phase of the Quality Assurance Algorithm involves replacement of bad data with estimated values. For the SRS Towers, this merely consists of substituting the mean of the accepted values for the rejected data. If all values of a particular type have been rejected, the last available Average SRS Tower is used. However, if this is more than two hours old, adjusted data from Bush Field in nearby Augusta, Georgia are used. If Bush Field data is more than two hours old, no Quality Assurance is performed and a message to that effect is displayed.

TV Tower data is replaced by fitting a curve to the accepted values, and substituting the value derived from the curve for the rejected value. If too few points exist to generate a reasonable curve, the average value calculated for the reference height is adjusted to the level in question. If all TV readings for a particular type of variable are missing, the Height-Adjusted SRS Tower Average is used.

After the new values have been inserted the program calls Subroutine ARCPUT to write the 15 minute data to disk. Both the original and the corrected set of data is written to disk by ARCPUT.

CALLING SEQUENCE:

Program called by ARCIV1 at 00, 15, 30, and 45 minutes after the hour. When finished, this program in turn calls program HRAV.

INITIALIZATION:

This program reads averaged results from its last execution from file WD:ARCIV2.DAT as well as Bush Field data from Global Common ARCOM. In addition, the program uses Quality Assurance Flags set either by ARCIV1 Subroutine ARCUBF or manual Flag setting Pgm FLAGS. These Flags are also located in Common ARCOM.

REAL TIME CONSIDERATIONS:

Program must complete execution prior to beginning of the next 15 minute period. It normally takes about 4 seconds to execute. Event flag 35 is used (by Subroutines COMGET and COMPUT) to secure access to Global Common ARCOM.

AUTHOR: M.M. Pendergast, ETD, Savannah River Lab, E.I. DuPont
J.R. Froggatt, On-Line Systems Group, ESD, E.I. DuPont

DATE: 12/82
LAST REVISED: 1/88

ADDITIONAL REMARKS ON THE Q/A ALGORITHM:

The Q/A values computed during the last 15 min are used to replace SR tower data in the event that all towers are out. QA7TO is used to Q/A the SR tower data. 1st, data screened by pgm ARCIV1 is flagged. Then D1 data is saved and replaced with last 15 min Q/A values or Bush field values if last Q/A values are 0. (NOTE: Bush field data is not height adjusted) Next the expected variation for speed is computed using previous Q/A speed and SIGA (per DP-1551). Then subroutine TEST is used to determine bad values and replace with avg. of remaining or last period ave. if all readings bad. Next Q/A SIGE per DP-1551 using the just computed Q/A speed average to determine expected SIGE variation. Next Q/A SIGA per DP-1551 again using current speed average. Now AZM is corrected. The first approximation of expected SIGA variation is set equal to the current value of the average SIGA value multiplied by the number of standard deviations allowed. This value is clamped to a maximum of 45 deg. Then plume meander is accounted for if current SIGE is less than 2 degrees. (NOTE: this supposedly applies during very stable conditions - this is not checked for). The plume meander is clamped at 90 degrees rather than the 30 degrees mentioned in DP-1551. Next, the latest stability class is calculated based on the current average value of SIGA. Finally, the

Q/A values of D1 are computed. Using the current estimate of stability, the quality assured values of D2 are converted to the D1 18 M level using subroutines ADJUST and LLJUST for exponential and linear conversions respectively. Any values found to be incorrect are replaced with the corresponding D2 values using subroutine CHECK.

Now the TV tower is quality assured by QATVTO. 1st the temps. are converted from degrees C to absolute. Then they are Q/A'd. Per DP-1551, the expected variation is 5 degrees C, however, the program uses 10 deg. Subroutine TTFIX converts all temps to the 36 M using subr. LLJUST. The bad values are flagged and replaced by subroutine TEST. These replacement values are overlayed with curve fitted values by subroutine TTCOR. This only occurs for points above 10 M and only if three or more good points exist. Then a scheme is used to alter any corrected values above 10 M taking potential temperature into consideration. Next, speed is Q/A'd. The same formula is used as with the SR towers except the current SR tower speed, SIGA, and SIGE are used as opposed to last period. This was out of necessity the case with the SR towers. Subroutine TVFIX is called to identify the invalid data.

TVFIX is subsequently called for SIGA, SIGE, and AZM corrections. It is a general subroutine designed to correct different kinds of data. First it calls either ADJUST or LLJUST to convert the data to the 61 M level. Then, COMPAR is called to flag invalid data (NOTE: COMPAR does not compare the values against each other but against the Q/A SR tower average just computed). Then the bad values are replaced with the average of the remaining TV values or the SR average if all TV values were bad. Then the replaced data is adjusted to the appropriate height using ADJUST or LLJUST. At this point control is returned to QATVTO. Currently all TV tower corrected values are overlayed by TVCOR (except TTCOR for temps) which supplies curve fitted values unless fewer than 3 good points exist. In this case, the corrected values stand as is.

After speed, SIGA and SIGE are corrected using the same formulas (and values in this case) as with the SR towers. TVFIX and TVCOR perform the calculations as was the case for speed corrections. The final speed values are clamped to no less than 0.2 M/S and the SIGA and SIGE values are restricted to lie between 0.1 and 90.0.

Azimuths are handled nearly the same way that they are first run through the Kern/Pendergast algorithm to properly handle values in the 0 degree or 360 degree region. Then they are run through TVFIX and TVCOR. Then they are run through a MOD 360 operation to restrict values to 0 - 360 deg.

Finally, temperatures are converted back from absolute to degrees C.

The main program then replaces the "number of points" variable for all QA'd data with a number appropriate to the amount of time that the machine was collecting data during the 15 minute period (15 min = 600 points).

The U, V variables are then changed to agree with any AZM or speed corrections (COMPUT). Then the data is inserted into the data base by subroutine (ARCPUT).

1.12 ARCPUT (ARCIV2 SUBR - Write 15 Minute Average Data To Disk)

IDENTIFICATION:

Quality Assurance Program

NAME: ARCIV2

INSTALLED NAMES: ARCIV2

TYPE: WIND System Data Collection Program

FUNCTION:

This program is called at the end of every 15 minute period by data collection program ARCIV1. The program's purpose is to QUALITY Assure the data and then write it to disk using subroutine ARCPUT. Erroneous data is replaced with appropriate estimates. The Quality Assurance Algorithm implemented in the WIND System operates in two phases. The first phase consists of three schemes for determining invalid data, while the second phase is concerned with replacing the invalid data with estimated values.

Data is flagged as being invalid if an instrument is known to be out of service (done manually using pgm flags). In addititon, if less than 10% of the expected number of readings for a 15 minute period for a particular instrument are within the known ranges of the instrumentation, the data for that period and instrument is flagged (performed by ARCIV1 subroutine ARCUBF). The data that pass this initial screening is further checked against the average of all similar data items (e.g., all SRS Tower Azimuths). Any observation that differs by more than a specific amount from the mean is flagged, and a new mean is calculated from the remaining points. This procedure continues until an acceptable mean is computed that involves at least two instruments, or all of the observations are flagged.

The allowed deviation from the mean is a specified number of estimated standard deviations of the variable in question.

The estimated standard deviations are calculated with empirical relationships based on meteorological conditions and statistical paramters, such as Atmosperic Stability and Sampling Period.
An additional consideration is required in Quality Controlling the data collected on the WJBF-TV Tower. Here the individual data points are compared to the mean values of like data items at different tower heights. In order to make meaningful comparisons, all data are adjusted to a common reference height. Speed, Siga, and Sige are adjusted using a power law relationship while Temperature and Wind Direction are adjusted using a linear relationship. In all cases, equation parameters are used that have been empirically developed as a function of Atmospheric Stability.

The second phase of the Quality Assurance Algorithm involves replacement of bad data with estimated values. For the SRS Towers, this merely consists of substituting the mean of the accepted values for the rejected data. If all values of a particular type have been rejected, the last available Average SRS Tower is used. However, if this is more than two hours old, adjusted data from Bush Field in nearby Augusta, Georgia are used. If Bush Field data is more than two hours old, no Quality Assurance is performed and a message to that effect is displayed.

TV Tower data is replaced by fitting a curve to the accepted values, and substituting the value derived from the curve for the rejected value. If too few points exist to generate a reasonable curve, the average value calculated for the reference height is adjusted to the level in question. If all TV readings for a particular type of variable are missing, the Height-Adjusted SRS Tower Average is used.

After the new values have been inserted the program calls Subroutine ARCPUT to write the 15 minute data to disk. Both the original and the corrected set of data is written to disk by ARCPUT.

CALLING SEQUENCE:

Program called by ARCIV1 at 00, 15, 30, and 45 minutes after the hour. When finished, this program in turn calls program HRAV.

INITIALIZATION:

This program reads averaged results from its last execution from file WD:ARCIV2.DAT as well as Bush Field data from Global Common ARCOM. In addition, the program uses Quality Assurance Flags set either by ARCIV1 Subroutine ARCUBF or manual Flag setting Pgm FLAGS. These Flags are also located in Common ARCOM.

REAL TIME CONSIDERATIONS:

Program must complete execution prior to beginning of the next 15 minute period. It normally takes about 4 seconds to execute. Event flag 35 is used (by Subroutines COMGET and COMPUT) to secure access to Global Common ARCOM.

