DATA COLLECTION FOR THE HAZARDOUS WASTE IDENTIFICATION RULE

SECTION 4.0 METEOROLOGICAL DATA

Work Assignment Manager and Technical Direction:

Stephen M. Kroner
David A. Cozzie
U.S. Environmental Protection Agency
Office of Solid Waste
Washington, DC 20460

Prepared by:

Center for Environmental Analysis
Research Triangle Institute
3040 Cornwallis Road
Research Triangle Park, NC 27709-2194
Under Contract No. 68-W6-0053

U.S. Environmental Protection Agency
Office of Solid Waste
Washington, DC 20460

October 1999

ACKNOWLEDGMENTS

This work was performed by the Research Triangle Institute (RTI) under U.S. Environmental Protection Agency (EPA) contract 68-W-98-085 with the Office of Solid Waste. Stephen Kroner, the U.S. EPA Work Assignment Manager, provided overall technical direction and review throughout this work. Terry Pierson, the RTI Work Assignment Leader, along with Robert Truesdale, leader of the data collection task, provided day-to-day management and technical direction at RTI.

A number of individuals have been involved in the development and implementation of the data collection methodologies and computer programs described herein. Paul Andrews helped to direct and document the overall effort and was responsible for quality control. Chengwei Yao, Amar Trivedi, Wayne Winstead, and Ed Rickman assisted with methodology development and programming. Linda Andrews, Susie Tyndall, and Hilary Solomon assisted with document preparation, with Kathy Restivo serving as technical editor and Keith Little and Robert Truesdale providing senior technical review and input. Cindi Salmons was the quality assurance officer.

DISCLAIMER

Table of Contents

Section **Page**

Table of Contents (continued)

Appendices

List of Figures

Number **Page**

List of Tables

Number **Page**

List of Acronyms

3MRA	multimedia, multiple-exposure pathway, multiple-receptor risk assessment
ASCII	American Standard Code for Information Interchange
DOC	U.S. Department of Commerce
DOE	U.S. Department of Energy
EPA	U.S. Environmental Protection Agency
FSL	Forecast Systems Laboratory
GIRAS	Geographic Information Retrieval and Analysis System
GIS	geographic information system
HWIR	Hazardous Waste Identification Rule
ISCST3	Industrial Source Complex Short Term, Version 3, air model
ISMCS	international station meteorological climate summaries
LAU	land application unit
MUSLE	modified universal soil loss equation
NCDC	National Climatic Data Center
NOAA	National Oceanic and Atmospheric Administration
NWS	National Weather Service
QA/QC	quality assurance/quality control
RCRA	Resource Conservation and Recovery Act
SAMSON	Solar and Meteorological Surface Observation Network
SCRAM	Support Center for Regulatory Air Models
USAF	U.S. Air Force
USGS	U.S. Geological Survey
USN	U.S. Navy
USLE	universal soil loss equation
WMU	waste management unit

4.0 Meteorological Data

The U.S. Environmental Protection Agency (EPA) designed the Hazardous Waste Identification Rule (HWIR) risk assessment to estimate potential risks from the long-term management of HWIR waste by waste management facilities typically expected to handle exempted waste: Resource Conservation and Recovery Act (RCRA) Subtitle D nonhazardous industrial waste management units (WMUs). It employs an integrated, multimedia, multiple-exposure pathway, and multiple-receptor risk assessment model (the 3MRA model) to evaluate risks that may occur from the long-term, multimedia release of a chemical from these WMUs, as represented by the 201 Industrial Subtitle D (Industrial D) facilities selected for the analysis.

Of the 18 media-specific pollutant release, fate, transport, exposure, and risk modules in the 3MRA model, meteorological data are used directly by the source, air, watershed, and waterbody modules. Although 3MRA model risk estimates are long-term estimates, the meteorological data timescales differ by component module, with shorter timescales necessary to accurately estimate release or fate and transport in media sensitive to fluctuations in meteorological data. For example, the surface impoundment and tank modules need monthly data to capture temperature extremes that can impact volatilization. The wastepile, land application unit (LAU), and watershed modules require daily data to accurately estimate precipitation-driven runoff and erosion events.

During system execution, the HWIR 3MRA modeling system calls meteorological data for the appropriate meteorological station directly as separate American Standard Code for Information Interchange (ASCII) files containing hourly, daily, monthly, annual, and long-term meteorological data. Table 4-1 shows which of the 3MRA modules access which file types.[1]

Meteorological data were collected regionally by meteorological station, with each of the 201 Industrial D sites modeled for HWIR assigned to the nearest station with similar weather conditions and adequate weather data for the analysis. In making these assignments, EPA considered all available data from 218 meteorological stations across the United States to find the best data for each site. This process resulted in 99 meteorological stations being matched to the 201 industrial sites. This is a considerable improvement from the 29 meteorological stations used to represent the nation in the 1995 HWIR proposal and was made possible by several automated programs developed to reduce the effort needed to prepare the data files needed by the 3MRA model.

[1] Note that the data files read by the air module include hourly data, the twice-daily upper air data, and long-term meteorological station-related data such as surface roughness length, friction velocity, and Monin-Obukhov length.

Table 4-1. Meteorological Data Timescale, by HWIR 3MRA Model Module

HWIR Module	Hourly	2X Daily	Daily	Monthly	Annual	Long-term
Air	!	!				!
Wastepile			!	!	!	!
Land application unit			!	!	!	!
Surface impoundment				!		
Aerated tank				!		
Landfill			!		!	!
Watershed			!	!		!
Surface water						!

This remainder of this section describes this meteorological data collection effort, including the parameters collected (Section 4.1), data sources (Section 4.2), methodology (Section 4.3), results (Section 4.4), issues and uncertainties (Section 4.5), and references (Section 4.6).

4.1 Parameters Collected

Meteorological data were used by all five source models and all media models, except for ground water. Table 4-2 shows the meteorological inputs needed, by model component. Meteorological data were collected regionally by surface and upper air meteorological stations. Surface data were needed for five different timescales: hourly, daily, monthly, annual, and long-term annual average. Upper air mixing height data were collected twice daily, but passed to the system in the hourly data files. Table 4-3 shows the meteorological parameters, by output file.

4.2 Data Sources

Most meteorological data were extracted from Solar and Meteorological Surface Observation Network (SAMSON) hourly data files and converted as necessary to daily time series, monthly time series, annual time series, and long-term averages. Because SAMSON precipitation data were inadequate, precipitation data were obtained from cooperative station daily summaries, with SAMSON data used to help allocate these daily data to hourly time series. Mixing heights were obtained from upper air station data. Land use data also were required in the vicinity of each meteorological station. Sources for these data are described in this section.

4.2.1 Surface Meteorological Data: SAMSON

Surface data were collected from SAMSON CD-ROMs (U.S. Department of Commerce [DOC] and U.S. Department of Energy [DOE], 1993). SAMSON contains data from 1961 through 1990 for 218 meteorological facilities throughout the United States. The following

Table 4-2. Meteorological Inputs, by Model Component

Model Input	Source Models					Media Models		
	Aerated Tank	Land-fill	LAU	Surface Imp.	Waste-pile	Air (ISCST3)	Surface Water	Water-shed
Anemometer height (surface station)						!		
Cloud cover (opaque, hourly, surface data)						!		
Cloud cover (annual average, long-term)							!	
Cloud ceiling height (hourly, surface data)						!		
Evaporation (daily)		!	!					!
Evaporation (average monthly)	!			!				
Evaporation (long-term annual average)							!	
Fraction of net radiation (ground surface)						!		
Mixing height (afternoon/morning, daily, upper air data)						!		
MUSLE rainfall erosivity (daily)			!		!			!
Precipitation days/yr with >0.25 mm (mean annual)			!		!			
Precipitation days/yr with >0.25 mm (long-term average)		!						
Precipitation amount and flag (hourly, surface data)						!		
Precipitation (daily)		!	!		!			!
Precipitation (average monthly)	!			!				
Precipitation (long-term annual average)							!	
Present weather (hourly, surface data)						!		
Station pressure (hourly, surface data)						!		
Temperature (hourly, dry bulb, surface data)						!		
Temperature (daily air)		!	!		!			!
Temperature (mean monthly air)	!			!				
Temperature (maximum monthly air)		!	!					!
Temperature (minimum monthly air)		!	!					!
Temperature (long-term annual average air)							!	
Temperature (soil column)			!	!				!
Thornthwaite precipitation evaporation index (PE)		!	!					!
Wind direction (hourly, surface data)						!		
Windspeed (hourly, surface data)						!		
Windspeed (mean monthly)	!			!				
Windspeed (mean annual)			!		!			
Windspeed (long-term annual average)		!					!	
Windspeed: % time >5.4 m/s @ mean pile height					!			
Wind, fastest mile of (mean annual)			!					
Wind, fastest mile of (long-term average)		!						
Julian day of meteorological data		!	!		!			!
Time zone (surface data)						!		
Year of surface meteorological data						!		
Year of upper air meteorological data						!		
Friction velocity						!		
Minimum Monin-Obukhov length						!		
Surface roughness length						!		

MUSLE = modified universal soil loss equation

Table 4-3. Meteorological Parameters Collected by Data File

Meteorological Parameter	HWIR 3MRA Meteorological Data File				
	h.dat	d.dat	m.dat	a.dat	l.dat
Surface roughness length	!				
Minimum Monin-Obukhov length	!				
Friction velocity	!				
Precipitation code	!				
Stability category	!				
Cloud cover		!	!		!
Temperature	!	!	!	!	!
Wind direction	!				
Windspeed	!	!	!	!	!
Precipitation	!	!	!	!	!
Evaporation		!	!	!	!
Windspeed at pile height		!			
Number hours wind > 5.4 m/s at pile height		!			
% time wind > 5.4 m/s at pile height				!	
Thornthwaite precipitation-evaporation index (PE)				!	
Days with precipitation > 0.01 in.				!	!
Fastest mile of wind				!	!
USLE rainfall erosivity factor		!			!
Maximum/minimum temperature			!		
Temperature of soil column					!
Mixing height	!				

USLE = universal soil loss equation

hourly data were downloaded for use in the HWIR Industrial Source Complex Short Term, Version 3 (ISCST3) air model and used to calculate daily, monthly, annual average, and long-term climatological data: opaque cloud cover, dry bulb temperature, station pressure, wind direction, windspeed, ceiling height, present weather, and hourly precipitation amount.

4.2.2 Surface Meteorological Data: Cooperative Summaries

During QC checks, it was discovered that SAMSON files were missing significant amounts of precipitation data. To replace the SAMSON data, daily precipitation totals were collected from a set of CD-ROM data called the cooperative summary of the day (National Climatic Data Center [NCDC] and National Weather Service [NWS], 1995). These summaries contain several meteorological parameters, including daily precipitation, collected between the 1850s and 1993 at cooperating meteorological stations throughout the United States. These daily precipitation data were used to prepare both hourly meteorological files and long-term climatological files.

4.2.3 Upper Air Mixing Height Data

Twice-daily mixing height data were calculated from upper air data contained in the radiosonde data of North America CD-ROM set (Forecast Systems Laboratory [FSL] and NCDC, 1997). This set contains upper air data from 1946 through 1996 for most upper air stations in the United States. These data were combined with the SAMSON data to create the mixing height files used in the ISCST3 air model using the data processing program PCRAMMET. EPA's Support Center for Regulatory Air Models (SCRAM) bulletin board (U.S. EPA, 1996) was used to obtain 1990 mixing height data (if available) when mixing height data could not be successfully calculated from the radiosonde data.

4.2.4 Land Use Data

Land use parameters for PCRAMMET input were based on Geographic Information Retrieval and Analysis System (GIRAS) data. GIRAS provides comprehensive land use data, in digital geographic information system (GIS) format, for the conterminous United States (U.S. Geological Survey [USGS], 1990). The spatial data set used for this analysis represents digital data originally collected by the USGS and then converted into the ARC/INFO GIS format by EPA (U.S. EPA, 1994). These digital coverages, available from EPA by one-degree quadrangles (1:250,000 scale), are based on Anderson land use codes (Anderson et al., 1976). The full metadata record may be found online at http://nsdi.epa.gov/nsdi/projects/giras.htm.

Following is a list of PCRAMMET inputs whose values were based on land use derived from GIRAS:

- Anthropogenic heat flux
- Bowen ratio
- Minimum Monin-Obukhov length
- Noontime albedo
- Fraction of net radiation absorbed by the ground
- Surface roughness length.

In addition, anemometer heights were taken from the annual summaries of local climatological data (National Oceanic and Atmospheric Administration [NOAA], 1983).

Anemometer height was used to process hourly meteorological files, as well as certain climatological parameters.

4.3 Methodology

Meteorological data were collected and processed by NOAA meteorological stations. The first data collection step was to assign each of the 201 Industrial D sites to a meteorological station. Then, data for the assigned meteorological stations were processed through a series of programs designed to prepare the data files for the HWIR 3MRA modeling system. Figure 4-1 depicts the data preparation process conducted for each meteorological station to which Industrial D sites were assigned.

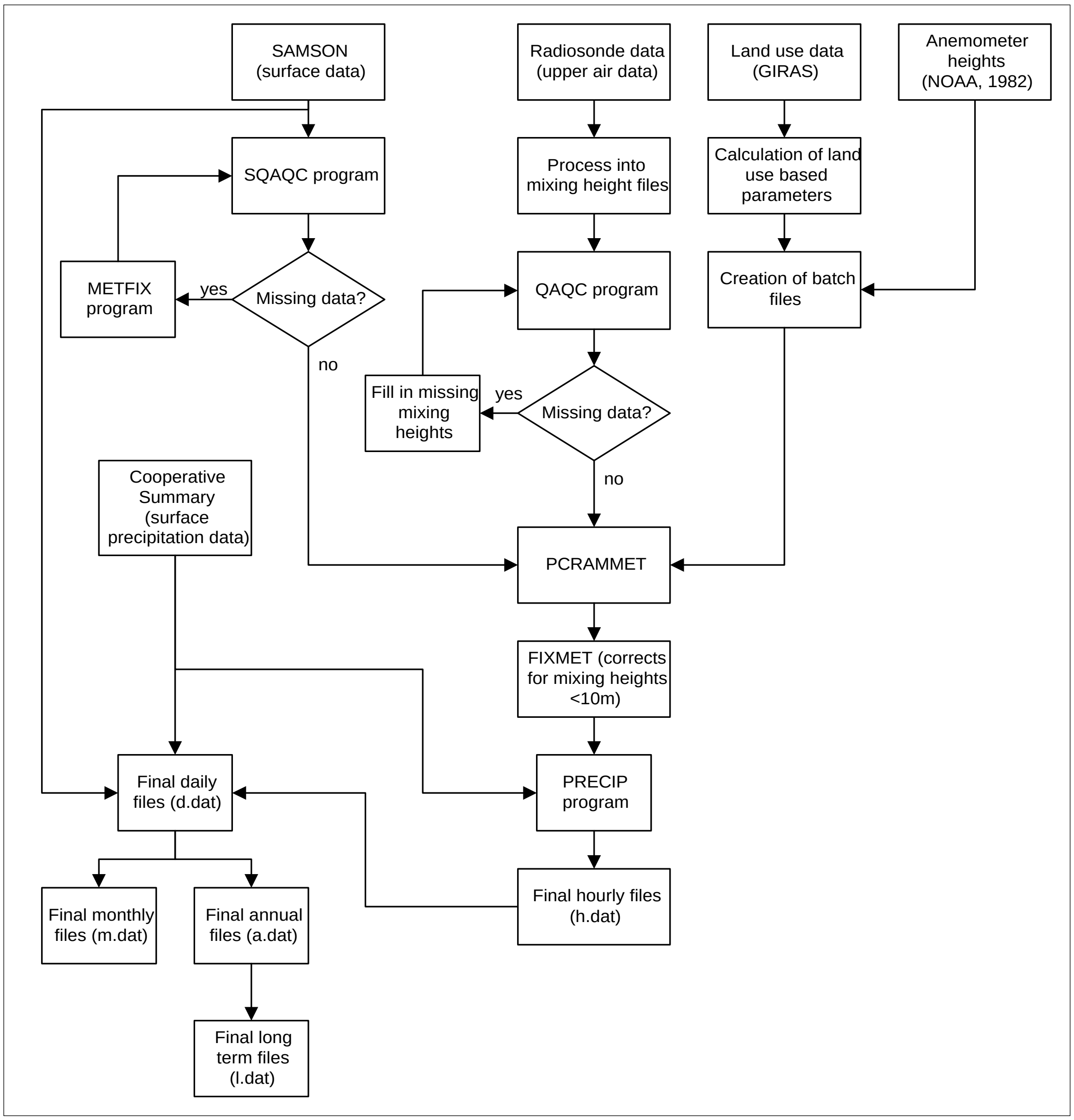

Figure 4-1. HWIR meteorological data collection process.