AUTHOR: M.M. Pendergast, ETD, Savannah River Lab, E.I. DuPont
J.R. Froggatt, On-Line Systems Group, ESD, E.I. DuPont

DATE: 12/82
LAST REVISED: 1/88

ADDITIONAL REMARKS ON THE Q/A ALGORITHM:

The Q/A values computed during the last 15 min are used to replace SR tower data in the event that all towers are out. QA7TO is used to Q/A the SR tower data. 1st, data screened by pgm ARCIV1 is flagged. Then D1 data is saved and replaced with last 15 min Q/A values or Bush field values if last Q/A values are 0. (NOTE: Bush field data is not height adjusted) Next the expected variation for speed is computed using previous Q/A speed and SIGA (per DP-1551). Then subroutine TEST is used to determine bad values and replace with avg. of remaining or last period ave. if all readings bad. Next Q/A SIGE per DP-1551 using the just computed Q/A speed average to determine expected SIGE variation. Next Q/A SIGA per DP-1551 again using current speed average. Now AZM is corrected. The first approximation of expected SIGA variation is set equal to the current value of the average SIGA value multiplied by the number of standard deviations allowed. This value is clamped to a maximum of 45 deg. Then plume meander is accounted for if current SIGE is less than 2 degrees. (NOTE: this supposedly applies during very stable conditions - this is not checked for). The plume meander is clamped at 90 degrees rather than the 30 degrees mentioned in DP-1551. Next, the latest stability class is calculated based on the current average value of SIGA. Finally, the Q/A values of D1 are computed. Using the current estimate of stability, the quality assured values of D2 are converted to the D1 18 M level using subroutines ADJUST and LLJUST for exponential and linear conversions respectively. Any values found to be incorrect are replaced with the corresponding D2 values using subroutine CHECK.

Now the TV tower is quality assured by QATVTO. 1st the temps. are converted from degrees C to absolute. Then they are Q/A'd. Per DP-1551, the expected variation is 5 degrees C, however, the program uses 10 deg. Subroutine TTFIX converts all temps to the 36 M using subr. LLJUST. The bad values are flagged and replaced by subroutine TEST. These replacement values are overlayed with curve fitted values by subroutine TTCOR. This only occurs for points above 10 M and only if three or more good points exist. Then a scheme is used to alter any corrected values above 10 M taking potential temperature

into consideration. Next, speed is Q/A'd. The same formula is used as with the SR towers except the current SR tower speed, SIGA, and SIGE are used as opposed to last period. This was out of necessity the case with the SR towers. Subroutine TVFIX is called to identify the invalid data.

TVFIX is subsequently called for SIGA, SIGE, and AZM corrections. It is a general subroutine designed to correct different kinds of data. First it calls either ADJUST or LLJUST to convert the data to the 61 M level. Then, COMPAR is called to flag invalid data (NOTE: COMPAR does not compare the values against each other but against the Q/A SR tower average just computed). Then the bad values are replaced with the average of the remaining TV values or the SR average if all TV values were bad. Then the replaced data is adjusted to the appropriate height using ADJUST or LLJUST. At this point control is returned to QATVTO. Currently all TV tower corrected values are overlayed by TVCOR (except TTCOR for temps) which supplies curve fitted values unless fewer than 3 good points exist. In this case, the corrected values stand as is.

After speed, SIGA and SIGE are corrected using the same formulas (and values in this case) as with the SR towers. TVFIX and TVCOR perform the calculations as was the case for speed corrections. The final speed values are clamped to no less than 0.2 M/S and the SIGA and SIGE values are restricted to lie between 0.1 and 90.0.

Azimuths are handled nearly the same way that they are first run through the Kern/Pendergast algorithm to properly handle values in the 0 degree or 360 degree region. Then they are run through TVFIX and TVCOR. Then they are run through a MOD 360 operation to restrict values to 0 - 360 deg.

Finally, temperatures are converted back from absolute to degrees C.

The main program then replaces the "number of points" variable for all QA'd data with a number appropriate to the amount of time that the machine was collecting data during the 15 minute period (15 min = 600 points).

The U, V variables are then changed to agree with any AZM or speed corrections (COMPUT). Then the data is inserted into the data base by subroutine (ARCPUT).

1.13 T_SWITCH_SET (Set T-Switches To Allow Data Collection By Selected Computer)

IDENTIFICATION:

Archiving system T-Switch set program

NAME: TSWITCH

INSTALLED NAMES: SWNORM/SWIND1/SWIND2

TYPE: WIND System general support program

FUNCTION:

The program functions as a foreign command and switches the T-Switches directing data and terminal lines to either of the two WIND System VAX computers. If the command line contains "SWNORM" then the normal switch settings on T_SWITCH_1 of (0-7) go to A (WIND2::) and (8-15) go to B (WIND1::). The normal switch settings on T_SWITCH_2 are all switches set to A (WIND2::). If the the command line contains "SWIND2" then all switches are set to A on both T switches and if the command line contains "SWIND2" all switches are set to B on both T switches.

INITIATION:

Program is requested for execution from a terminal by typing SWNORM, SWIND1, or SWIND2 to switch the A/B switches as described above.

AUTHOR: J.R. Froggatt, Process Systems Group, ESD, E. I. DuPont

DATE: 2/14/85
LAST MODIFIED: 9/15/88

1.14 HRAV (Update WD:HRAIX1/2.DAT File For Plots / Calcs Requiring Hourly Data)

NAME: HRAV

FUNCTION: Create data for plots and calculations requiring hourly data

TYPE: WIND System Data Collection Program

FUNCTION DESCRIPTION:

This program obtains data from the last four 15 minute records stored by the WIND System in either file WD:ARCDT1.DAT or WD:ARCDT2.DAT. The program updates file WD:TVSVDT.DAT which that can then be accessed by a number of downstream programs. This program also calls subroutine HRIXUP to update the current week's hourly SAM File (WD:HRAIX1.DAT OR WD:HRAIX2.DAT). The current week's SAM file is used by program HRAVPT for plotting the plume of an actual or hypothetical release. The program then opens and closes SLEOC1::[WIND_GET]LOGIN.COM to allow that system to

copy over files that could be used for emergency response calculations in the event that communications were cut between both WIND SYSTEM VAX's and the EOC VAX.

Routine modified 3/87 to write file WD:WIND_EOC.DAT containing current system time along with time of last WIND data stored on current system. This data will be used by SLEOC1:: to set it's time equal to that of current system. The time of the last data stored will be used by WD:WIND_EOC:: to determine which archiving files to access (WD:HRAIX1.DAT and WD:ARCDT1.DAT or WD:HRAIX2.DAT and WD:ARCDT2.DAT).

INITIATION:

Program initiated by ARCIV2 every 15 minutes after it has updated file WD:ARCDT1.DAT or WD:ARCDT2.DAT at 0, 15, 30, and 45 minutes after the hour.

REAL TIME CONSIDERATIONS:

Precaution is taken using the Lock Manager to secure exclusive access to the various data files used.

AUTHOR: J.R. Froggatt, Process Computer Services Group, ESD, E.I. DuPont

DATE: 6/15/83
LAST MODIFIED: 2/21/89

1.15 HRAV1 (Program To Update TEST Hourly SAM File WA:SAMyyyymm.DATA)

NAME: HRAV1

FUNCTION: TEST program to update indexed sequential hour avg file

TYPE: WIND System Data Collection Program

FUNCTION DESCRIPTION:

This TEST program obtains data from the last four 15 minute records stored by the WIND System in either file WD:ARCDT1.DAT or WD:ARCDT2.DAT. It then calls subroutine HRIXUP to update the current months's hourly indexed sequential SAM File (WA:SAMyyyymm.DAT).

INITIATION:

Program initiated by HRAV every 15 minutes after it has updated the weekly (actual production file) SAM file WD:HRAIX2.DAT or WD:HRAIX2.DAT at 0, 15, 30, and 45 minutes after the hour.

REAL TIME CONSIDERATIONS:

Precaution is taken using the Lock Manager to secure exclusive access to the various data files used.

DATE: 11/85

1.16 SAMUP (HRAV1 SUBR - Update WA:SAMyyyymm.DAT With Data Supplied By HRAV1)

IDENTIFICATION:

Subroutine to update SAM file

NAME: SAMUP

TYPE: HRAV1 Subroutine

FUNCTION:

This subroutine is called by program HRAV1 to update the hourly SAM file. Subroutine SAMLOC is called to determine which file (WA:SAMyyyymm.DAT) is to be used and which record within it (key = yyyymmddhh0000.00) is to be written. The data which has been passed by HRAV1 is then written into the correct record of the file.

CALLING SEQUENCE:

Subroutine is called by program HRAV1. This subroutine calls subroutine SAMLOC.

REAL TIME CONSIDERATIONS:

Lock 'WA:SAMyyyymm.DAT' is used to obtain exclusive access to the whatever month's SAM file (WA:SAMyyyymm.DAT) is being updated. This is to avoid file contention with any program attempting to correct or retrieve data from the file.

AUTHOR: J.R. Froggatt, On-Line Systems Group, ESD, E.I. DuPont

DATE: 6/84

1.17 SAMLOC (HRAV1 SUBR - Determine File Name And Rec Key For WA:SAMyyyymm.DAT)

IDENTIFICATION:

Calculate file name and record key of next SAM file entry

NAME: SAMLOC

TYPE: HRAV1 subroutine

FUNCTION:

Subroutine determines file name (WA:SAMyyyymm.DAT) and record key (yyyymmddhh0000.00) of of next record to write into current month's SAM data file. It uses the date and time supplied to it by the calling program to do this.

CALLING SEQUENCE:

Called by subroutine SAMUP which in turn is called by HRAV1.

AUTHOR: J.R. Froggatt, On Line Systems Group, ESD, E.I. DuPont

DATE: 6/19/84

1.18 INCKEY (HRAV1 SUBR - Increment Key For TEST SAM File WA:SAMyyyymm.DAT)

IDENTIFICATION:

Subroutine to update SAM file

NAME: SAMUP

TYPE: HRAV1 Subroutine

FUNCTION:

This subroutine is called by program HRAV1 to update the hourly SAM file. Subroutine SAMLOC is called to determine which file (WA:SAMyyyymm.DAT) is to be used and which record within it (key = yyyymmddhh0000.00) is to be written. The data which has been passed by HRAV1 is then written into the correct record of the file.

CALLING SEQUENCE:

Subroutine is called by program HRAV1. This subroutine calls subroutine SAMLOC.

REAL TIME CONSIDERATIONS:

Lock 'WA:SAMyyyymm.DAT' is used to obtain exclusive access to the whatever month's SAM file (WA:SAMyyyymm.DAT) is being updated. This is to avoid file contention with any program attempting to correct or retrieve data from the file.

AUTHOR: J.R. Froggatt, On-Line Systems Group, ESD, E.I. DuPont

DATE: 6/84

1.19 ZEROAC (Zero Out Previous Week's Archiving Files)

NAME: ZEROAC

FUNCTION: Zero out previous week's Archiving files.

TYPE: WIND System Data Collection Program

FUNCTION DESCRIPTION:

This program runs every day and first determines the current day of the week. If it is Sunday - Wednesday, the program copies the previous week's SAM average file to directory [WIND.ARCH] (WA:) and exits. This is done four times in case the machine is down at some point.

If it is Thursday - Saturday, the program zeroes out the previous week's Archive file (WD:ARCDT1.DAT or WD:ARCDT2.DAT). In addition, the program zeros out the previous week's SAM Average file (WD:HRAIX1.DAT or WD:HRAIX2.DAT *** TO BE NAMED WD:SAMDTn.DAT AT SOME FUTURE DATE***). The program zeros out the previous week's files on Thursday - Saturday to insure that this is accomplished even though the machine may be down at the time some of the runs of this program are scheduled. Added protection can be achieved by running the program more than once a day and/or by calling for its execution in the startup file.

INITIATION: Scheduled as repetitive Process at system Boot time

AUTHOR: J.R. Froggatt, On Line Systems Group, ESD, E.I. Dupont

DATE: 6/8/83
LAST REVISED: 1/89

CHAPTER 2

** EOC DATA COLLECTION PROGRAMS **

2.1 ARCIV_EOC (Insert Data From SLWND:: Cluster Into EOC UVAX SLEOC1:: Files)

IDENTIFICATION:

PROGRAM TO WRITE CURRENT 15 MINUTE ARCHIVING RECORD ON EOC VAX

NAME: ARCIV_EOC

TYPE: WIND SYSTEM DATA COLLECTION PROGRAM

FUNCTION:

This program is run every 15 minutes on SLEOC1:: and reads file WD:ARCIV_EOC.DAT which has just been copied over from one of the two WIND system VAX's (SLWND1:: or SLWND2::). The file is two records long and contains the header record and current 15 minute data record from the current 15 minute WIND system file WD:ARCDT1.DAT or WD:ARCDT2.DAT. This program will insert the data into WD:ARCDT1.DAT or WD:ARCDT2.DAT on SLEOC1::. Then the header records of the updated files will be updated to reflect the data just inserted.

**** FUTURE FEATURE ****

This program will then compare the two header records to see if data on SLEOC1:: is as complete as that on the WIND system. If not, the most current missing four or less records will be retrieved and inserted into the file on SLEOC1::. If more then four records are missing, an additional four or less records will be retrieved during the next run of this program in 15 minutes.

CALLING SEQUENCE:

Program execution is requested every 15 min at 00, 15, 30, and 45 minutes after the hour by SLEOC1::[WIND_GET]LOGIN.COM which is initiated by SLWND1:: unless that system is down. In that case, it is initiated by SLWND2:: if that system is up.

REAL TIME CONSIDERATIONS:

Program must complete execution prior to the beginning of the next 15 minute period. It normally takes less than five seconds to execute. No consideration is given to file contention since VMS allows multiple access to sequential files. Also, this program only reads data from SLWND:: and is the only program writing to WD:ARCDT1.DAT or WD:ARCDT2.DAT on SLEOC1::.

AUTHOR: J.R. Froggatt, Process Computer Services Group, ESD, E.I. DuPont

DATE: 3/19/87
REVISED: 1/89

2.2 TIMESET_EOC (Set Time On EOC UVAX SLEOC1:: To That Of SLWND:: Cluster)

IDENTIFICATION:

PROGRAM TO SET TIME ON SLEOC1:: TO TIME FOUND IN SLEOC::WD:WIND_EOC.DAT

NAME: TIMESET_EOC

TYPE: PROGRAM TO FACILITATE DATA GATHERING

FUNCTION:

This program was developed to allow EOC VAX SLEOC1:: to read SLEOC1::WD:WIND_EOC.DAT which contains the current time of the calling system (SLWND1:: or SLWND2::) and set SLEOC1:: time equal to it. SLEOC1::WD:WIND_EOC.DAT also contains the time and date of the last WIND data stored on the calling machine. This allows SLEOC1:: to determine the names of the WIND data files (ARCDT1.DAT and HRAIX1.DAT or ARCDT2.DAT and HRAIX2.DAT) that must be kept in sync between SLEOC1:: and the calling system.

CALLING SEQUENCE: Called by SLEOC1::SRLUSER:[WIND_GET]LOGIN.COM if login being made via mode "NETWORK".

AUTHOR: J.R. Froggatt, Process Computer Services Group, ESD, E.I. DuPont

DATE: 3/17/87

2.3 ZEROAC_EOC (Zero Out Previous Week's Archiving Files On EOC UVAX SLEOC1::)

NAME: ZEROAC_EOC

FUNCTION: Zero out previous week's Archiving files on EOC VAX.

TYPE: WIND System Data Collection Program (EOC version).

FUNCTION DESCRIPTION:

This program runs every day and first determines the current day of the week. If it is Sunday - Wednesday, the program exits.

If it is Thursday - Saturday, the program zeroes out the previous week's Archive file (WD:ARCDT1.DAT or WD:ARCDT2.DAT). In addition, the program zeros out the previous week's SAM Average file (WD:HRAIX1.DAT or WD:HRAIX2.DAT). The program zeros out the previous week's files on Thursday - Saturday to insure that this is accomplished even though the machine may be down at the time some of the runs of this program are scheduled. Added protection can be achieved by running the program more than once a day and/or by calling for its execution in the startup file.

INITIATION: Scheduled as repetitive Process at system Boot time

AUTHOR: J.R. Froggatt, Process Computer Services Group, ESD, E.I. Dupont

DATE: 3/11/87
LAST MODIFIED: 1/89

CHAPTER 3

** DATA CORRECTION PROGRAMS **

CHAPTER 4

** DATA RETRIEVAL PROGRAMS **

4.1 ARCLST (Generate Listing Of Archived 15 Minute Meteorological Data)

IDENTIFICATION: Program to list TV, SRS Twr, and Clim Twr Data

NAME: ARCLST

TYPE: Archive Retrieval Program

FUNCTION:

This program obtains data from the WIND System Archive File beginning at the time and date specified at the keyboard. It acquires information from file WD:ARCDT1/2.DAT using data from 1 - 96 15 minute periods as requested at the keyboard. It then calls subroutine ARCWRT (or CLMWRT if Central Climatology Data was requested) to list the data. The program then collects data from the next 1 - 96 records and again calls subroutine ARCWRT (or CLMWRT). This process continues until either the data from the WIND System Archive file is exhausted or the requested number of listings has been generated.

AUTHOR: J.R. Froggatt, On-Line Systems Group, ESD, E.I. DuPont

DATE: 3/82
LAST REVISED: 11/87

4.2 ARCWRT (ARCLST SUBR - Format And Write TV And SRS Tower Data)

IDENTIFICATION: Routine to list archived data

NAME: ARCWRT

TYPE: ARCLST Subroutine

FUNCTION:

Subroutine lists TV and SRS tower data accumulated by program ARCLST in response to a keyboard request for a listing of archived data.

AUTHOR: J.R. Froggatt, On-Line Systems Group, ESD, E.I. DuPont

DATE: 3/82
LAST REVISED: 11/87

4.3 CLMWRT (ARCLST SUBR - Format And Write Central Climatology Tower Data)

IDENTIFICATION: Routine to list archived data

NAME: CLMWRT

TYPE: ARCLST Subroutine

FUNCTION:

Subroutine lists Central Climatology Data accumulated by program ARCLST in response to a keyboard request for a listing of archived data.

AUTHOR: J.R. Froggatt, Process Systems Group, ESD, E.I. DuPont

DATE: 11/14/85

4.4 MATLST (Generate ASCII Formatted File Of Archived 15 Minute Met Data)

IDENTIFICATION: Program to format and write archived data to a file

NAME: MATLST

TYPE: WIND System Data Retrieval Program

FUNCTION:

This program obtains data from the WIND System Archive File beginning at the time and date specified at the keyboard. It acquires information from file WD:ARCDT1/2.DAT using data from 1 - 96 15 minute periods as requested at the keyboard. It then calls subroutine MATWRT to list the data on printer and write to disk. The program then collects data from the next 1 - 96

records and again calls subroutine MATWRT. This process continues until either the data from the WIND System Archive file is exhausted or the requested number of listings has been generated.

AUTHOR: J.R. Froggatt, On Line Systems Group, ESD, E.I. DuPont

DATE: 12/83
LAST REVISED: 2/89

4.5 MATWRT (MATLST SUBR - Write Data Supplied By MATLST To Disk)

IDENTIFICATION: Subroutine to format and write archived data to a file

NAME: MATWRT

TYPE: MATLST Subroutine

FUNCTION:

This subroutine formats WIND System Archive Data accumulated by program calling program MATLST and writes it to a disk file. The information is written in ASCII form to 80 character fixed length records. This makes the data easy to read on almost any other system. Program MATLST, supplies data to this routine in response to a keyboard request specifying the specific archived data that is required.