4.3.1 Meteorological Station Assignment

To ensure that Industrial D facilities were modeled with the correct meteorological data, sites were first assigned to the meteorological data stations most representative of conditions at meteorological stations with adequate surface data and to divide the country into an equal number of Thiessen polygons surrounding each station, whose boundaries are equidistant from the site. As a first step in this assignment, a GIS was used to produce a map containing all 199 adjacent meteorological stations. Next, a meteorologist conflated the polygon's boundaries based on climatic and physiographic regions of the country. Data from each meteorological station were considered representative of any Industrial D facility that fell within the surrounding conflated polygon.

Conflation was accomplished, in part, by studying the divisions established by Bailey's ecoregions (Bailey et al., 1994), which are defined primarily on physiography and climate (e.g., the marine regime redwood province for Seattle, WA, and Portland, OR). Figure 4-2 shows the 199 meteorological stations in the continental United States, conflated polygons for each meteorological station, Bailey's ecoregions, and the locations of the 201 Industrial D facilities modeled for HWIR.

To provide the necessary mixing height data for air modeling, a similar method was used to assign upper air meteorological stations to surface meteorological stations. Thiessen polygons were drawn around upper air stations using a GIS. These polygons were adjusted based on the climatic and topographic regimes in which the upper air station lies. Any surface station falling within an upper air station's polygon was assigned that upper air station.

In a few cases, adjustments were made by a meteorologist when it was discovered that at least 10 complete years of matching surface and upper air data were not available. These adjustments were made based on the proximity of the facility in question to the surrounding surface stations and the climatic representativeness of the nearby meteorological stations.

4.3.2 Hourly Surface Data Collection

Hourly surface data were collected from the NCDC's SAMSON CD-ROM. SAMSON contains 30 yr of meteorological data, beginning in 1961. These data were downloaded in their complete form from SAMSON, and the items needed for air modeling were extracted and put into the necessary format for PCRAMMET (U.S. EPA, 1995b) by a computer algorithm. It was found in all cases that significant data were missing, usually in the last half of the 1960s; in some cases, the missing data extended into the 1970s. In these cases, complete observations were available from SAMSON for only a third of the hours in the file. The years in which this occurred were eliminated on a site-by-site basis.

Identification of these missing data was accomplished using a program called SQAQC (see Appendix 4A), which searched for incidents of missing data on the observation indicator, opaque cloud cover, temperature, station pressure, wind direction and speed, and ceiling height. Years that were missing 10 percent or more of the data were discarded (Atkinson and Lee, 1992). Verification (QC) checks were performed on the SQAQC program by applying it to station data

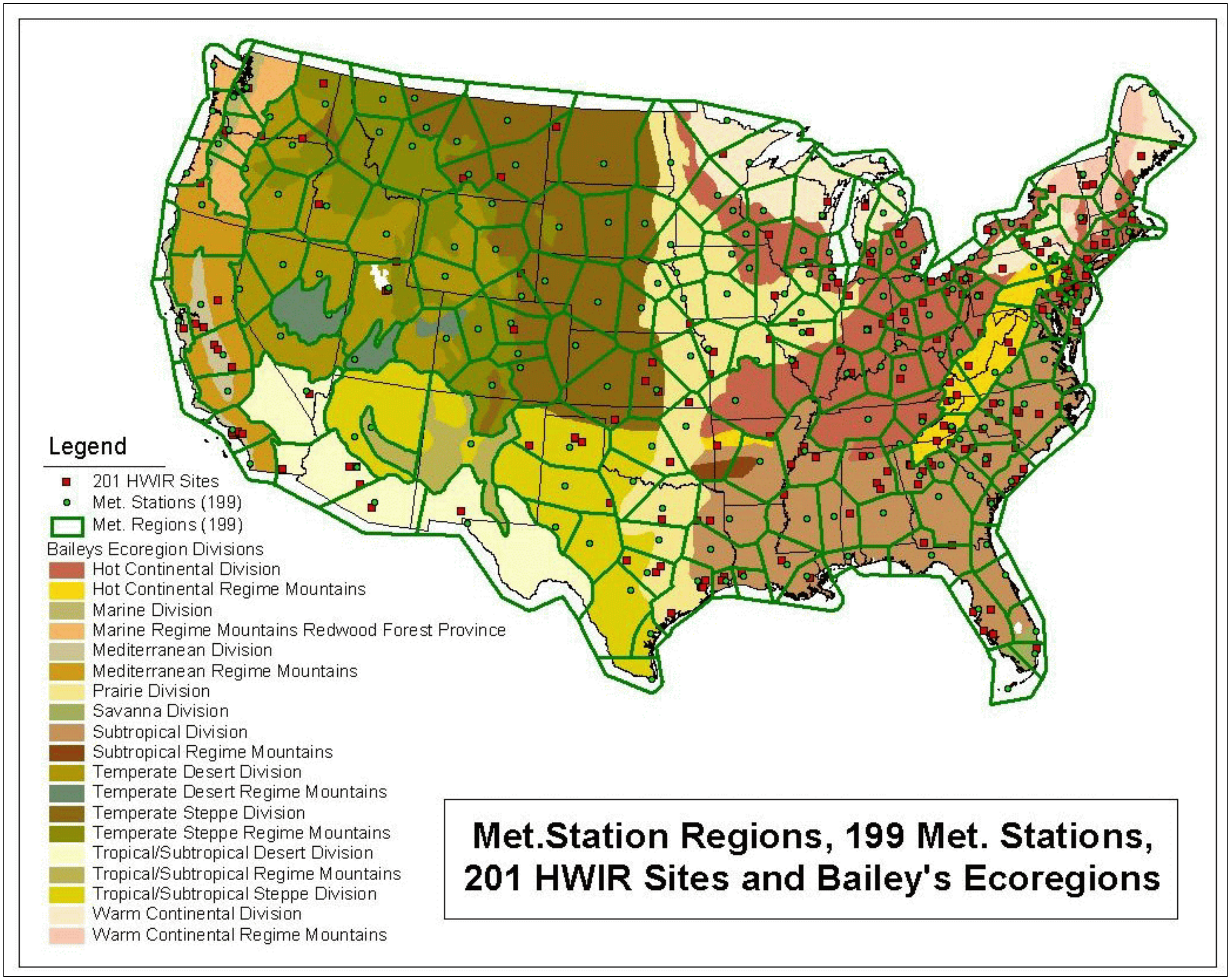

Figure 4-2. Meteorological station assignments for the 201 Industrial D facilities.

where the missing data were known and by intentionally degrading surface meteorological files and then running SQAQC to detect the missing values (see Appendix 4A).

Missing surface data were filled in by METFIX. Appendix 4B summarizes the procedures and provides the source code for this program. METFIX fills in up to five consecutive hours of data for cloud cover, ceiling height, temperature, pressure, wind direction, and windspeed. For single missing values, the program follows the objective procedures developed by Atkinson and Lee (1992). For one to five consecutive missing values other rules were developed because the subjective methods provided by Atkinson and Lee (1992) rely on professional judgment and could not be programmed. The METFIX program flagged files where missing data exceeded five consecutive values. In the few cases where this occurred and the missing data did not constitute 10 percent of the file, they were filled manually according to procedures set forth in Atkinson and Lee (1992). If more than 10 percent of the data were missing, the station was discarded and another nearby station was selected to represent the site.

4.3.3 Hourly Upper Air Data Collection

Twice-daily mixing heights were needed to construct the hourly data necessary to run ISCST3 (U.S. EPA, 1995a). Mixing-height data were not available from EPA's SCRAM bulletin board for more than 5 or 6 yr, in most cases. As a result, these data were constructed using a series of three programs obtained from the Andrews (1998). The first program reformatted the upper air radiosonde observations, which were gathered from the radiosonde data of North America CD-ROM (FSL and NCDC, 1997). These data were usually available for most of the 30-yr period of interest. The second program reformatted the surface meteorological files, which were needed for the surface temperature. These were gathered from SAMSON. Finally, the third program calculated the twice-daily mixing heights. This program required the input of an increment to add to the minimum daily temperature, to calculate the morning mixing height. Five degrees Celsius was chosen based on Holzworth (1972).

All upper air files were checked for missing data using a program called QAQC (see Appendix 4A). QAQC produces a log file containing occurrences of missing mixing height. Verification (QC) checks were performed on the QAQC program by applying it to station data where the missing data were known and by intentionally degrading existing mixing height files and then running QAQC to detect the missing values (see Appendix 4A).

Missing mixing heights were filled in, where possible, by running the files through another program written to interpolate one to five consecutive missing values. According to Atkinson and Lee (1992), if there are one to five consecutive missing values, the values should be filled in subjectively using professional judgement. Again, programming these subjective procedures was not feasible, and the program used simple linear interpolation to fill in these values automatically. Atkinson and Lee (1992) was used to determine which files should be discarded (i.e., files missing more than five consecutive missing values or missing 10 percent or more of the data). After the missing mixing heights were filled in for all upper air files, they were checked once more for missing data using the QAQC program.

4.3.4 Precipitation Data

QC of the meteorological data sources showed that SAMSON precipitation data were not adequate. Long-term average precipitation amounts calculated from SAMSON fell significantly short of the long-term values provided in the international station meteorological climate summaries (ISMCS; U.S. Navy [USN], U.S. Air Force [USAF], and DOC, 1992). Reliable daily precipitation totals were available from the CD-ROM set called the cooperative summary of the day (NCDC, ERL, and NWS, 1995). A program (PRECIP) was developed to disaggregate and distribute the daily data to an hourly basis to create the hourly time series data required for air modeling. Using the available SAMSON data as a template to identify hours when rain occurred at a station, PRECIP distributed the daily precipitation data across these hours according to a set of rules described in Appendix 4C. For example, when the cooperative files indicated rain and SAMSON did not, the rain was spread evenly across the entire day. When SAMSON data indicated precipitation, the daily data were spread across the hours with rain in proportion to the amounts in the SAMSON file. Other details of the PRECIP program are provided in Appendix 4C.

QC checks of PRECIP were performed in two ways. First, printouts of cooperative station data from selected stations were compared to the processed hourly meteorological files by randomly selecting several days throughout the year that contained different situations (e.g. zero precipitation, trace precipitation, measured amount). These numbers were compared to the totals from the matching day in the hourly meteorological file. A second QC effort focused on all stations that were processed. The annual sum of the cooperative station precipitation data was compared to the annual sum of the new hourly meteorological file using a program called RAINTOT. The results from both sources were almost always within 1 cm of each other for each year. In most cases where the results were a few centimeters off they were still within 2 percent of a perfect match. Only two meteorological stations of the 99 used in this analysis had slightly below 95 percent-complete hourly precipitation data by this method. One year from one of these stations was discarded because entire months of data were found to be missing.

4.3.5 Land Use-Related Parameters

The program used to prepare meteorological data files for PCRAMMET required certain land use-related parameters for each meteorological station for which data were to be processed. To develop these parameters, a GIS was used to determine the land use within a 3-km radius around each meteorological station by using GIRAS spatial data with Anderson land use codes (Anderson et al., 1976). To determine land use-related parameters around each station, Anderson land use codes were related to PCRAMMET land use codes, as shown in Table 4-4.

A weighted average, based on the land use percentages for a 3-km radius around each meteorological station, was used to calculate the minimum Monin-Obukhov length, the roughness height at the meteorological station, the noontime albedo, the Bowen ratio, and the fraction of net radiation absorbed by the ground. The roughness height was assumed to be the same at the meteorological station and at the application site in order to avoid creating a separate meteorological input file for every Industrial D site modeled. The anthropogenic heat flux was assumed to be zero because of a lack of data at different meteorological stations. Anemometer

Table 4-4. Relation Between Anderson Land Use Codes and PCRAMMET Land Use Codes

Anderson Code and Description		RAMMET Type and Description	
51	Streams and canals	1	Water surface
52	Lakes	1	Water surface
53	Reservoirs	1	Water surface
54	Bays and estuaries	1	Water surface
41	Deciduous forest land	2	Deciduous forest
61	Forested wetland	2	Deciduous forest
42	Evergreen forest land	3	Conifuerous forest
43	Mixed forest land	4	Mixed forest
62	Nonforested wetland	5	Swamp (non forested)
84	Wet tundra	5	Swamp (non forested)
21	Cropland and pasture	6	Agricultural
22	Orchards-groves-vineyards-nurseries-ornamental	6	Agricultural
23	Confined feeding operations	6	Agricultural
24	Other agricultural land	6	Agricultural
31	Herbaceous rangeland	7	Rangeland (grassland)
32	Shrub and brush rangeland	7	Rangeland (grassland)
33	Mixed rangeland	7	Rangeland (grassland)
11	Residental	9	Urban
12	Commercial and services	9	Urban
13	Industrial	9	Urban
14	Transportation-communication-utilities	9	Urban
15	Industrial and commercial complexes	9	Urban
16	Mixed urban or built-up land	9	Urban
17	Other urban or built-up land	9	Urban
71	Dry salt flats	10	Desert shrubland
72	Beaches	10	Desert shrubland

(continued)

Table 4-4. (continued)

Anderson Code and Description		RAMMET Type and Description	
73	Sandy areas not beaches	10	Desert shrubland
74	Bare exposed rock	10	Desert shrubland
75	Strip mines-quarries-gravel pits	10	Desert shrubland
76	Transitional areas	10	Desert shrubland
81	Shrub and brush tundra	10	Desert shrubland
82	Herbaceous tundra	10	Desert shrubland
83	Bare ground	10	Desert shrubland
85	Mixed tundra	10	Desert shrubland
91	Perennial snowfields	10	Desert shrubland
92	Glaciers	10	Desert shrubland

Anderson codes from Anderson et al. (1976), RAMMET codes from U.S. EPA (1995b).

height was collected from the local climatic data summaries (NOAA, 1983). When anemometer height was not available, the station was assigned the most common anemometer height from the other stations. This value was 6.1 m.