AUTHOR: J.R. Froggatt, On Line Systems Group, ESD, E.I. DuPont

DATE: 12/83
LAST REVISED: 2/89

CHAPTER 5

** GENERAL SUPPORT PROGRAMS AND COMMAND PROCEDURES **

5.1 ARCTAP (Transfer Archive/SAM Files Between Disk And Tape)

NAME: ARCTAP.COM

FUNCTION: Transfer WIND System Files Between Disk and Tape or WORM disk

TYPE: WIND System General Support Procedure

FUNCTION DESCRIPTION:

This procedure will operate in either of four modes depending upon the value of a parameter supplied to it when it is initiated from a keyboard. If the parameter value begins "SAMTAP" the procedure will operate on SAM (Site Area Mean) data files. If the parameter is non-existant or if it is any other value, the procedure will operate on the main WIND system archiving files. The comments below generally apply to either mode that the procedure may be operating in. However, when a comment applies only to the main WIND system archiving files, the corresponding information pertaining to SAM data files follows immediately in parentheses.

This procedure will either transfer archived WIND System Archive (SAM) data from disk to tape or from tape back to disk into tape restore file WD:ARCDT3.DAT (WD:HRAIX3.DAT). If it is requested that data be transferred to tape, the procedure will dump the previous week's Archive (SAM) File to tape. This will be either WD:ARCDT1.DAT or WD:ARCDT2.DAT (WD:HRAIX1.DAT or WD:HRAIX2.DAT) depending upon which of those two alternating files is currently being used to store Archive (SAM) Data. Program FILSEL is called to determine which file was used last week and what the name of the mag tape file should be. The mag tape file will be named ARCyymmdd.DAT (SAMyymmdd.DAT) where yymmdd represents the the date of Sunday of last week. This is the 1st date of data storage for that week. In addition to dumping and retrieving tape data, the procedure is also able to provide a directory of the files on tape. It can also rerun a dump operation if the 1st one was done incorrectly.

It is also able to overlay the previous week's disk data file with the corresponding tape file (as opposed to writing it into tape restore file WD:ARCDT3.DAT (WD:HRAIX3.DAT)).

The procedure is able to store 2 month's (2 year's) data on a 1200 ft tape and 4 month's (4 year's) data on a 2400 ft tape at a density of 1600 BPI. The procedure has been modified to use the new tape drive and writes to tape at a density of 6250 BPI. However, the new data file record format has changed from 6 block records for a 15 minute time period to 15 block records. The amount of data that can be stored on tape under these circumstances has not yet been determined as of 12/07/88. The procedure asks whether the tape to be written to is new or old. If new, it asks for confirmation that it is ok to write over all existing tape data. If so, the dump proceeds. If the tape is old, it already has at least one week's data on it. In this case, the procedure asks for confirmation that it is ok to add a new data file to the end of the tape. If so, the procedure by default advances to the end of the data already existing on the tape and proceeds to add the new data. However, if it is desired to skip over a user specified number of files before writing, this can be accomplished by the proper keyboard input.

It is suggested that FILECK be run in conjunction with the dumping operation. That program reads the header records of the Archiving Files (same info from FILECK applies to SAM tapes) and indicates the time span covered by each and what data if any is missing. The portion of this information that applies to the previous week's file should be kept with the tape as a record of the data stored on it.

If it is desired to restore tape data to disk, the procedure will ask the name of the file to be retrieved. File names correspond to the beginning date stored in each file (eg. ARC890108.DAT (SAM890108.DAT)). Each set of data starts on a Sunday. Even if Sunday data is missing due to system outage, the data file is still named for that Sunday. The procedure will ask for confirmation that it is ok to write to disk. If so, the data will be written to tape restore file WD:ARCDT3.DAT (WD:HRAIX3.DAT). This file can then be accessed by ARCDSP (PLMDSP) to generate plots of the data. There is one exception to the way data is restored from tape. If the data is from before 01/01/89, then it was stored on tape without Central Climatology or Area Tower Temperature data. In this case, this procedure first restores the data to the short format file WD:ARCIX3.DAT. It then runs the conversion program IX3_TO_DT3 to read the data from WD:ARCIX3.DAT and output the data to WD:ARCDT3.DAT in expanded format. Zeroes will appear in the WD:ARCDT3.DAT file where Climatology or Area Temperture data would normally be found. Although it is not prompted for, if an operation of (O)verlay is specified instead of (RE)store, the tape data will overlay the previous week's Archive (SAM) File WD:ARCDT1.DAT or WD:ARCDT2.DAT (WD:HRAIX1.DAT or WD:HRAIX2.DAT)

rather than the tape restore file. This is not prompted for because it is not likely to be required very often and because it has the potential of destroying the previous week's data.

INITIATION: Keyboard Request

INITIALIZATION:

Through keyboard dialog, the procedure is supplied with the information required to perform the desired operation.

AUTHOR: J.R. Froggatt, On-Line Systems Group, ESD, E.I. DuPont

DATE: 9/24/82
LAST MODIFIED: 1/89

5.2 COEF_CL (Update Engr Units Conversion Coefficient File - Cent Climatology)

IDENTIFICATION:

Program to keyboard adjust Central Climatology coefficients

NAME: COEF_CL

INSTALLED NAMES: COEF_CL

TYPE: WIND System general support program

FUNCTION:

This program displays and allows the alteration of the engineering units calculation coefficients contained in WD:COEF_CL.DAT. The values are read by ARCIV1 subroutine ARCINI when ARCIV1 begins execution. After values are modified by this program and inserted into WD:COEF_CL, the user is given the option of also modifying the coefficients as they appear in common ARCOM_CL. This eliminates the need to stop and start ARCIV1 in order to allow the coefficients to be used. However, this must be done on both SLWND1 and SLWND2 since either machine may collect data depending upon whether the primary data collection machine (SLWND2) or the standby data collection machine (SLWND1) is performing data collection.

INITIATION:

The program is initiated from a terminal.

REAL TIME CONSIDERATIONS:

In order to put changes into common ARCOM_CL without

restarting ARCIV1, the modify memory option of this program may be used when running it on both SLWND1:: and SLWND2::.

AUTHOR: J.R. Froggatt, Process Systems Group, ESD, E.I. DuPont

DATE: 11/6/85

5.3 COEF_TP (Update Engr Units Conversion Coefficient File - SRS Twr Temps)

IDENTIFICATION:

Program to keyboard adjust Met tower temperature coefficients

NAME: COEF_TP

INSTALLED NAMES: COEF_TP

TYPE: WIND System general support program

FUNCTION:

This program displays and allows the alteration of the engineering units calculation coefficients contained in WD:COEF_TP.DAT. The values are read by ARCIV1 subroutine ARCINI when ARCIV1 begins execution. After values are modified by this program and inserted into WD:COEF_TP, the user is given the option of also modifying the coefficients as they appear in common ARCOM_TP. This eliminates the need to stop and start ARCIV1 in order to allow the coefficients to be used. However, this must be done on both WIND1:: and WIND2:: since either machine may collect data depending upon whether the primary data collection machine (WIND2::) or the standby data collection machine (WIND1::) is performing data collection.

INITIATION:

The program is initiated from a terminal.

REAL TIME CONSIDERATIONS:

Corrections made on WIND2:: will be copied to WIND1:: at the end of the current 15 minute interval although the changes will not be read by ARCIV1 on that machine until the program is restarted. In order to put changes into common ARCOM_TP without restarting ARCIV1, the modify memory option of this program may be used when running it on both WIND1:: and WIND2::

AUTHOR: J.R. Froggatt, Process Computer Services Group, ESD, E.I. DuPont

DATE: 9/11/87

5.4 FILSEL (Determine Disk And Tape File Names For Procedure ARCTAP)

NAME: FILSEL

FUNCTION: Determine tape/disk file names corresponding to prev week

TYPE: WIND System general support program

FUNCTION DESCRIPTION:

This program determines which file was used by the Archiving System during the previous week (1 or 2) and the date of the beginning data (Sunday of previous week). It communicates with the calling command procedure (ARCTAP running in ARCTAP, SAMTAP, ARCWRM, or SAMWRM mode depending on whether regular WIND 15 minute files or SAM average files are being handled and whether tape or WORM disk is being used) thru the use of logical name assignments. Output is in the form of three process logical name assignments. The error code (0 or 1), disk file number (1 or 2) of the file used the previous week, and the date of the Sunday of that week (yymmdd) are returned respectively thru logical name equivalence assignments to LOGICAL_ERROR_INDICATOR LOGICAL_FILE_NO, and LOGICAL_FILE_DATE.
The calling procedure then uses this information to transfer files between disk and tape. For example, if the calling procedure is ARCTAP (ARCTAP running in ARCTAP mode), it can upon request transfer last week's Archive file (WD:ARCDT1.DAT or WD:ARCDT2.DAT) to tape or WORM disk with the name reflecting the date of the Sunday (1st day) of that week (ARCH$DEV:ARCyymmdd.DAT). The date will always be of Sunday even if there are gaps in the data and actual data does not begin until later in the week.

INITIATION:

Program is initiated by disk/tape (or WORM disk) transfer procedure ARCTAP.COM which has been invoked to act as ARCTAP, SAMTAP, ARCWRM, or SAMWRM.

CALLING SEQUENCE:

The program calls subroutine INXCLC to determine which disk file is being used during the current week and therefore which was used the previous week (1 or 2). It then calls subroutine TIMCLC to determine the 1st day of the previous week. Although the results would be the same regardless of whether one of the regular archiving files is accessed (WD:ARCDT1.DAT or WD:ARCDT2.DAT) or one of the SAM files is accessed (WD:HRAIX1.DAT or WD:HRAIX2.DAT), for consistency the file appropriate to the calling procedure (ARCxxx or SAMxxx) is the one that is accessed for this information.