4.3.5.1 Anthropogenic Heat Flux. Anthropogenic heat flux for a meteorological station can usually be neglected in areas outside of highly urbanized locations; however, in areas with high population densities or energy use, such as an industrial facility, this flux may not always be negligible (U.S. EPA, 1995b). However, anthropogenic heat flux was assumed to be zero for all meteorological stations because little information was available to assume any anthropogenic heat flux value for most locations.

4.3.5.2 Bowen Ratio. The Bowen ratio is a measure of the amount of moisture at the surface around a meteorological station. The range of values is provided in Table 4-5 as a function of land use type, season, and moisture condition. Consistent with previous Office of Solid Waste (OSW) analyses, average seasonal and wet/dry values were used in the HWIR assessment. The wetness of a location was determined based on the annual average precipitation amount.

4.3.5.3 Monin-Obukhov Length. The minimum Monin-Obukhov length, a measure of the atmospheric stability at a meteorological station, was correlated with the land use classification, as shown in Table 4-6.

Table 4-5. Daytime Bowen Ratio by Land Use and Season (Meteorological Stations Only)

Land Use Type	Spring			Summer			Autumn			Winter			Average		
	Dry	Wet	Avg.	Dry	Wet	Avg.	Dry	Wet	Avg.	Dry	Wet	Avg.	Dry	Wet	Avg.
Water surface	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	2.0	0.3	1.5	0.575	0.15	0.45
Deciduous forest	1.5	0.3	0.7	0.6	0.2	0.3	2.0	0.4	1.0	2.0	0.5	1.5	1.53	0.35	0.875
Coniferous forest	1.5	0.3	0.7	0.6	0.2	0.3	1.5	0.3	0.8	2.0	0.3	1.5	1.4	0.275	0.825
Swamp	0.2	0.1	0.1	0.2	0.1	0.1	0.2	0.1	0.1	2.0	0.5	1.5	0.65	0.2	0.45
Cultivated land (agricultural)	1.0	0.2	0.3	1.5	0.3	0.5	2.0	0.4	0.7	2.0	0.5	1.5	1.63	0.35	0.75
Grassland	1.0	0.3	0.4	2.0	0.4	0.8	2.0	0.5	1.0	2.0	0.5	1.5	1.75	0.425	0.825
Urban	2.0	0.5	1.0	4.0	1.0	2.0	4.0	1.0	2.0	2.0	0.5	1.5	3.0	0.75	1.6
Desert shrub land	5.0	1.0	3.0	6.0	5.0	4.0	10.0	2.0	6.0	10.0	2.0	6.0	7.75	2.5	4.75

Source: U.S. EPA, 1995b. Averages computed for this effort.

Table 4-6. Minimum Monin-Obukhov Length (Stable Conditions)

Urban Land Use Classification	Length (meters)
Agriculture (open)	2
Residential	25
Compact residential/industrial	50
Commercial (19-40 story buildings) (> 40 story buildings)	100 150

Source: U.S. EPA, 1995b.

4.3.5.4 Albedo. Noontime albedo values also were correlated with land use around a meteorological station, as shown in Table 4-7. As with the Bowen ratio, the values also were correlated to the season, and the averaged values were used for the HWIR assessment.

4.3.5.5 Surface Roughness Length. The surface roughness length is a measure of the height of obstacles to the wind flow. It is not equal to the physical dimensions of the obstacles but is generally proportional to them. Surface roughness length data are shown in Table 4-8 along with their corresponding land use. These values were needed for the areas surrounding the meteorological station and the Industrial D facility. Area-averaged values were used for the HWIR analysis.

Table 4-7. Albedo of Natural Ground Covers for Land Use Types and Seasons

Land Use Type	Spring	Summer	Autumn	Winter	Average
Water surface	0.12	0.1	0.14	0.2	0.14
Deciduous forest	0.12	0.12	0.12	0.5	0.22
Coniferous forest	0.12	0.12	0.12	0.35	0.18
Swamp	0.12	0.14	0.16	0.3	0.18
Cultivated land (agricultural)	0.14	0.2	0.18	0.6	0.28
Grassland	0.18	0.18	0.20	0.6	0.29
Urban	0.14	0.16	0.18	0.35	0.21
Desert shrub land	0.3	0.28	0.28	0.45	0.33

Source: U.S. EPA, 1995b. Average values computed for this analysis.

Table 4-8. Surface Roughness Length (m) for Land Use Types and Seasons

Land Use Type	Spring	Summer	Autumn	Winter	Average
Water surface	0.0001	0.0001	0.0001	0.0001	0.0001
Deciduous forest	1.0	1.3	0.8	0.5	0.9
Coniferous forest	1.3	1.3	1.3	1.3	1.3
Swamp	0.2	0.2	0.2	0.05	0.16
Cultivated land (agricultural)	0.03	0.2	0.05	0.01	0.07
Grassland	0.05	0.2	0.01	0.001	0.04
Urban	1.0	1.0	1.0	1.0	1.0
Desert shrubland	0.3	0.3	0.3	0.15	0.26

Source: U.S. EPA, 1995b. Average values computed for this analysis.

4.3.5.6 Fraction of Net Radiation Absorbed by the Ground. During daytime hours, the heat flux into the ground is parameterized as a fraction of the net radiation incident on the ground. This fraction varies based on land use. A value of 0.15 was used for rural locations. Suburban and urban locations were given values of 0.22 and 0.27, respectively (U.S. EPA, 1995b).

4.3.6 PCRAMMET Input File Setup and Execution

PCRAMMET is a preprocessor program that integrates surface and upper air meteorological data into an input file for ISCST3. PCRAMMET calculates hourly stability values from surface observations, interpolates hourly mixing height values from twice-daily upper air data, and calculates parameters for wet and dry deposition/depletion calculations. PCRAMMET output can be selected as unformatted or ASCII format (U.S. EPA, 1995b). ISCST3 requires that meteorological data be in ASCII format.

PCRAMMET input files were set up in an automated fashion. They were created for all 218 surface stations from SAMSON, independent of the actual facilities being modeled. In addition to the hourly data, the following data were needed (U.S. EPA, 1995b):

- Mixing height data file name
- Hourly surface data file name
- Minimum Monin-Obukhov length (m)
- Anemometer height (m)
- Roughness length (m), surface meteorological station
- Roughness length (m), application site
- Noontime albedo
- Bowen ratio
- Anthropogenic heat flux (W/m^2)
- Fraction net radiation absorbed by the ground.

The surface and upper air data file names were based on the station numbers of the respective meteorological stations. Assignment of these was determined using a GIS, based on polygons drawn around each mixing height station (see Section 4.3.1). Outputs from PCRAMMET were the hourly meteorological files (see Appendix 4D), which include the meteorological parameters previously shown in Table 4-3.

PCRAMMET was run for the 99 surface stations assigned to the Industrial D sites using the automatically generated files. The meteorological files were then processed through an RTI program called FIXMET, which sets all mixing heights below 10 m to 10 m. This program was created and used to avoid unrealistically low mixing heights and resulting high air concentrations from the ISCST3 model.

4.3.7 Climatological Data Files

Climatological data (daily, monthly, annual, and long-term averages) are required by the source modules, the waterbody module, and the watershed module. For consistency, these data were calculated from the hourly SAMSON data described in the previous sections, along with the daily precipitation data from the cooperative summaries. Appendix 4D shows the format and content of the climatological files. Table 4-9 shows the data source used for calculation of each parameter, and Appendix 4E provides the programs used to prepare each data file.

4.3.7.1 **<u>Equations</u>**. Equations were used in some cases to calculate climatological parameters that were not available from SAMSON or from another source previously mentioned. Calculated parameters included daily evaporation, daily rainfall erosivity, windspeed at different wastepile heights, and Thornthwaite precipitation evaporation index (PE).

Daily evaporation was not available directly from any source for all of the meteorological stations. Therefore, it was necessary to calculate this parameter using the following equation (Viessman and Lewis, 1996):

$$E = (0.013 + 0.00016 * u_2)e_a\left(\frac{(100 - R_h)}{100}\right) \quad (4\text{-}1)$$

where

E	=	evaporation in cm/d
u_2	=	average daily windspeed at 2 m in km/d
e_a	=	vapor pressure in millibars
R_h	=	average daily relative humidity (RH) in percent (SAMSON).

This evaporation equation was chosen because the parameters were most easily available from the processed meteorological files and SAMSON. Other equations were available but relied on variables that could not be easily calculated or retrieved from the available data.

Equation 4-2 shows the calculation of windspeed at 2 m for the evaporation equation:

$$u_2 = \frac{\Sigma u_a\left(\frac{2}{a_n}\right)^{P(i)}}{24} \quad (4\text{-}2)$$

where

u_a	=	windspeed at anemometer height in m/s (SAMSON)
a_n	=	anemometer height in m (local climate summaries)
$P(i)$	=	wind exponent (U.S. EPA, 1995c).

Vapor pressure was calculated as follows:

$$e_a = V_P(R_{hd}) \quad (4\text{-}3)$$

Table 4-9. Climatological Parameters and Sources of Input Data

Climatological Input Parameter	Source of Parameter
Total daily precipitation	Cooperative station summaries
Daily average temperature	Processed hourly meteorological files
Daily average windspeed	Processed hourly meteorological files
Daily average cloud cover in tenths	Hourly SAMSON files
Daily average evaporation	Equation 4-1
Daily rainfall erosivity	Equation 4-4
Windspeed at 1 to 10 m wastepile heights	Equation 4-5
Number of hours where windspeed >5.4 m/s at wastepile height	Daily data file
Monthly mean temperature	Daily data file
Monthly mean windspeed	Daily data file
Monthly average precipitation	Daily data file
Monthly average evaporation	Daily data file
Monthly average cloud cover	Daily data file
Maximum daily average temperature for month	Daily data file
Minimum daily average temperature for month	Daily data file
Annual mean windspeed	Processed hourly meteorological file
Percent time wind >5.4 m/s at wastepile height	Daily data file
Annual fastest mile of wind	Processed hourly data file
Days/year with precipitation >0.01in	Daily data file
Thornthwaite precipitation-evaporation index (pe)	Equation 4-6
Annual precipitation	Daily data file
Annual evaporation	Daily data file
Annual average temperature	Daily data file
Long-term annual average fastest mile of wind	Annual data file
Long-term mean annual windspeed	Processed hourly data file
Mean annual days/year with >0.01 inch precipitation	Daily data file
Long-term average annual precipitation	Daily data file
Long-term average annual evaporation	Daily data file
Average annual usle rainfall erosivity factor	Wischmeier and Smith, 1978
Long-term average annual temperature	Processed hourly data file
Long-term average soil column temperature	Same as long-term average annual temperature
Long-term average cloud cover	Daily data file

where

V_P	=	vapor pressure (Viessman and Lewis, 1996)
R_{hd}	=	decimal relative humidity (SAMSON).

Daily rainfall erosivity is not a commonly used parameter. Usually, this parameter is used in its annual form. The long-term annual form of the rainfall erosivity is available from Wischmeier and Smith (1978). A methodology was designed to disaggregate this value into daily amounts. The following equation describes the methodology used to derive the daily rainfall erosivity:

$$R_d = R_{ann}(N_Y)(\frac{P_d}{P_{LT}}) \quad (4\text{-}4)$$

where

R_d	=	daily rainfall erosivity (1/d)
R_{ann}	=	long-term annual rainfall erosivity (1/y) (Wischmeier and Smith, 1978)
N_Y	=	number of years of meteorological data available
P_d	=	daily precipitation amount (in) (cooperative station data)
P_{LT}	=	sum of all precipitation for entire N_Y (in).

Windspeed at wastepile heights of 1 m, 2 m, 4 m, 6 m, 8 m, and 10 m was calculated using the following equation, which was modified from U.S. EPA (1995c):

$$u_{xd} = \frac{\Sigma u_a(\frac{x}{a})^{P(i)}}{24} \quad (4\text{-}5)$$

where

u_{xd}	=	daily average windspeed at pile height x
u_a	=	hourly windspeed at anemometer height a
P(i)	=	wind exponent (U.S. EPA, 1995c).

The Thornthwaite PE was calculated from the monthly data file using the following equation (Cowherd et al., 1985):

$$PE = 10[\Sigma(\frac{P_M}{E_M})] \quad (4\text{-}6)$$

where

P_M = monthly precipitation
E_M = monthly evaporation.

4.3.7.2 **Programs**. A series of programs was written to calculate and retrieve the climatological parameters for this analysis (see Appendix 4E). Separate programs were run for daily, monthly, annual, and long-term data, respectively. The programs performed the necessary calculations and unit conversions and placed parameters in comma-delimited text files of the same format found in Appendix 4D.

4.3.7.3 **Quality Control**. QC checks were performed on the climatological data using a hand calculator and Microsoft Excel's calculation function. Each field was checked for accuracy of calculation by a staff member following the equations on paper. In addition, the final long-term precipitation data were checked against ISMCS data (USN, USAF, USDOC, 1992) to ensure that the values were reasonable.

4.4 Results

The 201 HWIR Industrial D facilities were represented by 99 meteorological stations. Five data files were prepared for each station, for a total of 495 meteorological data files for the HWIR 3MRA model.

4.4.1 Meteorological Station Assignments

Figure 4-2 shows the facilities and their meteorological station assignments based on the polygon methodology (see Section 4.3.1). Meteorological data were collected for 99 surface stations chosen to represent 201 HWIR Industrial D facilities. Table 4-10 shows the 99 meteorological facilities, their associated mixing height stations, and the number of Industrial D facilities that use that particular surface station. Appendix 4F provides a list of meteorological station assignments for each of the 201 HWIR sites.

4.4.2 Meteorological Data

Examples of each HWIR 3MRA meteorological data input file are provided in Appendix 4D, along with their format and general content. The full meterological data set (over 1 gigabyte of data) is available as part of the HWIR 3MRA model system.

4.5 Issues and Uncertainties

The HWIR 3MRA meteorological data set represents the largest, most comprehensive set of meteorological data ever prepared for a risk assessment modeling effort. All available meteorological data were utilized for the effort to ensure that the most applicable data are used for each of the 201 Industrial D sites modeled for HWIR. In addition, extensive QC was conducted to ensure that the data were accurate and complete, as well as to identify and correct data gaps.

Table 4-10. Meteorological Stations Used

Surface Station Number	Surface Station	Surface State	Mixing Height Station Number	Mixing Height Station	Mixing State	Number of Ind. D Facilities
03812	Asheville	NC	13723	Greensboro	NC	4
03813	Macon	GA	13873	Athens	GA	1
03822	Savannah	GA	13880	Charleston	SC	1
03856	Huntsville	AL	13897	Nashville	TN	5
03860	Huntington	WV	03860	Huntington	WV	1
03870	Greenville	SC	13873	Athens	GA	2
03927	Fort Worth	TX	13901	Stephenville	TX	1
03928	Wichita	KS	13996	Topeka	KS	3
03937	Lake Charles	LA	03937	Lake Charles	LA	1
03940	Jackson	MS	03940	Jackson	MS	1
03945	Columbia	MO	03946	Monett	MO	1
03947	Kansas City	MO	13996	Topeka	KS	3
04725	Binghamton	NY	14735	Albany	NY	4
12839	Miami	FL	12839	Miami	FL	1
12842	Tampa	FL	12842	Tampa	FL	4
12916	New Orleans	LA	03937	Lake Charles	LA	1
12917	Port Arthur	TX	03937	Lake Charles	LA	2
12960	Houston	TX	03937	Lake Charles	LA	6
13722	Raleigh-Durham	NC	13723	Greensboro	NC	5
13723	Greensboro	NC	13723	Greensboro	NC	3
13739	Philadelphia	PA	93734	Sterling	VA	7
13741	Roanoke	VA	13723	Greensboro	NC	3
13781	Wilmington	DE	93734	Sterling	VA	3
13865	Meridian	MS	03940	Jackson	MS	1
13873	Athens	GA	13873	Athens	GA	2

(continued)

Table 4-10. (continued)

Surface Station Number	Surface Station	Surface State	Mixing Height Station Number	Mixing Height Station	Mixing State	Number of Ind. D Facilities
13874	Atlanta	GA	13873	Athens	GA	4
13877	Bristol	TN	03860	Huntington	WV	3
13880	Charleston	SC	13880	Charleston	SC	3
13881	Charlotte	NC	13723	Greensboro	NC	1
13883	Columbia	SC	13873	Athens	GA	3
13891	Knoxville	TN	13897	Nashville	TN	1
13893	Memphis	TN	13963	Little Rock	AR	4
13957	Shreveport	LA	03951	Longview	TX	2
13958	Austin	TX	12912	Victoria	TX	3
13967	Oklahoma City	OK	13967	Oklahoma City	OK	1
13968	Tulsa	OK	13967	Oklahoma City	OK	2
13970	Baton Rouge	LA	03937	Lake Charles	LA	1
13985	Dodge City	KS	13985	Dodge City	KS	1
13994	St. Louis	MO	03879	Salem	IL	1
13995	Springfield	MO	03946	Monett	MO	1
14733	Buffalo	NY	14733	Buffalo	NY	1
14734	Newark	NJ	94789	JFK	NY	1
14735	Albany	NY	14735	Albany	NY	2
14737	Allentown	PA	93734	Sterling	VA	4
14739	Boston	MA	14764	Portland	ME	2
14740	Hartford	CT	14735	Albany	NY	2
14742	Burlington	VT	14735	Albany	NY	4
14764	Portland	ME	14764	Portland	ME	2
14768	Rochester	NY	14733	Buffalo	NY	1
14777	Wilkes-Barre	PA	14735	Albany	NY	2

(continued)

Table 4-10. (continued)

Surface Station Number	Surface Station	Surface State	Mixing Height Station Number	Mixing Height Station	Mixing State	Number of Ind. D Facilities
14778	Williamsport	PA	94823	Pittsburgh	PA	1
14820	Cleveland	OH	14733	Buffalo	NY	3
14821	Columbus	OH	13840	Dayton	OH	1
14826	Flint	MI	14826	Flint	MI	1
14840	Muskegon	MI	14826	Flint	MI	1
14842	Peoria	IL	14842	Peoria	IL	1
14848	South Bend	IN	14826	Flint	MI	1
14852	Youngstown	OH	94823	Pittsburgh	PA	2
14895	Akron/Canton	OH	94823	Pittsburgh	PA	2
14898	Green Bay	WI	14898	Green Bay	WI	2
14913	Duluth	MN	14918	International Falls	MN	1
14923	Moline	IL	14842	Peoria	IL	2
23042	Lubbock	TX	23047	Amarillo	TX	1
23044	El Paso	TX	23044	El Paso	TX	1
23047	Amarillo	TX	23047	Amarillo	TX	4
23160	Tucson	AZ	23160	Tucson	AZ	1
23169	Las Vegas	NV	03160	Desert Rock	NV	1
23174	Los Angeles	CA	23230	Oakland	CA	4
23183	Phoenix	AZ	23160	Tucson	AZ	4
23232	Sacramento	CA	23230	Oakland	CA	3
23234	San Francisco	CA	23230	Oakland	CA	1
24018	Cheyenne	WY	24023	North Platte	NE	1
24033	Billings	MT	24143	Great Falls	MT	1
24037	Miles City	MT	24143	Great Falls	MT	2
24127	Salt Lake City	UT	24127	Salt Lake City	UT	2

(continued)

Table 4-10. (continued)

Surface Station Number	Surface Station	Surface State	Mixing Height Station Number	Mixing Height Station	Mixing State	Number of Ind. D Facilities
24131	Boise	ID	24131	Boise	ID	1
24155	Pendleton	OR	24157	Spokane	WA	1
24156	Pocatello	ID	24127	Salt Lake City	UT	1
24157	Spokane	WA	24157	Spokane	WA	1
24225	Medford	OR	24225	Medford	OR	1
24229	Portland	OR	24232	Salem	OR	3
24233	Seattle	WA	94240	Quillayute	WA	1
24243	Yakima	WA	24157	Spokane	WA	1
93193	Fresno	CA	23230	Oakland	CA	3
93721	Baltimore	MD	93734	Sterling	VA	1
93730	Atlantic City	NJ	93755	Atlantic City	NJ	1
93805	Tallahassee	FL	12832	Apalachicola	FL	1
93815	Dayton	OH	13840	Dayton	OH	3
93817	Evansville	IN	03879	Salem	IL	2
93819	Indianapolis	IN	13840	Dayton	OH	2
93820	Lexington	KY	03860	Huntington	WV	1
93822	Springfield	IL	14842	Peoria	IL	1
94018	Boulder	CO	23062	Denver	CO	1
94728	New York	NY	94789	JFK	NY	2
94822	Rockford	IL	14842	Peoria	IL	1
94823	Pittsburgh	PA	94823	Pittsburgh	PA	2
94846	Chicago	IL	14842	Peoria	IL	3
94847	Detroit	MI	14826	Flint	MI	3
94860	Grand Rapids	MI	14826	Flint	MI	2

The great volume of data (5 data files for 99 meteorological stations) required extensive automated data processing to compile and calculate . Most of the issues and uncertainties associated with this effort have to do with the assumptions and simplifications necessary to write the automation programs, or uncertainties associated with replacing missing data. In summary, in practically every case these uncertainties were not likely to have introduced any significant consistent bias to the data or the analysis. One advantage of automated processing is that, unlike subjective methods based on professional judgment, there is not the potential bias associated with different meteorological judgments.

4.5.1 Precipitation Data

Inaccurate precipitation data in the SAMSON data set necessitated creation of hourly precipitation data by distributing daily cooperative summary precipitation data across a 24-hour period, using SAMSON as an hourly precipitation template. Issues and uncertainties associated with this methodology are mainly associated with the lack of knowledge of when precipitation events did actually occur, given that the instruments used to measure the hourly rainfall totals in SAMSON could not measure small amounts of rainfall. This limited knowledge of hourly precipitation amounts could bias the record by stretching rain events over more hours than they occurred or by attributing more rain into a shorter amount of time than actually occurred, which should have opposing effects. One small source of a consistent bias is that the cooperative summary data can be missing hours at the beginning and end of the 24-hour period, which would underestimate precipitation totals by not including any rainfall or snowfall during those periods.

4.5.2 Meteorological Data Completeness

During data collection, it was found that most meteorological stations were missing significant portions of data. For instance, for most stations only about one-third of the observations are complete for the late 1960s and early 1970s. Data for these years were usually discarded, and, as a result, complete data for 30 years were never available. Other stations were completely discarded because too much data were missing. In these cases, alternative stations nearby were used instead. It is not known whether missing data for the late 1960s or early 1970s represents a consistent bias, but it is not thought so.

4.5.3 Meteorological Station Representativeness

Meteorological station assignments were based on both proximity and climatic regimes, as reflected by ecological habitats and physiography. Although these are likely to be accurate for most sites (and adequate for a national analysis), it is possible that microclimates at some sites station assignments could be different than the closest station. Although this could lead to some inaccuracy with respect to site-specific results, it is probably not a significant source of error for a national, site-based analysis like the HWIR 3MRA.

4.5.4 Replacing Missing Data

When replacing missing surface or mixing height data, Atkinson and Lee (1992) report that each occurrence of missing data can be filled in more accurately if it is examined individually and filled in using objectivity, where possible, and subjective meteorological experience as necessary. The objective methodology recommended by Atkinson and Lee (1992) was primarily used in this effort, with simple, automated linear interpolation being used for the mixing height data. Although for a particular site and time this method may not necessarily be representative of the nature of the atmosphere, this automation was necessary to complete the extensive data collection effort in a timely, consistent, and precise manner. In addition, the automated interpolation method should not introduce a consistent bias, while professional judgment can. However, independent examination of interpolation results, and, perhaps, comparison of replaced data with actual data for sites with data, may be necessary to confirm this lack of consistent bias.

4.6 References

Anderson, J. R., E. E. Hardy, J. T. Roach, and R. E. Witmer. 1976. A land use and land cover classification system for use with remote sensor data. Geological Survey Professional Paper 964. In: *U.S. Geological Survey Circular 671*. U.S. Geological Survey, Washington, DC. Website at http://www-nmb.usgs.gov/pub/ti/LULC/lulcpp964/lulcpp964.txt.

Andrews, Paul. 1998. "Personal communication with Dennis Atkinson, U.S. EPA." Memorandum to Docket File HWIR from Paul Andrews (Research Triangle Institute), Research Triangle Park, NC. April 15.

Atkinson, D., and R. F. Lee. 1992. *Procedures for Substituting Values for Missing NWS Meteorological Data for Use in Regulatory Air Quality Models*. U.S. Environmental Protection Agency, Research Triangle Park, NC. July 7.

Bailey, Robert G., Peter E. Avers, Thomas King, and W. Henry McNab. 1994. Ecoregions and Subregions of the United States. Forest Service, Washington, DC. Website at http://www.epa.gov/docs/grdwebpg/bailey/.

Cowherd, C.J., G.E. Muleski, P.J. Englehart, and D.A. Gillette. 1985. *Rapid Assessment of Exposure to Particulate Emissions from Surface Contamination Sites.* U.S. Environmental Protection Agency. Office of Health and Environmental Assessment. Office of Research and Development. Washington, DC.

FSL and NCDC (Forcast Systems Laboratory and National Climatic Data Center). 1997. Radiosonde Data of North America: 1946-1996. Version 1.0. Department of Commerce, Forecast Systems Laboratory, Boulder, CO. June.

Holzworth, G. C. 1972. *Mixing Heights, Wind Speeds, and Potential for Urban Air Pollution Throughout the Contiguous United States*. AP-101. U.S. Environmental Protection Agency, Office of Air Programs, Division of Meteorology, Research Triangle Park, NC. January.

NCDC, ERL, and NWS (National Climatic Data Center; Environmental Research Laboratories; National Weather Service;). 1995. Cooperative Summary of the Day: TD3200-Period of record through 1993 CD-ROM. Department of Commerce, National Oceanic and Atmospheric Administration, Asheville, NC. June.

NOAA (National Oceanic and Atmospheric Administration). 1983. *Local Climatological Data. Annual Summaries for 1982: Part I - ALA - MONT and Part II - NEB - WYO.* National Climatic Data Center, National Environmental Satellite, Data, and Information Service, Asheville, NC.

U.S. DOC and U.S. DOE (Department of Commerce and Department of Energy). 1993. Solar and Meteorological Surface Observation Network (SAMSON). Version 1.0. Department of Commerce, National Climatic Data Center, Asheville, NC. September.

U.S. EPA (Environmental Protection Agency). 1994. *1:250,000 Scale Quadrangles of Landuse/Landcover GIRAS Spatial Data in the Conterminous United States: Metadata.* Office of Information Resources Management, Washington, DC. Website at http://www.epa.gov/ngispgm3/nsdi/projects/giras.htm.

U.S. EPA (Environmental Protection Agency). 1995. *Draft User's Guide for the Industrial Source Complex (ISC3) Dispersion Models. Volume I: User Instructions.* (Revised). EPA-454/B-95-003a. U.S. Environmental Protection Agency, Emissions, Monitoring, and Analysis Division, Office of Air Quality Planning and Standards, Research Triangle Park, NC. July.

U.S. EPA (Environmental Protection Agency). 1995. *PCRAMMET User's Guide.* (Draft). U.S. Environmental Protection Agency, Office of Air Quality Planning and Standards, Research Triangle Park, NC. July.

U.S. EPA (Environmental Protection Agency). 1995. *User's Guide for the Industrial Source Complex (ISC3) Dispersion Models. Volume II: Description of Model Algorithms.* EPA-454/B-95-003b. U.S. Environmental Protection Agency, Emissions, Monitoring, and Analysis Division, Office of Air Quality Planning and Standards, Research Triangle Park, NC. September.

U.S. EPA (Environmental Protection Agency). 1998. Support Center for Regulatory Air Models Website. Office of Air Quality Planning and Standards, Research Triangle Park, NC. Website at http://www.epa.gov/scram001. March 2.

USGS (Geological Survey). 1990. *Land Use and Land Cover Digital Data from 1:250,000- and 1:100,000-Scale Maps. Data Users Guide 4.* U.S. Geological Survey, Reston, VA.

USN, USAF, and U.S. DOC (U.S. Navy; U.S. Air Force; Department of Commerce;). 1992. International Station Meteorological Climate Summary (ISMCS). Version 4.0. Department of Defense and Department of Commerce, Federal Climate Complex, Asheville, NC. June.

Viessman, Warren, Jr., and Gary L. Lewis. 1996. *Introduction to Hydrology.* 4th Edition. HarperCollins College Publishers, New York, NY. pp. 89-91.

Wischmeier, W. H., and D. D. Smith. 1978. Predicting rainfall erosion losses. A guide to consevation planning. In: *Agricultural Handbook.* 537 Edition. U.S. Department of Agriculture, Washington, DC.

Appendix 4A. Meteorological Data QA/QC Programs

Two FORTRAN 77 programs were written to check meteorological data for completeness before it was processed through PCRAMMET. The QAQC program checks mixing height files in the format retrieved from EPA's SCRAM Bulletin Board; Attachment 4A-1 lists the QAQC source code. The SQAQC program checks surface data retrieved from SAMSON; Attachment 4A-2 contains the SQAQC source code. SQAQC can process up to 30 years of full-length meteorological files from between 1961 and 1990 at one time.