INITIALIZATION:

This program determines whether it was called by ARCTAP, SAMTAP,

ARCWRM, or SAMWRM (actually ARCTAP.COM acting as ARCxxx or SAMxxx depending upon how it was invoked) by determining the equivalence name of process logical name LOGICAL_CALLING_PROCEDURE. It then accesses the appropriate files based upon this information.

REAL TIME CONSIDERATIONS: N/A

AUTHOR: J.R. Froggatt, On-Line Systems Group, ESD, E.I. DuPont

DATE: 10/01/82
LAST MODIFIED: 1/89

5.5 FLAGS (Display And Set Q/A Flags To Indicate Instrumentout Of Service)

IDENTIFICATION:

Program to keyboard adjust quality assurance flags

NAME: FLAGS

INSTALLED NAMES: FLAGS

TYPE: WIND System general support program

FUNCTION:

This program displays and allows the alteration of the quality assurance flags that exist in global commons ARCOM, ARCOM_TP, and ARCOM_CL. The flags are set by ARCIV1 subroutine ARCUBF at the end of each 15 minute interval. These flags are subsequently used by ARCIV2 to make decisions concerning estimated data that is substituted for readings that are considered to be in error. The flag values are then written to disk by ARCIV3 along with all the data collected in the 15 minute period. In addition, the flags are written to disk file WD:ARCQTY.DAT by ARCIV3. This allows ARCIV1 to read the last saved values of the flags when beginning execution during a system boot or at any other time that it begins operation.

INITIATION:

The program is initiated from a terminal. It displays current flag values and prompts for either corrections or CTRL/Z for program termination. An example correction entry is supplied to the user. When a correction is made the entire set of flags is again displayed to allow verification of proper handling of the correcting entry.

REAL TIME CONSIDERATIONS:

Corrections made will be reflected in the above mentioned files

after the current 15 minute period has ended.

AUTHOR: J.R. Froggatt, On Line Systems Group, ESD, E.I. DuPont

DATE: 9/25/80
REVISED: 11/13/85
REVISED: 10/29/87

5.6 IX3_TO_DT3 (Copy WD:ARCIX3.DAT To WD:ARCDT3.DAT For Procedure ARCTAP)

NAME: FILSEL

FUNCTION: Determine tape/disk file names corresponding to prev week

TYPE: WIND System general support program

FUNCTION DESCRIPTION:

This program determines which file was used by the Archiving System during the previous week (1 or 2) and the date of the beginning data (Sunday of previous week). It communicates with the calling command procedure (ARCTAP running in ARCTAP, SAMTAP, ARCWRM, or SAMWRM mode depending on whether regular WIND 15 minute files or SAM average files are being handled and whether tape or WORM disk is being used) thru the use of logical name assignments. Output is in the form of three process logical name assignments. The error code (0 or 1), disk file number (1 or 2) of the file used the previous week, and the date of the Sunday of that week (yymmdd) are returned respectively thru logical name equivalence assignments to LOGICAL_ERROR_INDICATOR LOGICAL_FILE_NO, and LOGICAL_FILE_DATE.
The calling procedure then uses this information to transfer files between disk and tape. For example, if the calling procedure is ARCTAP (ARCTAP running in ARCTAP mode), it can upon request transfer last week's Archive file (WD:ARCDT1.DAT or WD:ARCDT2.DAT) to tape or WORM disk with the name reflecting the date of the Sunday (1st day) of that week (ARCH$DEV:ARCyymmdd.DAT). The date will always be of Sunday even if there are gaps in the data and actual data does not begin until later in the week.

INITIATION:

Program is initiated by disk/tape (or WORM disk) transfer procedure ARCTAP.COM which has been invoked to act as ARCTAP, SAMTAP, ARCWRM, or SAMWRM.

CALLING SEQUENCE:

The program calls subroutine INXCLC to determine which disk file is being used during the current week and therefore which was used the previous week (1 or 2). It then calls subroutine

TIMCLC to determine the 1st day of the previous week. Although the results would be the same regardless of whether one of the regular archiving files is accessed (WD:ARCDT1.DAT or WD:ARCDT2.DAT) or one of the SAM files is accessed (WD:HRAIX1.DAT or WD:HRAIX2.DAT), for consistency the file appropriate to the calling procedure (ARCxxx or SAMxxx) is the one that is accessed for this information.

INITIALIZATION:

This program determines whether it was called by ARCTAP, SAMTAP, ARCWRM, or SAMWRM (actually ARCTAP.COM acting as ARCxxx or SAMxxx depending upon how it was invoked) by determining the equivalence name of process logical name LOGICAL_CALLING_PROCEDURE. It then accesses the appropriate files based upon this information.

REAL TIME CONSIDERATIONS: N/A

AUTHOR: J.R. Froggatt, On-Line Systems Group, ESD, E.I. DuPont

DATE: 10/01/82
LAST MODIFIED: 1/89

5.7 RDAGS (Create Bush Field Data File WD:AGSDAT.DAT)

NAME: RDAGS

FUNCTION: Acquire Bush Field AFOS Data and prepare for WIND System

TYPE: WIND Syatem General Support Program

FUNCTION DESCRIPTION:

This program is a modification and combination of program RDSAO written by R.J. FAUST and PDP 11 program AGSDAT written by M.M. Pendergast and COOP student G. Green. The purpose of the program is to obtain hourly Bush Field data from the VAX based AFOS file and format it to allow easy retrieval. Input is acquired from file AD:SAO.DAT and output is written to WD:AGSDAT.DAT. (Program is run every 15 minutes to ensure that hourly Bush field data is made available soon after it is received.) Stability class is computed from Bush field observations and included with the observed data that is written to the output file. In addition, the pertinent data is inserted in Global Comon ARCOM.

INITIATION: Wakes from hibernation state every 15 minutes as set up at system boot time.

AUTHORS: R.J. Faust/M.M. Pendergast/J.R. Froggatt

DATE: 11/20/82

5.8 SNAP (List Current Values For SRP Meterological Tower Readings)

IDENTIFICATION:

Archiving system TV, SR, TP, and CL tower data snapshot program

NAME: SNAP/MAC

INSTALLED NAMES: SNAP

TYPE: WIND System general support program

FUNCTION:

When operating in SNAP mode the program obtains last set of data acquired by ARCIV1 from global commons ARCOM, ARCOM_TP, ARCOM_TP and lists it on the terminal that initiated the program. Units will be English or metric depending on whether or not the string "ENGLISH" appears on the command line used to invoke the program as a foreign command. If the program is operating in MAC mode (depending on whether or not the string "MAC" appears on the command line), it requests the area to report, the total number of times to report it, and the frequency to report it (voltages only).

INITIATION:

Program is requested for execution from a terminal by typing MAC to acquire selective repetitive data or by typing SNAP to acquire a onetime overview of all SRS readings.

AUTHOR: J.R. Froggatt, On-Line Systems Group, ESD, E. I. DuPont

DATE: 4/13/84
REVISED: 11/12/85
REVISED: 7/10/87
REVISED: 10/30/87

CHAPTER 6

** GENERAL SUBROUTINES **

6.1 HIXCLC (Calculate File And Record Number For WD:HRAIX1.DAT / WD:HRAIX2.DAT)

IDENTIFICATION:

Calculate File and Record Number of Next Plume File Entry

NAME: HIXCLC

TYPE: HRAV Subroutine

FUNCTION:

Subroutine calculates index of next record to write into current week's plume data file (WD:HRAIX1.DAT or WD:HRAIX2.DAT) and which of the two files is current. It first calls subroutine INXCLC to determine which file and record would be used for the time and date of the data in the regular archiving system (for files WD:ARCDT1.DAT and WD:ARCDT2.DAT). The subroutine then adjusts the result to comply with HRAV file requirements.

CALLING SEQUENCE:

Called by subroutine HRIXUP which in turn is called by pgm HRAV.

AUTHOR: J.R. Froggatt, On-Line Systems Group, ESD, E.I. DuPont

DATE: 1/7/81

6.2 HRIXUP (Insert Data Into WD:HRAIX1.DAT Or WD:HRAIX2.DAT)

IDENTIFICATION:

Subroutine to Update Hour Average File

NAME: HRIXUP

TYPE: General Subroutine

FUNCTION:

This subroutine is called by program HRAV or HRAVCR to update or correct the Hourly Plume File. Subroutine HIXCLC is called to determine which file (WD:HRAIX1.DAT or WD:HRAIX2.DAT) is to be used and which record within it is to be written. The data, which has been passed by HRAV is then written into the correct record position of the proper file.

CALLING SEQUENCE:

Subroutine called by Program HRAV. Subroutine calls Subroutine HIXCLC.

REAL TIME CONSIDERATIONS:

The Lock Manager is used to obtain exclusive access to the Plume File being written into (WD:HRAIX1.DAT or WD:HRAIX2.DAT). This is to avoid file contention with any program attempting to correct data in the file or retrieve data from the file.

AUTHOR: J.R. Froggatt, On-Line Systems Group, ESD, E.I. DuPont

DATE: 1/7/81
LAST REVISED: 2/89

6.3 LOCK (Grant Resource Lock)

NAME: SUBROUTINE LOCK

FUNCTION: Subroutine to grant resource locks

TYPE: General subroutine

FUNCTION DESCRIPTION:

This subroutine locks a resource specified by the calling program thru the use of the lock manager. In addition to the resource name, the mode of the lock and a time limit in which to accomplish the locking function must be specified. This time limit will normally be expressed as a real number between 0.00 and 60.00 seconds. If the number is 0.00 or less, the value of 0.00 will be used. If the number is greater than 59.9, the subroutine will wait indefinitely for the lock to be granted. The subroutine will return a lock status block (a 6 longword array required to DEQ a lock), a lock id assigned by the system and required by any lock manager system service calls that subsequently reference the lock, a status indicating why a system call may have failed, and a general

subroutine status indicating overall success or failure of the subroutine. This variable will contain a 1 for successful completion and an even number for unsuccessful completion indicating that the lock could not be effected and why.