QAQC and SQAQC were verified using two sites for each program. Surface data were chosen from Evansville, IN, and Pittsburgh, PA, for 1985 to 1990. Upper air data were chosen from Saulte Saint Marie, MI, and Desert Rock, NV, for 1984 to 1990. Verification of each program was conducted in four steps, as follows:

- Step 1 involved running a few years of data through each program. Missing data indicated in the programs output matched what was really missing in the file in all cases. Table 4A-1 shows the occurrences of missing data for the surface stations chosen. Table 4A-2 shows the occurrences of missing mixing height data.

- During Step 2 each file was scanned by eye for additional missing data not identified by the program. None was found in any case.

- Step 3 involved artificially creating missing data by removing data from files and recording the date-time groups where this was done. Table 4A-3 shows the data removed from the surface stations and Table 4A-4 shows the dates for which mixing heights were removed.

- In Step 4, the program was run again, and the output was matched to the dates for which data were removed. The program successfully found all occurrences of synthetic missing data.

Table 4A-1. Occurrence of Missing Surface Data for Verification Step 1

Evansville, IN		**Pittsburgh, PA**	
Date/Time	Missing data	Date/Time	Missing data
861124/14	Ceiling	871231/24	Obs. indicator
861124/15	Ceiling	871231/24	Wind direction
871231/24	Obs. indicator	871231/24	Windspeed
871231/24	Wind direction	871231/24	Ceiling
871231/24	Windspeed	881231/24	Obs. indicator
871231/24	Ceiling	881231/24	Wind direction
881231/24	Obs. indicator	881231/24	Windspeed
881231/24	Wind direction	881231/24	Ceiling
881231/24	Windspeed	891231/24	Obs. indicator
881231/24	Ceiling	891231/24	Wind direction
891231/24	Obs. indicator	891231/24	Windspeed
891231/24	Wind direction	891231/24	Ceiling
891231/24	Windspeed	900625/03	Wind direction
891231/24	Ceiling	900625/03	Windspeed
901231/24	Obs. indicator	901231/24	Obs. indicator
901231/24	Opaque cloud cvr.	901231/24	Opaque cloud cvr.
901231/24	Temperature	901231/24	Temperature
901231/24	Station pressure	901231/24	Station pressure
901231/24	Wind direction	901231/24	Wind direction
901231/24	Windspeed	901231/24	Windspeed
901231/24	Ceiling	901231/24	Ceiling

Table 4A-2. Dates of Missing Mixing Height Data for Verification Step 1

Saulte Saint Marie, MI	Desert Rock, NV
None	890225

Table 4A-3. Data Removed from Surface Stations for SQAQC Verification Testing

Evansville, IN		Pittsburgh, PA	
Date/Time	Missingdata	Date/Time	Missing data
850101/07	Windsp./dir.	850325/01	Windsp./dir.
850124/12	Ceiling height	850325/07	Ceiling height
850124/12	Precipitation	851231/20	Temperature
850208/06	Temperature	861019/22	Obs. ind./opaque
850317/17	Station pressure	861016/17	Pressure
850320/06	Windsp./dir.		
850321/24	Opaque cloud cvr.		
850323/12	Observation indicator		

Table 4A-4. Mixing Height Data Removed for QAQC Verification Testing

Sault Saint Marie, MI	Desert Rock, NV
860112	841219
871111	841216
871220 AM	871283
871220 PM	880407
871231	880408
891103	901222
900112	

Attachment 4A-1. Source Code for QAQC

```
      PROGRAM QAQC
*********************************************
*PROGRAM TO SCREEN UPPER AIR METEOROLOGICAL *
*FILES FOR MISSING DATA.  OUTPUT CAN BE     *
*FOUND IN MISS.LOG.                         *
*VARIABLES INCLUDE:     STAT: STATION NUMBER*
*                       YEAR: YEAR OF RECORD*
*                       COMB: STAT//YEAR    *
*                       TXT:  .TXT EXTENSION*
*                       GROUP:FILE NAME     *
*                       A:    AM MIX HT.    *
*                       P:    PM MIX HT.    *
*                       T:    DATE/TIME GRP *
*                       S:    STATION ID    *
*                       J:    COUNTING VAR. *
*                             DEFINES YEAR  *
*                       X:    COUNTER       *
*********************************************
*DEFINE VARIABLES*
      INTEGER YEAR, YEARJ(30), A, P, S, T, X, J
      CHARACTER*6, STAT
      CHARACTER*4, TXT
      CHARACTER*12, GROUP
      CHARACTER*2, YEARA(30)
      DATA YEARA /'61','62','63','64','65','66','67','68','69','70',
     .'71','72','73','74','75','76','77','78','79','80','81','82','83',
     .'84','85','86','87','88','89','90'/
*OPENS LOG FILE*
      OPEN (UNIT = 1, TYPE='NEW', NAME = 'MISS.LOG')
*ASKS FOR INPUT*
      PRINT *, ' ENTER STATION #AND A DASH:'
      READ *, STAT
      PRINT *, ' ENTER THE 1ST YEAR TO BE RUN(61):'
      READ (*,'(I2)') YEAR
      PRINT *, ' ENTER FILE EXTENSION (.TXT):'
      READ *, TXT
      WRITE (1, '(1X,A,A,I2)')
     .'MISSING MIXING HEIGHT DATA FOR STATION AND START YEAR:',STAT,YEAR
*DO LOOP TO READ FILES*
10    DO J=YEAR,90
      YEARJ(J-60)=J
      WRITE(*,'(1X,i2)') YEARJ
      GROUP(1:6) = STAT
      GROUP(7:8) = YEARA(J-60)
```

```
      GROUP(9:12) = TXT
      OPEN (UNIT = J, NAME = GROUP, ERR = 10)
      WRITE(*,'(1X,A12)') GROUP
40    X=1
*READS UPPER AIR FILE*
50    READ (J,60,END=80) S, T, A, P
60    FORMAT (I5, I6, 1X, I5, 13X, I5)
*IF THEN STATEMENTS TO WRITE MESSAGES TO LOG FILE*
        IF (A .EQ. 0) THEN
          WRITE (1,'(/1X,A21,I6)') 'MISSING AM MIXING HT:', T
        END IF
61      IF (P .EQ. 0) THEN
          WRITE (1,'(/1X,A21,I6)') 'MISSING PM MIXING HT:', T
        END IF
*IF THEN STATEMENTS TO COUNT THROUGH NUMBER OF LINES IN FILE*
63    X=X+1
      IF (X .LT. 368) THEN
      GO TO 50
      END IF
*CLOSE FILE*
80    CLOSE (J)
      END DO
*CLOSES LOG*
200   CLOSE (1)
      WRITE(*,'(1X,A)') 'PROGRAM FINISHED CHECKING FILES!'
      END
```

Attachment 4A-2. Source Code for SQAQC

```
      PROGRAM SQAQC
********************************************************
*PROGRAM TO SCREEN SURFACE METEOROLOGICAL DATA FOR      *
*MISSING VARIABLES.  OUTPUT CAN BE FOUND IN SMISS.LOG *
*VARIABLES INCLUDE:    STAT:      STATION NUMBER*
*                 YEAR:      YEAR OF RECORD*
*                 DAT:       .DAT EXTENSION*
*                 GROUP:     FILE NAME    *
*                 Y:         YEAR IN FILE  *
*                 M:         MONTH IN FILE *
*                 D:         DAY IN FILE   *
*                 H:         HOUR IN FILE  *
*                 O:         OBS. INDIC.   *
*                 C:         OPAQUE SKY COV*
*                 T:         DRY BULB TEMP *
*                 W:         STATION PRESS *
*                 V:         WIND DIRECTION*
*                 S:         Windspeed    *
*                 R:         CEILING HEIGHT*
*                 P:         HOURLY PRECIP *
*                 J:         COUNTING VAR. *
*                            DEFINES YEAR  *
********************************************************
*DEFINE VARIABLES*
      INTEGER YEAR, YEARJ(30),Y, M, D, H, O, C, W, V, R, J
      REAL T, S
      CHARACTER*6, STAT
      CHARACTER*1, P
      CHARACTER*4, DAT
      CHARACTER*12, GROUP
      CHARACTER*2, YEARA(30)
      DATA YEARA /'61','62','63','64','65','66','67','68','69','70',
     .'71','72','73','74','75','76','77','78','79','80','81','82','83',
     .'84','85','86','87','88','89','90'/
*OPEN LOG FILE*
      OPEN (UNIT = 1, TYPE = 'NEW', NAME = 'SMISS.LOG')
*ASK FOR INPUT*
      PRINT*, 'ENTER STATION # AND A DASH:'
      READ *, STAT
      PRINT*, 'ENTER THE 1ST YEAR TO BE RUN (61):'
      READ (*,'(I2)') YEAR
      PRINT*, 'ENTER FILE EXTENSION (.DAT):'
      READ *, DAT
      WRITE (1,'(1X,A,A,I2)')
```

```
    .'MISSING DATA FOR STATION NUMBER/START YEAR:', STAT,YEAR
*DO LOOP TO READ FILES*
10    DO J=YEAR,90
      YEARJ(J-60)=J
      GROUP(1:6) = STAT
      GROUP(7:8) = YEARA(J-60)
      GROUP(9:12) = DAT
      OPEN (UNIT = J, NAME = GROUP, ERR = 10)
      WRITE(*,'(1X,A12)') GROUP
      X=1
*READ SURFACE DATA FILE*
*READ/FORMAT TWICE BECAUSE 1ST ONE SKIPS FIRST TWO LINES*
20    READ (J,30,END=90) Y, M, D, H, O, C, T, W, V, S, R, P
30    FORMAT (//1X,I2,1X,I2,1X,I2,1X,I2,1X,I1,1X,I2,1X,F5.0,1X,I4,1X,I3,
     .1X,F5.0,1X,I6,17X,A1)
      GO TO 70
*LOOP RETURNS TO HERE TO READ 2ND LINE OF DATA TO END OF PROGRAM*
50    READ (J,60,END=90) Y, M, D, H, O, C, T, W, V, S, R, P
60    FORMAT (1X,I2,1X,I2,1X,I2,1X,I2,1X,I1,1X,I2,1X,F5.0,1X,I4,1X,I3,
     .1X,F5.0,1X,I6,17X,A1)
*IF THEN STATEMENT TO FIND END OF FILE*
70    IF (Y .EQ. 0) THEN
        GO TO 90
      END IF
*IF THEN STATEMENTS TO WRITE TO FILE*
      IF (O .EQ. 9) THEN
        WRITE (1,'(/1X,A,I2,I2,I2,I2)') 'MISSING DATA:',Y,M,D,H
      END IF
      IF (C .EQ. 99) THEN
        WRITE (1,'(/1X,A,I2,I2,I2,I2)') 'MISSING OPAQUE:',Y,M,D,H
      END IF
      IF (T .EQ. 9999.) THEN
        WRITE (1,'(/1X,A,I2,I2,I2,I2)') 'MISSING TEMP:',Y,M,D,H
      END IF
      IF (W .EQ. 9999) THEN
        WRITE (1,'(/1X,A,I2,I2,I2,I2)') 'MISSING PRES.:',Y,M,D,H
      END IF
      IF (V .EQ. 999) THEN
        WRITE (1,'(/1X,A,I2,I2,I2,I2)') 'MISSING WIND:',Y,M,D,H
      END IF
      IF (S .EQ. 9999.) THEN
        WRITE (1,'(/1X,A,I2,I2,I2,I2)') 'MISSING WIND:',Y,M,D,H
      END IF
      IF (R .EQ. 999999) THEN
        WRITE (1,'(/1X,A,I2,I2,I2,I2)') 'MISSING CEIL:',Y,M,D,H
      END IF
```

```
      IF ((P .EQ. 'M') .OR. (P .EQ. 'A')) THEN
        WRITE (1,'(1X,A,I2,I2,I2,I2)') 'MISSING PREC:',Y,M,D,H
      END IF
      GO TO 50
*CLOSE THE FILE*
90    CLOSE (J)
      END DO
*CLOSES LOG FILE*
      CLOSE (1)
      PRINT*, 'PROGRAM IS FINISHED CHECKING SURFACE FILES!'
      END
```

Appendix 4B. METFIX Program

METFIX is a FORTRAN77 program that fills in missing meteorological parameters for the SAMSON surface meteorological data files including cloud cover, temperature, pressure, wind direction, windspeed, and ceiling height. The program was not designed to fill in precipitation data.

The program fills in up to five consecutive hours of missing data. The procedures used to fill in a single isolated missing value generally follows the objective procedures in Atkinson and Lee (1992). Because the subjective procedures recommended to fill in more than 1 missing values in Atkinson and Lee (1992) are based on professional judgment could not be programmed, and Atkinson and Lee (1992) did not discuss all the cases of missing data, procedures and rules were designed to fill in the missing data in an automated fashion.

The followings are the rules used to fill in missing data with the METFIX program. The METFIX program uses these rules to fill in missing data in a surface meteorological data file and writes the corrected meteorological data to another file. METFIX also writes an information message to a log file listing what changes were made for all the missing hours. The program generates a third file to list all the files that were processed. Attachment 4B-1 provides the source code for the program.

General Rules (for all variables)

(1) If 1 value is missing, use previous value.
(2) If 2-5 values are missing, interpolate between the single value before the missing values and the single value after the missing value.
(3) If more than 5 are missing, the program will not interpolate any values and will write a message to the log file to state those missing data are not filled in.
(4) If first 1-5 values of file are missing, simply copy 1st real value into the missing values. If more than 5 are missing, don't copy.
(5) If last 1-5 values of file are missing, simply copy last real value into the missing values. If more than 5 are missing, don't copy.
(6) If Ceiling Height is listed as 88888, change it to 77777.

Rules for Specific Variables

If Cloud Cover is missing, METFIX will check the Ceiling Height for that hour.
(1) If Ceiling Height is 77777 or 88888, 0 will be filled in for Cloud Cover.
(2) If Ceiling Height is non-77777 or non-88888, 10 will be filled in for Cloud Cover.
(3) If Ceiling Height is also missing, use General Rules.

If Ceiling Height is missing, METFIX will first check the Cloud Cover for that hour.
(1) If Cloud Cover can not be filled in for that hour, Ceiling Height will not be filled in.
(2) If Cloud Cover is 5 or less, 77777's will be filled in for Ceiling Height.
(3) If Cloud Cover is 6 or more, the program will do followings:

(a) If 1 Ceiling Height is missing, use the non-77777 value either before or after. If both of the values are 77777's, replace the missing value with 77777's.
(b) If 2 to 5 values are missing, interpolate if before and after values are not 77777's. If one of the values is not 77777's, simply copy the "real" value to the missing values. If both of the values are 77777's, replace all the missing values with 77777's.

If 2 to 5 values of wind direction are missing, the program will check the values before and after to determine which wind direction might be for the missing period. For example, interpolate 2 values between 340 and 10 would be 350 and 0 (not 120 and 230).

Attachement 4B-1. METFIX Source Code

METFIX program 8/25/98 3:40pm

```
        program metfix
* --------------------------------------------------------------------------
* Program:      METFIX
* Date: August 1998
* By:           Wayne Winstead, Research Triangle Institute
*
* Actions:      METFIX reads the file STATION.DAT to get a list of met
*               files to operate on.  METFIX then opens, reads, and fills
*               in missing meteorological data based on rules provided by
*               Chengwei Yao and Paul Andrews for project 7200-15A
*
*                -Filename-      #    -Description-
* Input:        STATION.DAT   - (4) list of met files
*               xxxxxxxx.DAT  - (1) actual met files
* Output:       xxxxxxxx.DA2  - (2) met files that have been filled/corrected
*               xxxxxxxx.LOG  - (3) changes made from .DAT to .DA2 file
*               METFIX.LOG    - (6) master log file listing files operated on
*
* Warnings/Details
*       All input xxxxxxxx.DAT files are expected to have 8 character
*               filenames (12345-99.dat)
* --------------------------------------------------------------------------
* METFIX rules:    August 19 & 20
*
*General Rules for all variables except Cloud Cover and Ceiling Height:
* At beginning of file, if first 1-5 values are missing, fill with first real #.
* At end of file, if last 1-5 values are missing, fill with last real #.
*
* If 1 value is missing, use previous value.
* If 2-5 values are missing, interpolate to fill the missing values.
* If more than 5 values are missing, do not fill in any values.
*
*Cloud Cover:
* If more than 5 values are missing, do not replace any values.
* If Ceiling Height is 77777, Cloud Cover = 0
* If Ceiling Height is real #, Cloud Cover = 10
* If Ceiling Height is missing, Interpolate/replace Cloud Cover values.
*
*Ceiling Height:
* If more than 5 values are missing, do not replace any values.
* If Cloud Cover is missing, Do not fill in Ceiling Height
* If Cloud Cover is 5 or less, Ceiling Height = 77777
```

```
* If Cloud Cover is 6 or more, then
*   If 1 Ceiling height is missing:
*         If it is bounded by 77777's, replace missing value with 77777.
*         If it is bounded by 77777's and a real #, replace with the real #.
*         If it is bounded by 2 real #s, replace with the previous real #.
*   If 2-5 Ceiling heights are missing,
*         If they are bounded by 77777's, replace all missing with 77777.
*         If they are bounded by 77777 and real#, copy real # into missing values.
*         If they are bounded by real #'s, interpolate.
*
* -----------------------------------------------------------------------
        implicit none

        character header*62,   !Read/write 2 file header lines
     .            varname(6)*14,          !Variable name for output log
     .            infile*12,   !Station name for file creation
     .            station*8    !Station name
        integer i, j,          ! Counters: I = record, J = Variable
     .            k, L, M, N, P, ! Other Counters
     .            intval,      ! Integer used for rounding wind direction
     .            modecnt(6),  ! How many values missing for this variable
     .            passcnt      ! 1st or 2nd pass at the data
        character mode(6)*1,   ! D-Date, I-Interpolate, 9-Missing, 0-Beginning
     .            okprint(7,6), ! Flag variable as ready to print
     .            message(6)*22,          ! Messages for log file
     .            lastwrite(6)*1, ! EOF flag to only print messages once
     .            both9*1                 ! Flag indicating missing both cloudc & ceiling
        real      oldval(6),   ! 1st value used in interpolation
     .            newval(6),   ! Last value used in interpolation
     .            newval2                 ! Used in a mini loop to interpolate

        character dates(7)*12,            ! Input date
     .              fill1(7)*3  ! Unused data from record (1 of 2)
        integer   cloudc       ! 99
        real      temp         ! 9999.
        integer   pres         ! 9999
        integer   winddir      ! 999
        real      windspd      ! 99.0   *or*  9999.
        integer   ceiling      ! 999999
        character fill2(7)*17  ! Unused data from record (2 of 2)
        real      invar(7,6)   ! Input variables - up to 7 records
        real      nines(6)     ! Different "missing value numbers"
        integer   sevens, eights          ! For all 8's or all 7's

        nines(1) = 99          ! Critical that these be correct!!!!
        nines(2) = 9999.       ! I am testing for .ge. these values
```

```
      nines(3) = 9999                 !    instead of just .eq.
      nines(4) = 999         !    This will make it easier to deal with
      nines(5) = 99. !or 9999.!    WindSpeed which has 2 Missing value flags
      nines(6) = 999999      !
      eights   = 88888
      sevens   = 77777

      varname(1) = 'Cloud Cover   '! Messages info for log file
      varname(2) = 'Temperature   '
      varname(3) = 'Pressure      '
      varname(4) = 'Wind Direction'
      varname(5) = 'Windspeed     '
      varname(6) = 'Ceiling Height'
      message(1) = 'Missing Cloud Cover   '
      message(2) = 'Missing Temperature   '
      message(3) = 'Missing Pressure      '
      message(4) = 'Missing Wind Direction'
      message(5) = 'Missing Windspeed     '
      message(6) = 'Missing Ceiling Height'

      open (unit=6, file='metfix.log', status='new')
      open (unit=4, file='station.dat', status ='old')

      ! This will act as the MAIN loop - looping once for each input file.
1     read (4,'(a)',err=9997,end=9999)station
      write(*,'(/,a,a)')' Operating on station ',station
      write(6,'(/,a,a)')' Operating on station ',station
      infile(1:8) = station

      infile(9:12) = '.dat'
      open (unit=1, file=infile, status='old', err=9998)
      infile(9:12) = '.da1'
      open (unit=2, file=infile, status='new')
      infile(9:12) = '.log'
      open (unit=3, file=infile, status='new')

      passcnt = 0
5     passcnt = passcnt + 1
      if (passcnt .eq. 2) then
        write(*,'(/,a,/)')' Open/Close files for 2nd pass at data...'
        close (unit=1)
        close (unit=2)
        infile(9:12) = '.da1'
        open(unit=1, file=infile, status='old', err=9998)
        infile(9:12) = '.da2'
        open(unit=2, file=infile, status='new')
```

```
          write(3,'(a)')' Starting 2nd pass at data...'
          endif
        if (passcnt .eq. 3) goto 5000

        do j = 1, 6
          mode(j) = '0'
          oldval(j) = -1.0
          newval(j) = -1.0
          lastwrite(j) = 'Y'
          modecnt(j) = 0
          enddo


* Read & write header (2 lines)
        read(1,'(a)')header
        write(2,'(a)')header
        read(1,'(a)')header
        write(2,'(a)')header
        if (passcnt .eq. 1) write(3,'(a,a)')
     .    ' Log file for station ',infile(1:8)

*       Read records
        i = 0   ! Current record (1-7)

        ! This will be a secondary loop, looping to read each record in a file
10      i = i + 1
        if (i .gt. 6) then
          write(3,'(a,a,i4)')dates(i-1),
     .    ' (<-prev date) ** ERROR.  Stepped off end of Array!  i=',i
          write(*,'(a,a,i4)')dates(i-1),
     .    ' (<-prev date) ** ERROR.  Stepped off end of Array!  i=',i
          write(6,'(a,a,i4)')dates(i-1),
     .    ' (<-prev date) ** ERROR.  Stepped off end of Array!  i=',i
          goto 999
          endif
        read(1,11,end=999)dates(i),fill1(i),cloudc,temp,pres,
     .          winddir,windspd,ceiling,fill2(i)
11      format(a,a,i2,f6.1,i5,i4,f6.1,i7,a)

        invar(i,1) = cloudc * 1.0         ! Put in array (convert to real #)
        invar(i,2) = temp * 1.0
        invar(i,3) = pres * 1.0
        invar(i,4) = winddir * 1.0
        invar(i,5) = windspd * 1.0
        invar(i,6) = ceiling * 1.0
```

```
* Deal with records     WgW

        do j = 1, 6                ! Do for all 6 input variables          !DO J

          ! Check all missing variables
          if (invar(i,j) .ge. nines(j)) then                               !IF C
            if ((mode(j) .eq. '0') .or. (mode(j) .eq. '9')) then  !'0,9'!IF D
              okprint(i,j) = 'N'
              modecnt(j) = modecnt(j) + 1
              if (modecnt(j) .gt. 6) then
                  okprint(i,j) = 'Y'
                  endif
              if (modecnt(j) .eq. 6) then
                  do k = 1, 6
                    okprint(k,j) = 'Y'
                    if (passcnt .eq. 1)
     .              write(3,'(a,a,a)')Dates(k),' Unable to replace ',
     .                varname(j)
                    enddo
                  mode(j) = '9'
                  endif
              goto 100
              endif                                                        !END D
            if (mode(j) .eq. 'D') then                              !'D'     !IF E
              okprint(i,j) = 'N'
              modecnt(j) = 1
              mode(j) = 'I'
              goto 100
              endif                                                        !END E
            if (mode(j) .eq. 'I') then                              !'I'     !IF F
              modecnt(j) = modecnt(j) + 1
              okprint(i,j) = 'N'
              if (modecnt(j) .eq. 6) then
                  mode(j) = '9'
                  do k = 1, 6
                    okprint(k,j) = 'Y'
                    if (passcnt .eq. 1)
     .              write(3,'(a,a,a)')dates(k),' Unable to replace ',
     .                varname(j)
                    enddo
                  endif
              goto 100
              endif                                                        !END F
* WgW
            else   ! (invar(i,j) <> .ge. nines(j))   - Deal with Data
              if ((j .eq. 6) .and. (invar(i,j) .eq. eights)) then
```

```
               invar(i,j) = sevens
               if (passcnt .eq. 1)
     .         write(3,'(a,a)')Dates(i),
     .           ' Converted Ceiling from 88888 to 77777'
               endif

            if (mode(j) .eq. 'D') then              !'D'     !IF G
               okprint(i,j) = 'Y'
               oldval(j) = invar(i,j)
               modecnt(j) = 0
               goto 100
               endif                                               !END G
            if (mode(j) .eq. '9') then              !'9'     !IF H
               modecnt(j) = 0
               mode(j) = 'D'
               okprint(i,j) = 'Y'
               oldval(j) = invar(i,j)
               goto 100
               endif                                               !END H
            if (mode(j) .eq. '0') then              !'0'     !IF I
               mode(j) = 'D'
               oldval(j) = invar(i,j)
               L = (i-modecnt(j))
               do k = L,i-1
                 invar(k,j) = oldval(j)
                 write(3,'(a,a,a)')dates(k),' Replaced ',varname(j)
                 okprint(k,j) = 'Y'
                 enddo
               okprint(i,j) = 'Y'
               modecnt(j) = 0
               goto 100
               endif                                               !END I
            if (mode(j) .eq. 'I') then   ! Interpolate!!!! !'I'     !IF K

               ! First deal with special cases for Cloud Cover
               if (j .eq. 1) then                            !   1
                 mode(j) = 'D'
                 newval(j) = invar(i,j)
                 L = (i-modecnt(j))
                 okprint(i,j) = 'Y'

                 both9 = 'N'
                 do k = L,i-1
                   if (invar(k,6) .lt. nines(6)) then
                     if (invar(k,6) .eq. sevens) then
                           write(3,'(a,a)')dates(k),
```

```
     .                  ' Replaced Cloud Cover based on Ceiling value'
                        invar(k,j) = 0
                        okprint(k,j) = 'Y'
                        else
                        write(3,'(a,a)')dates(k),
     .                  ' Replaced Cloud Cover based on Ceiling value'
                        invar(k,j) = 10
                        okprint(k,j) = 'Y'
                        endif
                    else
                    both9 = 'Y'
                    endif
                  enddo

                M = 0
                do k = L,i-1
                  if (invar(k,j) .lt. nines(j)) then
                    oldval(j) = invar(k,j)
                    M = M + 1
                    else
                    L = L + M  ! Move L value up and continue
                    goto 44
                    endif
                  enddo

44              continue
                if (both9 .eq. 'Y') then !go back & clean up missing values...
45                do k = L,i-1
                    N = 0
                    newval2 = newval(j)
                    if (invar(k,j) .lt. nines(j)) then
                        oldval(j) = invar(k,j)
                        else
                        ! See how many are missing and interpolate
                        N = 0  ! Count of missing
                        both9 = 'N'
                        do M = k,i-1
                          if ((invar(M,j) .ge. nines(j)) .and.
     .                        (both9 .eq. 'N')) then
                            N = N + 1
                            else
                            newval2 = invar(M,j)
                            both9 = 'Y'
                            endif
                          enddo
                        if (N .gt. 0) then
```

```
                      do M = k,(k+N)-1     ! Interpolate (mini)
                    P= (M-k)+1
                        invar(M,j) = oldval(j) +
     .                  (P * ((newval2 - oldval(j)) / (N+1)))
                        okprint(M,j) = 'Y'
                        write(3,'(a,a,a)')dates(M),
     .                    ' Interpolated ', varname(j)
                        enddo
                      endif
                    endif
              enddo
            if (N .gt. 0) goto 45
            endif

          goto 100
          endif
* WGW
         ! Now deal with special cases for Ceiling Height
         if (j .eq. 6) then                              !   6     !IF L
          mode(j) = 'D'
          newval(j) = invar(i,j)
          L = (i-modecnt(j))
          okprint(i,j) = 'Y'

          do k = L,i-1
            if (invar(k,1) .ge. nines(1)) then
              okprint(k,j) = 'Y'
              if (passcnt .eq. 1)
     .        write(3,'(a,a,a)')dates(k),
     .              ' Unable to replace ', varname(j)
              endif
            if (invar(k,1) .le. 5) then
              invar(k,j) = sevens
              okprint(k,j) = 'Y'
              write(3,'(a,a)')dates(k),
     .        ' Replaced Ceiling Height based on Cloud Cover'
              endif
            enddo

47        N = 0             ! Will return to 47 when unable to fill all
          do K = L, i-1             ! Now "skip" ones fixed above...
            if (okprint(k,j) .eq. 'Y') then
              N = N + 1
              oldval(j) = invar(k,j)
              else
              goto 50
```

```
        endif
      enddo

50    L = L + N   ! Move L up to first missing position
      if (L .eq. I) then        ! Then this set is finished
        modecnt(j) = 0
        oldval(j) = invar(i,j)
        goto 100
      endif

55    modecnt(j) = 0
      do k = L,i-1              ! See how many missing ahead
        if ((invar(k,j) .ge. nines(j)) .and.
     .          (invar(k,1) .lt. nines(1))) then
          modecnt(j) = modecnt(j) + 1
          newval(j) = invar(k+1,j)
          else
          goto 60
          endif
      enddo

60    continue

      if (modecnt(j) .eq. 0)
     .  write(3,'(a,a)')' Modecnt(j) = 0 *** UNEXPECTED !'
      M = L + (modecnt(j) - 1)

      if (oldval(j) .ge. nines(j)) then         ! 99 & (x,7,9)
        do k = L,M
          invar(k,j) = newval(j)
          okprint(k,j) = 'Y'
          write(3,'(a,a,a)')dates(k),
     .          ' Replaced ', varname(j)
          enddo
        oldval(j) = invar(M,j)
        goto 75
        endif

      if ((oldval(j) .eq. sevens) .and.
     .    (newval(j) .lt. nines(j))) then       ! 77 & (x,7)
        do k = L,M
          invar(k,j) = newval(j)
          okprint(k,j) = 'Y'
          write(3,'(a,a,a)')dates(k),
     .          ' Replaced ', varname(j)
          enddo
```

```
                  oldval(j) = invar(M,j)
                  goto 75
                  endif

                 if ((oldval(j) .eq. sevens) .and.
     .              (newval(j) .ge. nines(j))) then       ! 77 & 99
                  do k = L,M
                   invar(k,j) = oldval(j)
                   okprint(k,j) = 'Y'
                   write(3,'(a,a,a)')dates(k),
     .                ' Replaced ', varname(j)
                   enddo
                  oldval(j) = invar(M,j)
                  goto 75
                  endif

                 if ((oldval(j) .lt. sevens) .and.
     .              (newval(j) .ge. sevens)) then         ! x & (7,9)
                  do k = L,M
                   invar(k,j) = oldval(j)
                   okprint(k,j) = 'Y'
                   write(3,'(a,a,a)')dates(k),
     .                ' Replaced ', varname(j)
                   enddo
                  oldval(j) = invar(M,j)
                  goto 75
                  endif

                 if ((oldval(j) .lt. sevens) .and.
     .              (newval(j) .lt. sevens)) then          ! x & x
                  if (modecnt(j) .eq. 1) then   ! Replace
                   k=i-1
                   invar(k,j) = oldval(j)
                   okprint(k,j) = 'Y'
                   write(3,'(a,a,a)')dates(k),
     .                ' Replaced ', varname(j)
                   oldval(j) = invar(k,j)
                   goto 75
                   else                         ! Interpolate
                   do k = L,M
                      P = (k-L) + 1
                      invar(k,j) = oldval(j) +
     .                (p*((newval(j)-oldval(j))/(modecnt(j)+1)))
                      write(3,'(a,a,a)')dates(k),
     .                ' Interpolated ', varname(j)
                      okprint(k,j) = 'Y'
```

```
            enddo
          oldval(j) = invar(M,j)
          goto 75
          endif
        endif

75      continue
        if (M .ne. i-1) then    ! If not finished with this set,
          goto 47               !  go back to 47 and continue.
          else
          oldval(j) = invar(i,j)
          endif

        endif                                          !END L

      ! Deal with special interpolating of wind direction
      if (j .eq. 4) then                        !   4      !IF M
        mode(j) = 'D'
        L = i - modecnt(j)
        newval(j) = invar(i,j)
        if (modecnt(j) .eq. 1) then
          invar(L,j) = oldval(j)
          write(3,'(a,a,a)')dates(i-1),
     .      ' Replaced ',varname(j)
          modecnt(j) = 0
          okprint(L,j) = 'Y'
          okprint(i,j) = 'Y'
          oldval(j) = invar(i,j)
          goto 100
          endif
        if (modecnt(j) .gt. 1) then
          ! Determine which possibility we have...
          M = newval(j) - oldval(j)
          N = oldval(j) - newval(j)
          if ((M .le. 180) .and. (M .ge. 0)) then
            do k = L,i-1
              P= (k-L)+1
                invar(k,j) = oldval(j) +
     .          P * ((newval(j)-oldval(j)) / (modecnt(j)+1))
              write(3,'(a,a,a)')dates(k),
     .          ' Interpolated ',varname(j)
                intval = (invar(k,j)+5)/10
                invar(k,j) = intval * 10.0
                enddo
            endif
          if ((N .le. 180) .and. (N .ge. 0)) then
```

```
                    do k = L,i-1
                      P= (k-L)+1
                          invar(k,j) = oldval(j) -
     .                    P * ((oldval(j)-newval(j)) / (modecnt(j)+1))
                          intval = (invar(k,j)+5)/10
                          invar(k,j) = intval * 10.0
                      write(3,'(a,a,a)')dates(k),
     .                  ' Interpolated ',varname(j)
                          enddo
                    endif
                  if (M .gt. 180) then
                    if (oldval(j) .lt. newval(j)) then
                          oldval(j) = oldval(j) + 360
                          else
                          newval(j) = newval(j) + 360
                          endif
                    do k = L,i-1
                      P= (k-L)+1
                          invar(k,j) = oldval(j) +
     .                    P * ((newval(j)-oldval(j)) / (modecnt(j)+1))
                          intval = (invar(k,j)+5)/10
                          invar(k,j) = intval * 10.0
                          if (invar(k,j) .gt. 350)
     .                      invar(k,j) = invar(k,j) - 360
                          if (invar(k,j) .lt. 0)
     .                      invar(k,j) = invar(k,j) + 360
                      write(3,'(a,a,a)')dates(k),
     .                ' Interpolated ',varname(j)
                          enddo
                    endif
                  if (N .gt. 180) then
                    if (oldval(j) .lt. newval(j)) then
                          oldval(j) = oldval(j) + 360
                          else
                          newval(j) = newval(j) + 360
                          endif
                    do k = L,i-1
                      P= (k-L)+1
                          invar(k,j) = oldval(j) -
     .                    P * ((oldval(j)-newval(j)) / (modecnt(j)+1))
                          intval = (invar(k,j)+5)/10
                          invar(k,j) = intval * 10.0
                          if (invar(k,j) .gt. 350)
     .                      invar(k,j) = invar(k,j) - 360
                          if (invar(k,j) .lt. 0)
     .                      invar(k,j) = invar(k,j) + 360
```

```
                      write(3,'(a,a,a)')dates(k),
     .                 ' Interpolated ',varname(j)
                        enddo
                     endif
                   endif
                 modecnt(j) = 0
                 oldval(j) = invar(i,j)
                 do k = L, i
                   okprint(k,j) = 'Y'
                   enddo
                 endif                                        !END M

               ! Deal with the rest (more general rules)     ! 2-3,5
               if ((j .eq. 2) .or. (j .eq. 3) .or. (j .eq. 5)) then   !IF N
                 mode(j) = 'D'
                 newval(j) = invar(i,j)
                 L = (i-modecnt(j))
                 if (modecnt(j) .eq. 1) then
                   invar(L,j) = oldval(j)
                   write(3,'(a,a,a)')dates(i-1),
     .               ' Replaced ',varname(j)
                   endif
                 if (modecnt(j) .gt. 1) then
                   do k = L,i-1
                     P= (k-L)+1
                     invar(k,j) = oldval(j) +
     .               P * ((newval(j)-oldval(j)) / (modecnt(j)+1))
                     write(3,'(a,a,a)')dates(k),
     .                 ' Interpolated ',varname(j)
                     enddo
                   endif
                 do k = L, i
                   okprint(k,j) = 'Y'
                   enddo
                 oldval(j) = invar(i,j)
                 modecnt(j) = 0
                 endif                                        !END N
               goto 100
               endif                                          !END K

           endif                                              !END C

100      continue
         enddo                                                !END J

* Now if all 6 vars for record 1 are clear to print, print them.
```

```
            do M = 1, i
            if      ((okprint(1,1) .eq. 'Y') .and. (okprint(1,2) .eq. 'Y')
     .        .and. (okprint(1,3) .eq. 'Y') .and. (okprint(1,4) .eq. 'Y')
     .        .and. (okprint(1,5) .eq. 'Y') .and. (okprint(1,6) .eq. 'Y'))
     .        then
              cloudc  = (invar(1,1) * 1)
              temp    =  invar(1,2)
              pres    = (invar(1,3) * 1)
              winddir = (invar(1,4) * 1)
              windspd =  invar(1,5)
              ceiling = (invar(1,6) * 1)

              write(2,11)dates(1),fill1(1),cloudc,temp,pres,
     .                 winddir,windspd,ceiling,fill2(1)

* Now shift the records up...
              do k = 1, 6
                do j = 1, 6
                  invar(k,j) = invar(k+1,j)
                  okprint(k,j) = okprint(k+1,j)
                  enddo
                dates(k) = dates(k+1)
                fill1(k) = fill1(k+1)
                fill2(k) = fill2(k+1)
                enddo
              i = i - 1
              endif
              enddo

            goto 10

999         continue   ! deal with unwritten records before closing

            i = i -1
        do k = 1, i
          do j = 1, 5
            if ((mode(j) .eq. '9') .or. (mode(j) .eq. 'I')) then
              if ((modecnt(j) .lt. 6) .and. (invar(k,j) .ge. nines(j)))
     .               then
               invar(k,j) = oldval(j)
               write(3,'(a,a,a)')Dates(k),
     .           ' Replaced ', varname(j)
                     else
                     if (passcnt .eq. 1)
     .               write(3,'(a,a,a)')dates(k),
     .                 ' Unable to replace ', varname(j)
```

```
                  endif
               endif
             enddo
           j = 6
           if ((mode(j) .eq. '9') .or. (mode(j) .eq. 'I')) then
             if ((modecnt(j) .lt. 6) .and.
     .             (invar(k,j) .ge. nines(j))) then
               if (invar(k,1) .ge. nines(1)) then
                   oldval(j) = invar(k,j)
                   write(3,'(a,a,a)')dates(k),
     .               ' Unable to replace ',varname(j)
                   endif
               if (invar(k,1) .le. 5) then
                   invar(k,j) = sevens
                   oldval(j) = sevens
                   endif
               if (invar(k,j) .ge. 6) then
                   invar(k,j) = oldval(j)
                   write(3,'(a,a,a)')dates(k),
     .               ' Replaced ', varname(j)
                   endif
               endif
             endif

           cloudc  = (invar(k,1) * 1)      ! Convert back to Integers
           temp    =  invar(k,2)
           pres    = (invar(k,3) * 1)
           winddir = (invar(k,4) * 1)
           windspd =  invar(k,5)
           ceiling = (invar(k,6) * 1)

           write(2,11)dates(k),fill1(k),cloudc,temp,pres,
     .             winddir,windspd,ceiling,fill2(k)
           enddo

         if (passcnt .eq. 1) goto 5

5000     close (unit=1)
         close (unit=2)
         close (unit=3)

         write(*,'(a,a)')' Completed data check on ',station
         write(6,'(a,a)')' Completed data check on ',station
         goto 1

9997     write(*,'(a)')' *** Unable to open STATION.DAT file.'
```

```
      write(6,'(a)')' *** Unable to open STATION.DAT file.'
      goto 9999

9998  write(*,'(a,a)')' *** Unable to open ',station
      write(6,'(a,a)')' *** Unable to open ',station
      goto 1

9999  close (unit=4)
      write (*,'(/,a,/,a)')' METFIX: End of station data files.',
     .      ' Files completed are listed in METFIX.LOG.'
      end
```