*** *** CURRENTLY LAST 4 ARRAY POSITIONS REPRESENTING LOCK VALUE BLOCK
*** ARE NOT USED BY WIND SYSTEM AND ARE SET TO 0 BY THIS SUBROUTINE.
*** OCCASIONAL SYSTEM HANGUPS WERE EXPERIENCED WITH "%SYSTEM-W-VALNOTVALID,
*** value block is not valid" ERROR CODE 2544 ERRORS WHEN LOCK VALUE BLOCK
*** WAS FORMERLY SUPPLIED TO $ENQ ***

ARGUMENTS:

NAME	I/O	DESCRIPTION
RESOURCE	I	A 1-31 character resource name
LOCK_MODE	I	A 2 character designation of the type of lock desired as follows: NL -- null lock CR -- concurrent read CW -- concurrent write PR -- protected read PW -- protected write EX -- exclusive
TIME_LIMIT	I	A real number indicating the time limit in seconds allowed for granting the lock. If a timeout occurs, the subroutime returns with ISUB_STATUS set to 8. If number is less than 0.0 a value of 0.0 will be used. Actually 0.0 translates into the system granting the lock upon return from lock granting system service $ENQ. If the number is greater than than 59.9 the subroutine will wait indefinitely for the granting of the lock through the synchronous version of the lock granting system service, $ENQW.
LOCK_STATUS	O	A 6 integer array to contain status information about the lock. It will primarily be used as input to the UNLOCK subroutine when it is desired to unlock the resource. It is returned directly by the ENQ or ENQW system service called by this subroutine
LOCK_ID	O	The integer ID number assigned by the system to the lock. It is actually the second entry in the LOCK_STATUS array. It would be required if subsequent reference to the lock were to be

		made using an appropriate system service.
ICALL_STATUS	O	An integer containing the system service return status of the last system service called before any errors may have been detected. This would be of use mainly in the debugging of any unexpected future errors.
ISUB_STATUS	O	An integer containing the return status of the subroutine. The return values are as follows: 1 -- success. 2 -- could not grant lock when $ENQ or $ENQW called. 4 -- incorrect LOCK_MODE supplied. 8 -- time-out. 16 -- internal subroutine problem such as the inability to reserve an event flag.

INITIATION:

Segments of a sample FORTRAN program calling this subroutine and companion subroutine UNLOCK are shown below:

```
        PROGRAM SAMPLE
                .
                .
                .
        INTEGER*4 LOCK_STATUS(6)
                .
                .
                .
C...
C...    Lock file WD:ARCDT1.DAT for purposes of reading using name
C...    ATCDT as agreed upon by cooperating users. Allow 20.0
C...    seconds to accomplish locking.
C...
        CALL LOCK('ARCDT','PR',20.0,LOCK_STATUS,LOCK_ID,ICALL_STATUS,
        1                ISUB_STATUS)
        IF (ISUB_STATUS) THEN
                OPEN (UNIT=1,NAME='WD:ARCDT1.DAT',...
        ELSE
                GO TO (error handling routine)
        ENDIF
                .
                .
                .
        CLOSE(UNIT=1)
C...
C...    Release lock.
```

```
C...
      CALL UNLOCK(LOCK_STATUS,ICALL_STATUS,ISUB_STATUS)
      IF (.NOT. ISUB_STATUS) GO TO (error handling routine)
                .
                .
                .
      END
```

CALLING SEQUENCE:

This subroutine should be called prior to accessing a file (or other system resource) that is subject to being accessed by other programs in an incompatible way. The system services reference manual details the types of locks that can be placed on a resource to ensure that cooperating programs do not interfere with each other when accessing system resources. The two most common locks required are 'EX' (exclusive) for writing into a file and 'PR' (protected read) for reading from a file. 'EX' grants exclusive access to a file and 'PR' only allows other readers into the file. When access to a resource is no longer required, companion subroutine UNLOCK should be called to release the resource.

INITIALIZATION: N/A

REAL TIME CONSIDERATIONS: N/A

AUTHOR: J.R. Froggatt, On-Line Systems Group, ESD, E.I. DuPont

DATE: 7/23/82
LAST MODIFIED: 6/11/86 - eliminate lock value block handling
by $ENQ.

6.4 T_SWITCH_SET (Set T Switches To Allow Data Collection By Desired Computer)

IDENTIFICATION:

Archiving system T-Switch set function

NAME: T_SWITCH_SET

TYPE: WIND System function

FUNCTION:

This routine switches the T-Switches directing data and terminal lines to either of the two WIND System VAX computers. If the argument SETTINGS supplied to this routine is equal to the string 'SWNORM' then the normal switch settings on T_SWITCH_1 of (0-7) going to A (WIND2::) and (8-15) going to B (WIND1::) are made.

Also, the normal switch settings on T_SWITCH_2 with all switches set to A (WIND2::) are made. If SETTINGS is equal to 'SWIND2' then all switches are set to A on both T switches. If SETTINGS is equal to 'SWIND2' all switches are set to B on both T switches.

AUTHOR: J.R. Froggatt, Process Computer Services Group, ESD, E. I. DuPont

DATE: 9/15/88

6.5 TMPGET (Temporary Routine To Retrieve Current Met Data From WD:ARCCUR.DAT)

IDENTIFICATION:

Temporary sub to read current 15 minute data record from WD:ARCCUR.DAT and place into local commons ARCOM, ARCOM_TP, and ARCOM_CL.

NAME: TMPGET

6.6 UNLOCK (Release Resource Lock)

NAME: SUBROUTINE UNLOCK

FUNCTION: Subroutine to release resource locks

TYPE: General subroutine

FUNCTION DESCRIPTION:

This subroutine releases a resource lock specified by the calling program thru the use of the lock manager system service $DEQ. The calling program only needs to supply the LOCK_STATUS Block, a 6 integer array created when subroutine LOCK calls system service $ENQ or $ENQW to establish the lock. The subroutine returns the completion status supplied by $DEQ indicating that it succeded or why it failed and a general subroutine status indicating success or failure of the subroutine. This variable will contain a 1 for successful completion and a 2 for unsuccessful completion.

ARGUMENTS:

NAME	I/O	DESCRIPTION
LOCK_STATUS	I	A 6 integer array containing status information about the lock. It was returned by the LOCK subroutine when the

		lock was initially granted. It contains all the information required by system service DEQ to unlock the resource. *** Currently the Lock Value Block (last four integers in the array) are not used by the LOCK subroutine and are set to 0 by it. ***
ICALL_STATUS	O	An integer containing the return status of system service DEQ after it attempted to remove the lock. This would be of use mainly in the debugging of any unexpected future errors.
ISUB_STATUS	O	An integer containing the return status of the subroutine. The return values are as follows: 1 -- success 2 -- could not release lock when $DEQ called

INITIATION:

Segments of a sample FORTRAN program calling companion subroutine LOCK and this subroutine are shown below:

```
      PROGRAM SAMPLE
              .
              .
              .
      INTEGER*4 LOCK_STATUS(6)
              .
              .
              .
C...
C...  Lock file WD:ARCDT1.DAT for purposes of reading using name
C...  ARCDT as agreed upon by cooperating users. Allow 20.0
C...  seconds to accomplish locking.
C...
      CALL LOCK('ARCDT','PR',20.0,LOCK_STATUS,LOCK_ID,ICALL_STATUS,
     1                 ISUB_STATUS)
      IF (ISUB_STATUS) THEN
            OPEN (UNIT=1,NAME='WD:ARCDT1.DAT',...
      ELSE
            GO TO (error handling routine)
      ENDIF
              .
              .
              .
      CLOSE(UNIT=1)
C...
C...  Release lock.
C...
```

```
CALL UNLOCK(LOCK_STATUS,ICALL_STATUS,ISUB_STATUS)
IF (.NOT. ISUB_STATUS) GO TO (error handling routine)
        .
        .
        .
END
```

CALLING SEQUENCE:

This subroutine should be called after a file (or other system resource) that has been secured by companion subroutine LOCK is no longer being accessed by the calling program.

INITIALIZATION: N/A

REAL TIME CONSIDERATIONS: N/A

AUTHOR: J.R. Froggatt, On-Line Systems Group, ESD, E.I. DuPont

DATE: 7/23/82

CHAPTER 7

** DATA FILES **

7.1 WD:HRAIX1.DAT (Weekly SAM Average File)

GENERAL DESCRIPTION:

File contains hourly SAM averages for a period of 1 week beginning at 23:30:00Z Saturday and extending thru 23:29:59z the following Saturday. Each Hourly SAM average represents four 15 minute periods of the week and consists of the two 15 minute periods prior to the represented hour and the two 15 minute periods following the represented hour. Thus the range of hours represented by the above specified time span is from 00Z Sunday to 23Z the following Saturday. The first record contains header record information and the remaining records contain TV and SRS Tower SAMs averaged over four 15 minute periods. Each data record contains data for a particular 1 hour period beginning with 00z Sunday and extending thru 23Z the following Saturday as indicated above. This file is used every other week. On alternate weeks WD:HRAIX2.DAT, an identically structured file, is used. A general subroutine HIXCLC is used to determine which of these two files and which record in the file corresponds to a particular time expressed as HH:MM:SS MM/DD/YY. Data from this file is used as input to program HRAVPT to generate hourly plume displays at 32 minutes after the hour.
After the file is complete 23:29:59Z Saturday, companion file WD:HRAIX2.DAT is started 23:30:00Z the same day. The data will remain in WD:HRAIX1.DAT until the following Thursday at which time ZEROAC will execute and zero out the file in preparation for the collection of new data beginning the following Saturday. ZEROAC actually clears the file out Thursday, Friday, and Saturday redundantly in case of computer outage during one or two of the scheduled run times. Data must be transferred from the file to mag tape by Procedure SAMTAP for long term storage before the information is zeroed out on Thursday.
Files WD:HRAIX1.DAT and WD:HRAIX2.DAT alternate back and forth on a weekly basis with 1st one collecting data and then the other.