Appendix 4C. PRECIP Program

Read STATION.DAT to get list of WBAN (MET) files to process.
Run GET_DIST

Read the MET data record by record. For each hour that rain was measured, increment the frequency count for that hour by 1.
Repeat through entire MET file until the number of times rain was measured for each hour is recorded.
For each hour (1-24) divide the counts of rain for the hour by the total incidents of rain in the entire file.
Grand total for 24 hours will be 1.0.

Use the WBANCOOP.PRN file to locate the appropriate COOP file for the MET file being processed..
Read each Met day and match it with the appropriate COOP day and process according to rules listed below.

If there is rain in the COOP file but no rain in the MET file:
use frequency distribution to spread the rain from COOP file across entire day. Start with the appropriate hour from the frequency distribution.
If there is rain in the COOP file and rain in the MET file:
distribute the rain from the COOP file into the hours of the METfile that have rain, in the proportion that already exists in the MET file.
If there is no rain in the COOP file for the day, set the MET file precipitation to zero.

If the COOP file indicates that the rain measurement is taken at 1800 hours, the current "day" is considered to run from 1900 hours the previous day until 1800 on the current day.

The program looks for and recognizes when the COOP file repeats a record. (This was a known problem with the data.)

The first day may have less than 24 hours if the COOP measurement time is not "24". Therefore, the data from the COOP file has to be squeezed into less than 24 hours. The frequency data is split into 144 units (each hour of 24 is divided by 6. The first 6 units would have one-sixth of the first hour's frequency, etc..) Then if the day has 19-24 hours, every 6 units of the 144 frequency units are placed into each hour of the day. If the day has 13-18 hours, every 8 units of the 144 frequency units are placed into each hour of the day. Similarly, 7-12 hours use 12 units and 1-6 hours use 24 units. This is not perfect, but gives a very close approximation of fitting the rain from the COOP file into the shortened MET day.

If the COOP file is using an hour other than 24 (e.g. 18), and the year ends and skips the next year, the remaining hours (e.g. 19-24) in the MET file are simply copied to the corrected MET file with zero precipitation.

If the COOP file is using an hour other than 24 (e.g. 18), and the year ends and continues to the next year, the remaining hours from the old year are continued into the first day of the new year.

The program does not change COOP measurement hour except at the beginning of each year. This was a known limitation to the program.

If the COOP measurement hour is greater than 24, or if the amount of rain is greater than 9998, the data is considered invalid or missing and the rain is set at zero.

The program RAINTOT will read the same STATION.DAT file and create an output file that compares the COOP data with the output (corrected) MET DATA. This program was used as a check to confirm that rain measurement data was accurately converted from the COOP file to the MET file.

Appendix 4D. Formats and Examples of HWIR Meteorological Data Files

Provided are the hourly, daily, monthly, annual, and long term files. The hourly data are used for the ISCST3 Air dispersion model and the rest contain climatological data for use in other models. Hourmet.zip contains 43 hourly meteorological files. The remainder of the 99 files can be found in Hourmet2.zip and Hourmet3.zip. All can be extracted using WINZIP or Pkunzip. Single files may be extracted using PKUNZIP by typing:

PKUNZIP HOURMET.zip ?????h.dat

where ????? indicates the station number and h indicates hourly data.

The length of the hourly meteorological files varies based on availability of data. The minimum length of file is 10 years. Substitutions were made when the 10 year criteria could not be met.

Climonew.zip contains the daily, monthly, annual, and long term files for 99 meteorological stations. This file may be unzipped in the same manner as hourmet.zip. Again the ????? in the file names denotes the meteorological station number. The letter indicates the type of data (d=daily, m=monthly, a=annual, and l=long term). The length of these files varies by the number of years of data available.