CREATED/UPDATED BY: HRAV, HRAVNP, HRAVCR

READ BY: HRAVPT, PLMDSP, ... Any pgm using Subroutine HRAGET.

FILE TYPE: Indexed/Unformatted

HEADER RCD CONTENT: HDR1: IFIRST, ILAST, Array IHDARR(12), Array RECFLG(11)
(I*2, I*2, 12 I*2, 11 I*2)

IFIRST = Index of 1st record containing data
ILAST = Index of last record containing data
Array IHDARR as follows:
1) Month of first data reading
2) Day "
3) Year "
4) Hour "
5) Min "
6) Sec "
7) Month of last data reading
8) Day "
9) Year "
10) Hour "
11) Min "
12) Sec "

Array RECFLG(11) as follows:
(Each successive bit corresponds to a record in the file)

Bit 0 RECFLG(1) = 1 if any data in file
= 0 if no data in file

Bits 1-15 RECFLG(1):
= 1 if data in corresponding record
= 0 if no data in corresponding record

BitS 0-15 RECFLG(N) (for N = 2 - 10):
= 1 if data in corresponding record
= 0 if no data in corresponding record

Bits 0-9 RECFLG(11):
= 1 if data in corresponding record
= 0 if no data in corresponding record

RECORD FORMAT: RECS 2 - 169:

ITEST	(I*2)
IHR	(I*2)
IMIN	(I*2)
ISEC	(I*2)
IMO	(I*2)
IDAY	(I*2)
IYR	(I*2)
SEC	(R*4)
DIREC	(R*4)
SPEED	(R*4)
SIGAZ	(R*4)
SIGEL	(R*4)

Significance of data as follows:
ITEST: = 0 If no data in record
= 1 If data in record
IHR - IYR: Time _Date of last data collected

SEC: of seconds over which data collected
DIREC - SIGEL: SAM AVERAGE AZM, SPEED(M/S), SIGAZ, SIGEL
OVER THE 60 MINUTE INTERVAL

NUMBER OF RECORDS: 169

SIZE IN BLOCKS: 14

7.2 WD:HRAIX2.DAT (Alt Weekly SAM Average File)

GENERAL DESCRIPTION:

Companion file to WD:HRAIX1.DAT. The description for that file applies to this file.

CREATED/UPDATED BY: HRAV, HRAVNP, HRAVCR

READ BY: Same as WD:HRAIX1.DAT

FILE TYPE: Indexed/Unformatted

HEADER RCD FORMAT: Same as WD:HRAIX1.DAT

RECORD FORMAT: Same as WD:HRAIX1.DAT

NUMBER OF RECORDS: 169

SIZE IN BLOCKS: 14

7.3 WD:ARCDT1.DAT (Weekly WIND Archive Data)

GENERAL DESCRIPTION:

File contains Archive Data for a period of 1 week beginning at 00:00:00Z Sunday and extending thru 23:59:59 Saturday. The first record contains header record information and the remaining records contain TV, Plant, and Climatology tower data averaged over 15 minute periods. Each data record contains data for a particular 15 minute period beginning with 00:00:00 - 00:14:59 Sunday (the first day of the week) and extending thru 23:45:00 - 23:59:59 Saturday. This file is used every other week. On alternate weeks WD:ARCDT2.DAT, an identically structured file, is used. A general subroutine INXCLC is used to determine which of these two files and which record in the file corresponds to a particular time expressed as HH:MM:SS MM/DD/YY. Data from this file is used as input to the Emergency Response Codes as well as to other displays and reports of weather conditions at the plant site. After the file is complete 23:59:59Z Saturday, companion file WD:ARCDT2.DAT is started 00:00:00Z Sunday. the data will remain in WD:ARCDT1.DAT until the following

Thursday at which time ZEROAC will execute and zero out the file in preparation for the collection of new data beginning the following Sunday. ZEROAC actually clears the file out Thursday, Friday, and Saturday redundantly in case of computer outage during one or two of the scheduled run times. Data must be transferred from the file to mag tape by procedure ARCTAP for long term storage before the information is zeroed out on Thursday. Files WD:ARCDT1.DAT and WD:ARCDT2.DAT alternate back and forth on a weekly basis with 1st one collecting data and then the other.

CREATED/UPDATED BY: ARCIV2

READ BY: HRAV, ARCDSP, ... Any pgm using subroutine ARCGET.

FILE TYPE: Direct Access/Unformatted

HEADER RCD CONTENT: HDR1: Array IHDARR(14), Array RECFLG(674)
(14 I*2, 674 L*1)

Array IHDARR as follows:
1) Number of first record containing data
2) Number of last record containing data
3) Month of first data reading
4) Day "
5) Year "
6) Hour "
7) Min "
8) Sec "
9) Month of last data reading
10) Day "
11) Year "
12) Hour "
13) Min "
14) Sec "

Array RECFLG as follows:
RECFLG(1) = 1 If any data in file
= 0 If no data in file
RECFLG(N) (FOR N = 2 - 673)
= 1 If data in corresponding record
= 0 If no data in corresponding record

RECORD FORMAT: RECS 2 - 673:

ITEST	(I*2)
SECNDS	(R*4)
NOOBS	(I*2)
NOTVRD	(I*2)
NOSVRD	(I*2)
NOAPRD	(I*2)
ARRAY TDTBYT(6,2)	(12 B*1)
ARRAY TVBYT(36,6)	(216 B*1)
ARRAY TVINT(36)	(36 I*2)
ARRAY TVSP(36,8)	(288 R*4)
ARRAY SVBYT(36,6)	(216 B*1)
ARRAY SVINT(36)	(36 I*2)
ARRAY SVSP(36,8)	(288 R*4)

ARRAY APSP(10,3)	(30 R*4)
NOTPRD	(I*2)
ARRAY TPBYT(36,6)	(216 B*1)
ARRAY TPINT(36)	(36 I*2)
ARRAY TPSP(36,8)	(288 R*4)
NOCLRD	(I*2)
CLBYT(72,6)	(432 B*1)
CLINT(72)	(72 I*2)
CLSP(72,8)	(576 R*4)

Significance of data as follows:

ITEST: = 0 If no data in record
= 1 If data in record

SECNDS: of seconds over which data collected in the 15 min period

NOOBS: Number of data reading passes made in the 15 min period

NOTVRD: Number of TV Twr channels (currently 32)

NOSVRD: Number of SV Twr channels (currently 24) This remains 24 instead of 27 with L area since algorithms might assume contiguous storage of "like" variables (eg. speed).

NOAPRD: Number of Airport data fields (Currently 8)

TDTBYT(6,2):

1st Subscript:

1) Hour
2) Min
3) Sec
4) Month
5) Day
6) Year

2nd Subscript:

1) 1st data collection time
2) Last data collction time

For the 3 TV tower data arrays, significance of 1st subscript depends upon whether scalar or vector values are involved.

Significance of 1st subscript when vector values involved:

1 - 8) Unused
9) U, V Vectors - 18 M
10) " - 36 M
11) " - 91 M
12) " - 137 M
13) " - 182 M
14) " - 243 M
15) " - 304 M
16) Unused
17) Sum U Vectors - 18 M
18) " - 36 M
19) " - 91 M
20) " - 137 M
21) " - 182 M
22) " - 243 M

23) " - 304 M
24) Unused
25) Sum V Vectors - 18 M
26) " - 36 M
27) " - 91 M

28) " - 137 M
29) " - 182 M
30) " - 243 M
31) " - 304 M
32 - 36) Unused

Significance of 1st subscript when scalar values involved (Array indices = Channel + 1):

1) TV Temperature - 2 M
2) " - 18 M
3) " - 36 M
4) " - 91 M
5) " - 137 M
8) " - 182 M
6) " - 243 M
7) " - 304 M
9) TV Speed - 18 M
10) " - 36 M
11) " - 91 M
12) " - 137 M
13) " - 182 M
14) " - 243 M
15) " - 304 M
16) Spare channel
17) TV Azimuth - 18 M
18) " - 36 M
19) " - 91 M
20) " - 137 M
21) " - 182 M
22) " - 243 M
23) " - 304 M
24) Spare channel
25) TV Elevation - 18 M
26) " - 36 M
27) " - 91 M
28) " - 137 M
29) " - 182 M
30) " - 243 M
31) " - 304 M
32) Spare channel
33 - 36) Unused

TVBYT(36,6): Array containing Quality Assurance flags for TV tower

2nd subscript:

1) U, V Vector component sum Q/A flags (vector 1st subscript) (** future **)
2) Maximum value Q/A flags (** future **) (scalar 1st subscript)

3) Mean value Q/A flags
(scalar 1st subscript)
4) Std Dev Q/A flags
(scalar 1st subscript)
5) General Q/A flags (Kern Algorithm)
(scalar 1st subscript)
6) Unused

TVINT: Array containing of data points read
(scalar subscripts)

TVSP(36,8): Array containing floating point values associated with TV Tower

2nd subscript:

1) Original 15 min U,V vector sums (vector)
2) " maximum value (scalar)
3) " mean value (scalar)
4) " standard deviation (scalar)
5) Q/A 15 min U, V vector components (vector)
6) " maximum value (scalar)
7) " mean value (scalar)
(Corrected mean value speed and azm same as corrected vector speed and azm)
8) " Std Deviation (scalar)

For the 3 plant tower non temperature data arrays, significance of 1st subscript depends upon whether scalar or vector values are involved.
Significance of 1st subscript when vector values involved: (*** NOTE L AREA DATA LOCATIONS ***)

1) U, V Vector components - A
2) " - C
3) " - D1 (36 M)
4) " - D2 (61 M)
5 " - F
6) " - H
7) " - K
8) " - P
9) Sum U Vectors - A
10) " - C
11) " - D1 (36 M)
12) " - D2 (61 M)
13) " - F
14) " - H
15 " - K
16) " - P
17) Sum V Vectors - A
18) " - C
19) " - D1 (36 M)
20) " - D2 (61 M)
21) " - F
22) " - H
23) " - K
24) " - P
25) U, V Vector components - L
26) Sum U Vectors - L

27) Sum V Vectors - L
28 - 36) Unused

Significance of 1st subscript when scalar values involved (Array indices = Channel + 1):
(*** NOTE L AREA DATA LOCATIONS ***)

1) Wind speed - A
2) " - C
3) " - D1 (36 M)
4) " - D2 (61 M)
5) " - F
6) " - H
7) " - K
8) " - P
9) Azimuth - A
10) " - C
11) " - D1 (36 M)
12) " - D2 (61 M)
13) " - F
14) " - H
15) " - K
16) " - P
17) Elevation - A
18) " - C
19) " - D1 (36 M)
20) " - D2 (61 M)
21) " - F
22) " - H
23) " - K
24) " - P
25) Wind speed - L
26) Azimuth - L
27) Elevation - L
28 - 36) Unused

SVBYT(36,6): Array containing Quality Assurance Flags for SV towers

2nd subscript:

1) U, V Vector component sum Q/A flags (vector 1st subscript) (** future **)
2) Maximum value Q/A flags (** future **) (scalar 1st subscript)
3) Mean value Q/A flags (scalar 1st subscript)
4) Std Dev Q/A flags (scalar 1st subscript)
5) General Q/A flags (Kern Algorithm) (scalar 1st subscript)
6) Unused

SVINT(36): Array containing of data points read (scalar subscripts)

SVSP(36,8): Array containing floating point values associated with Plant towers

2nd subscript:

1) Original 15 min U,V vector sums (vector)
2) " maximum value (scalar)

3) " mean value (scalar)
4) " standard dev (scalar)
5) Q/A 15 min U, V vector sums (vector)
6) " maximum value (scalar)
7) " mean value (scalar)
(corrected mean value
speed and azimuth same
as corrected vector
speed and azimuth)
8) " standard dev (scalar)

APSP(10,3): Array to contain up to 10 values
for up to 3 Airports. Currently
6 values are being stored
for Bush Field.
1,1) Bush Field direction
2,1) " speed (knots)
3,1) " temp (deg F)
4,1) " dew pt temp (deg F)
5,1) " stability
6,1) " hour (Zulu time)
7,1) " sea level pressure
8,1) " visibility
9,1 - 10,1) Bush Field spares
1,2 - 10,2) 10 Airport 2 spares
1,3 - 10,3) 10 Airport 3 spares

NOTPRD: Number of plant temperature channels
(currently 24). This remains 24 instead of 27
with L area since algorithms might assume
contiguous storage of "like" variables (eg. dew point).
For the 3 plant tower temperature data arrays,
significance of 1st subscript is as follows:
(Array indices = Channel + 1)
(*** NOTE L AREA DATA LOCATIONS ***)
1) Temperature (61M) - A
2) " - C
3) Unused
4) Temperature (61M) - D
5) " - F
6) " - H
7) " - K
8) " - P
9) Dew Point (61M) - A
10) " - C
11) Unused
12) Dew Point (61M) - D
13) " - F
14) " - H
15) " - K
16) " - P
17) Temperature (2M) - A
18) " - C
19) Unused
20) Temperature (2M) - D
21) " - F
22) " - H
23) " - K

24) " - P
25) Temperature (61M) - L
26) Dew Point (61M) - L
27) Temperature (2M) - L
28 - 36) Unused

TPBYT(36,6): Array containing Quality Assurance Flags for plant temps
2nd subscript:
1) unused
2) Maximum value Q/A flags (** future **)
3) Mean value Q/A flags (** future **)
4) Std Dev Q/A flags (** future **)
5) General Q/A flags (Kern Algorithm)
6) Unused

TPINT(36): Array containing of data points read

TPSP(36,8): Array containing floating point values associated with Plant towers
2nd subscript:
1) Unused
2) Original 15 min maximum value
3) " mean value
4) " standard dev
5) Unused
6) Q/A 15 min maximum value (** future - curr. same as original values)
7) " mean value (curr. same as orig.)
8) " standard dev (curr. same as orig.)

NOCLRD: Number of CL Twr channels (currently 40)

For the 3 CL tower data arrays, significance of 1st subscript depends upon whether scalar or vector values are involved.
Significance of 1st subscript when vector values involved:
1) Sum U Vectors - 2 M
2) " - 18 M
3) " - 36 M
4) " - 61 M
5) Sum V Vectors - 2 M
6) " - 18 M
7) " - 36 M
8) " - 61 M
9) # U, V Vectors - 2 M
10) " - 18 M
11) " - 36 M
12) " - 61 M
13 - 72) Unused

Significance of 1st subscript when scalar values involved (Array indices = Channel + 1):
1) CL Azimuth - 2 M
2) " - 18 M
3) " - 36 M
4) " - 61 M
5) CL Elevation - 2 M

6) " - 18 M
7) " - 36 M
8) " - 61 M
9) CL AZ Speed - 2 M
10) " - 18 M
11) " - 36 M
12) " - 61 M
13) CL EL Speed - 2 M
14) " - 18 M
15) " - 36 M
16) " - 61 M
17) CL Dew Point - 2 M
18) " - 18 M
19) " - 36 M
20) " - 61 M
21) CL Fast Temperature - 2 M
22) " - 18 M
23) " - 36 M
24) " - 61 M
25) CL Reference Temperature - 2 M
26) CL Delta Temperature - 18 M
27) CL Delta Temperature - 36 M
28) CL Delta Temperature - 61 M
29) CL Soil Temperature - 1"
30) CL Soil Temperature - 6"
31) CL Soil Temperature - 12"
32) CL Pyrhel
33) CL Eppley B _W Radiation
34) CL Net Radiation
35) CL Water Temp (Evaporation)
36) CL Up Total Radiation
37) CL Barometric Pressure
38) CL Down Total Radiation
39) CL Actual Evaporation
40) CL Precipitation 1 (Rain Guage)
41 - 72) Unused

CLBYT(72,6): Array containing Quality Assurance flags for CL tower
2nd subscript:
1) U, V Vector component sum Q/A flags (vector 1st subscript) (** future **)
EXCEPTIONS: For first subscripts 37, 39, and 40 (scalar), the value contained is the first reading in the 15 minute period.
2) Maximum value Q/A flags (** future **) (scalar 1st subscript)
EXCEPTIONS: For first subscripts 37, 39, and 40 (scalar), the value contained is the last reading in the 15 minute period.
3) Mean value Q/A flags (** future **) (scalar 1st subscript)

4) Std Dev Q/A flags (** future **)
(scalar 1st subscript)
5) General Q/A flags (Kern Algorithm)
(scalar 1st subscript)
6) Unused

CLINT(72): Array containing of data points read
(scalar subscripts)

CLSP(72,8): Array containing floating point
values associated with CL Tower
2nd subscript:
1) Original 15 min U,V vector sums (vector)
2) " maximum value (scalar)
3) " mean value (scalar)
4) " standard deviation (scalar)
5) Q/A 15 min U, V vector components (vector)
(** future - curr.
same as original
values)
6) " maximum value(curr. same as orig.)
7) " mean value (curr. same as orig.)
8) " standard dev (curr. same as orig.)

NUMBER OF RECORDS: 673

SIZE IN BLOCKS: 10095

7.4 WD:ARCDT2.DAT (Alternate Weekly WIND Archive Data)

GENERAL DESCRIPTION:

Companion file to WD:ARCDT1.DAT. The description
for that file applies to this file.

CREATED/UPDATED BY: ARCIV2

READ BY: Same as WD:ARCDT1.DAT

FILE TYPE: Direct Access/Unformatted

HEADER RCD FORMAT: Same as WD:ARCDT1.DAT

RECORD FORMAT: Same as WD:ARCDT1.DAT

NUMBER OF RECORDS: 673

SIZE IN BLOCKS: 10095

7.5 WD:ARCDT3.DAT (WIND Archive Tape Restore File)

GENERAL DESCRIPTION:

File is used to contain Archived Data restored from tape covering a period of 1 week beginning at 00:00:00 Sunday and extending thru 23:59:00 Saturday. Data is inserted into the file by Procedure ARCTAP. The data can then be accessed in a manner similar to that used to retrive data from either WD:ARCDT1.DAT or WD:ARCDT2.DAT.

CREATED/UPDATED BY: WC:ARCTAP.COM

READ BY: ARCDSP

FILE TYPE: Direct Access/Unformatted

HEADER RCD FORMAT: Same as WD:ARCDT1.DAT

RECORD FORMAT: Same as WD:ARCDT1.DAT

NUMBER CF RECORDS: 673

SIZE IN BLOCKS: 10095

END

DATE
FILMED
6/10/93