Hourly Meteorological File (03812h.dat, for 1 day only)

Hourly time series format(?????h.dat):

Header Record

Sfc Station #   Sfc Sta. Year   Mixing Ht. Station #   Mixing Ht. Sta. Year

Data Records (in each row)

Variable	Description	Units
001	Year	
002	Month	
003	Day	
004	Hour	
005	Random flow vector (wind dir)	(degrees)
006	Windspeed	(m/s)
007	Ambient Temperature	(K)
008	Stability Category	
009	Rural Mixing Height	(m)
010	Urban Mixing Height	(m)
011	Friction Velocity at Application Site	(m/s)

012	Monin-Obukhov Length at app. Site	(m)
013	Roughness length at application site	(m)
014	Precipitation Code	
015	Precipitation Amount	(mm)

```
  3812     73  13723     73
73 1 1 1   1.0000   5.1444 287.6 4  611.0  611.0   0.8413    1153.3  0.5419   0    .00
73 1 1 2   8.0000   3.6011 287.0 4  611.0  611.0   0.5727     304.8  0.5419   0    .00
73 1 1 3 154.0000   5.6588 285.4 4  611.0  611.0   0.9244    1036.9  0.5419   0    .00
73 1 1 4 163.0000   6.1733 284.3 4  611.0  611.0   1.0127    1651.6  0.5419   0    .00
73 1 1 5 153.0000   6.1733 283.7 4  611.0  611.0   1.0127    1648.3  0.5419   0    .00
73 1 1 6 152.0000   7.2022 283.2 4  611.0  611.0   1.1837    2247.7  0.5419   0    .00
73 1 1 7 165.0000   7.7166 281.5 4  611.0  611.0   1.2694    2690.5  0.5419   0    .00
73 1 1 8 153.0000   7.7166 280.9 4  611.0  611.0   1.2694    2693.4  0.5419   0    .00
73 1 1 9 157.0000   8.2310 280.9 4  611.0  611.0   1.3617    -999.0  0.5419   0    .00
73 1 110 151.0000   9.2599 282.6 4  611.0  611.0   1.5330    -999.0  0.5419   0    .00
73 1 111 164.0000  10.2888 283.7 4  611.0  611.0   1.7033    -999.0  0.5419   0    .00
73 1 112 156.0000   7.7166 286.5 4  611.0  611.0   1.2812    -999.0  0.5419   0    .00
73 1 113 173.0000   7.2022 287.6 4  611.0  611.0   1.1968    -999.0  0.5419   0    .00
73 1 114 159.0000   5.1444 288.7 4  611.0  611.0   0.8606    -709.1  0.5419   0    .00
73 1 115 162.0000   8.7455 288.2 4  611.0  611.0   1.4472    -999.0  0.5419   0    .00
73 1 116 164.0000   7.2022 286.5 4  611.0  611.0   1.1903    -999.0  0.5419   0    .00
73 1 117 161.0000   7.7166 284.8 4  611.0  611.0   1.2694    2693.4  0.5419   0    .00
73 1 118 157.0000   7.2022 283.2 4  604.2  604.2   1.1837    2247.7  0.5419   0    .00
73 1 119 164.0000   6.1733 282.0 4  593.3  593.3   1.0127    1638.5  0.5419   0    .00
73 1 120 157.0000   5.6588 281.5 4  582.3  582.3   0.9270    1370.2  0.5419   0    .00
73 1 121 160.0000   6.1733 280.9 4  571.3  571.3   1.0126    1631.9  0.5419   0    .00
73 1 122 162.0000   6.1733 280.4 4  560.4  560.4   1.0112    1362.9  0.5419   0    .00
73 1 123 160.0000   5.1444 280.4 4  549.4  549.4   0.8379     819.9  0.5419   0    .00
73 1 124 160.0000   4.1155 280.4 4  538.5  538.5   0.6688     710.5  0.5419   0    .00
```

Daily Meteorological File (03812d.dat, for 1 year only)

Daily time series format(?????d.dat):
jday,dailyppt,temp,ud,acc,Ed,dailyR,u1,Nu1,u2,Nu2,u4,Nu4,u6,Nu6,u8,Nu8,u10,Nu10

where,
jday=Julian day
dailyppt=total daily precip. in cm/day
temp=daily average temperature in degrees Celsius
ud=daily average windspeed in m/s
acc=daily average cloud cover in tenths
Ed=daily average evaporation in m/day
dailyR=daily rainfall erosivity
u1-u10= windspeed at 1m to 10m pile hts in m/s
Nu1-Nu10=number of hours where the windspeed > 5.4m/s at the indicated pile height.

Also note that the number of days in the given year also occurs before each year

```
365
1,0.0064,10.7,6.7,8,0.00282836,0.0124,5.1,12,5.7,12,6.3,17,6.7,19,7.0,19,7.2, 22
2,0.0000,6.6,4.2,9,0.00149589,0.000,3.1,2,3.5,4,3.9,8,4.2,10,4.4,10,4.6, 12
3,1.9050,2.2,3.2,10,0.00026872,3.709,2.4,0,2.7,0,3.0,0,3.2,0,3.4,0,3.5, 2
4,0.0508,8.1,4.2,6,0.00098084,0.0989,3.1,2,3.5,3,3.9,7,4.2,8,4.5,10,4.6, 12
5,0.1524,4.5,6.2,9,0.0007981,0.2967,4.6,10,5.2,13,5.8,18,6.2,18,6.5,18,6.8, 18
6,0.0508,2.2,7.5,10,0.00075895,0.0989,5.7,13,6.3,18,7.0,21,7.5,22,7.8,22,8.1, 23
7,1.2954,-1.6,4.3,10,0.00043816,2.5221,3.2,0,3.6,0,4.0,0,4.3,3,4.5,3,4.6, 8
8,0.4826,-4.0,3.5,10,0.00025049,0.9396,2.7,0,3.0,0,3.3,1,3.5,2,3.7,2,3.8, 3
9,0.0000,-3.5,7.3,7,0.00071656,0.000,5.5,15,6.1,18,6.8,21,7.2,21,7.6,22,7.8, 22
10,0.0000,-4.3,3.2,10,0.00038367,0.000,2.3,1,2.6,2,2.9,3,3.2,3,3.3,3,3.5, 3
11,0.0000,-3.5,6.5,4,0.00099035,0.000,5.0,8,5.5,11,6.1,14,6.5,16,6.8,16,7.0, 17
12,0.0000,-5.2,7.2,5,0.00100345,0.000,5.4,15,6.0,16,6.7,17,7.2,19,7.5,19,7.8, 21
13,0.0000,-5.8,1.1,1,0.0002295,0.000,0.6,0,0.8,0,1.0,0,1.1,0,1.3,0,1.4, 0
14,0.0000,-2.0,1.9,3,0.00039707,0.000,1.2,0,1.4,0,1.7,0,1.9,0,2.0,0,2.2, 0
15,0.0000,1.5,6.1,2,0.00100777,0.000,4.3,10,4.9,11,5.6,12,6.1,15,6.5,15,6.8, 16
16,0.0000,2.2,2.0,1,0.00043629,0.000,1.2,0,1.4,0,1.8,0,2.0,0,2.2,0,2.4, 1
17,0.0000,4.2,2.1,7,0.00055464,0.000,1.2,0,1.5,0,1.8,0,2.1,0,2.3,0,2.4, 0
18,0.0000,6.8,3.2,8,0.00073528,0.000,2.2,1,2.5,4,2.9,5,3.2,6,3.4,6,3.5, 6
19,0.3810,10.0,5.0,5,0.00158616,0.7418,3.8,6,4.2,8,4.7,10,5.0,11,5.3,13,5.5, 13
20,0.0000,6.9,5.8,4,0.00196173,0.000,4.1,10,4.7,11,5.3,12,5.8,14,6.1,15,6.4, 15
.
.
.
.
.
.
344,0.0064,0.1,4.8,4,0.0007867,0.0124,3.5,9,3.9,9,4.5,10,4.8,11,5.1,11,5.3, 11
345,0.0064,-3.9,6.2,3,0.00092978,0.0124,4.5,12,5.1,14,5.7,15,6.2,16,6.5,17,6.8, 18
346,0.0000,3.1,2.1,7,0.00067219,0.000,1.4,0,1.6,1,1.9,1,2.1,2,2.2,2,2.3, 2
347,0.2794,6.4,4.6,8,0.0007463,0.544,3.4,3,3.8,5,4.3,7,4.6,8,4.8,8,5.0, 12
348,0.0000,3.0,6.4,7,0.00110924,0.000,4.7,13,5.3,14,5.9,15,6.4,17,6.7,18,7.0, 18
349,0.3810,2.5,3.3,10,0.00018992,0.7418,2.4,0,2.7,0,3.1,0,3.3,1,3.5,1,3.7, 5
350,0.1270,1.0,9.2,10,0.00078427,0.2473,7.0,21,7.8,22,8.7,24,9.2,24,9.6,24,9.9, 24
351,0.0064,-4.6,9.3,3,0.00115562,0.0124,7.0,17,7.8,20,8.7,21,9.3,21,9.7,23,10.1, 24
352,0.0000,-2.7,4.1,5,0.00075461,0.000,3.0,3,3.3,3,3.8,4,4.1,4,4.3,4,4.5, 6
353,0.0762,-0.9,4.4,7,0.0008400,0.1484,3.0,1,3.5,3,4.0,4,4.3,6,4.6,6,4.8, 7
354,3.5814,2.8,3.7,10,0.00008341,6.9729,2.8,1,3.1,4,3.5,4,3.7,4,3.8,4,4.0, 6
355,0.0762,-3.4,10.9,5,0.00146761,0.1484,8.3,24,9.2,24,10.2,24,10.9,24,11.4,24,11.8, 24
356,0.0000,-3.2,4.8,2,0.00079436,0.000,3.3,4,3.8,4,4.4,6,4.8,7,5.1,9,5.4, 12
357,0.0000,2.9,3.2,4,0.00068304,0.000,2.1,0,2.5,1,2.9,1,3.2,1,3.4,1,3.6, 3
358,0.0000,4.5,3.3,6,0.00060765,0.000,2.3,0,2.6,0,3.0,1,3.3,2,3.5,2,3.7, 7
359,2.0320,4.3,4.4,10,0.00017778,3.9562,3.4,0,3.7,0,4.1,2,4.4,4,4.6,4,4.8, 6
360,5.2578,13.3,5.8,9,0.00065068,10.2367,4.3,8,4.8,8,5.4,9,5.8,10,6.0,10,6.3, 11
361,0.0000,8.7,4.4,1,0.00150473,0.000,3.0,2,3.5,3,4.0,7,4.4,8,4.6,9,4.9, 9
362,0.0000,3.9,4.4,0,0.00119788,0.000,3.1,3,3.5,4,4.0,6,4.4,7,4.6,8,4.9, 8
363,0.3048,6.2,4.1,6,0.00075381,0.5934,3.0,7,3.4,9,3.8,9,4.1,9,4.4,9,4.5, 9
364,0.0254,7.4,5.4,8,0.0010629,0.0495,4.1,5,4.6,8,5.0,10,5.3,10,5.6,10,5.8, 14
365,4.1402,8.9,4.3,10,0.00002162,8.0608,3.3,2,3.6,2,4.0,4,4.3,4,4.5,4,4.6, 7
```

Monthly Meteorological File (03812m.dat, for 1 year only)

Monthly time series format(?????m.dat):
month,AvgTemp,um,p,E,mcc,maxtemp,mintemp

where,
month format is mo1yr1-mo12yr1.... ex. 0161-1261,0162-1262
AvgTemp=monthly mean temperature in degrees Celsius
um=monthly mean windspeed in m/s
p=monthly average precip. in m/day
E=monthly average evaporation in m/day
mcc=monthly average cloud cover in tenths
maxtemp=maximum daily average temperature for month in degrees Celsius
mintemp=minimum daily average temperature for month in degrees Celsius

```
173,2.45,4.46,.00349,0.0009,6,10.7,-5.8
273,2.95,4.61,.00385,0.00092,6,9.8,-5.7
373,10.92,4.15,.00731,0.00115,8,17.5, 4
473,11.52,4.33,.00484,0.00157,7,18.9, 2.7
573,15.64,4.19,.00724,0.00202,6,20.9, 9.7
673,21.12,2.88,.00329,0.00168,7,23, 17.9
773,22.58,2.86,.00571,0.00176,7,24.4, 20.1
873,22.45,2.72,.00375,0.00181,7,24.5, 18.6
973,20.28,2.77,.00265,0.00131,7,23.7, 15.3
1073,13.85,3.14,.00198,0.0011,6,19.1, 6
1173,9.13,4.05,.00303,0.00114,5,18,-0.4
1273,3.93,4.88,.00695,0.00088,6,14.1,-4.6
```

Annual Meteorological File (03812a.dat)

Annual average time series format(?????a.dat):
year,u,fw1,fw2,fw4,fw6,fw8,fw10,Uplus,p_days,PE,Annppt,Annevap,Anntemp

where,
year is in the format yr ex. 61 or 62
u=annual mean windspeed in m/s
fw1 to fw10=% time u1 to u10 are > 5.4m/s at respective pile heights
Uplus=annual average fastest mile of wind in m/s
p_days=Days per yr. with precipitation > 0.01 in. in days/yr.
PE=Thornthwaite Precipitation-Evaporation index
Annppt=annual precipitation in cm/year
Annevap=annual evaporation in cm/year
Anntemp=annual average temperature in degrees Celsius

```
61,3.686,5,9,12,16,16,23,6.605,101,0.5386,118.2574,81.61317,17.676
62,3.539,5,10,14,18,18,25,6.654,108,0.4309,122.8682,85.83046,18.069
63,3.372,5,10,14,17,17,24,6.581,89,0.3395,118.9754,78.99883,17.385
64,3.428,7,11,15,18,18,25,6.743,108,0.4123,134.5014,70.06808,17.36
82,3.312,1,5,8,11,11,16,6.158,103,0.5345,124.0428,67.04747,18.385
84,3.162,2,6,8,11,11,15,5.978,102,0.3783,111.9454,73.86912,18.204
```

85,3.322,2,5,8,11,11,16,5.903,95,0.3636,91.4736,74.92225,18.058
86,2.992,1,3,4,7,7,10,5.502,91,0.3899,97.9630,75.88924,18.818
87,2.919,1,2,4,6,6,10,5.391,91,0.5304,96.9912,68.1815,17.855
88,2.851,0,1,3,5,5,7,5.314,102,0.3785,111.1522,67.29345,17.31
89,2.723,0,1,3,4,4,7,5.071,106,0.344,130.9834,61.21514,17.616
90,3.035,1,3,5,8,8,11,5.626,81,0.4221,91.6514,80.00489,19.158

Long Term Meteorological File (03812l.dat)

Long term annual format(?????l.dat):
Uplusp,up,pp,Ap,AE,Rann,Ltemp,tsc,lcc

where,
Uplusp=Long term average annual fastest mile of wind in m/s
up=Long term mean annual windspeed in m/s
pp=mean annual days per year with >0.01 in of precipitation in days
Ap=Long term average annual precipitation in cm/yr
AE=Long term average annual evaporation in cm/yr
Rann=average annual USLE Rainfall erosivity factor
Ltemp=Long term average annual temperature
tsc=Long term average temperature of the soil column
lcc=Long term average cloud cover in tenths

6.829,3.452,111,115.57,48.117,120,12.6,12.6, 5

Appendix 4E. HWIR Climatological Data Collection Programs

Project1.vbp (Form1.frm)

Project1 is used to extract data from SAMSON CD using ZCAT program available on the CD. There are three modules in the program.

```
Sub dummy(yr)
 'stn = 14847

        If stn = "1000" Then
            End
        End If
            k = k + 1
            yr = yr + k

            If k = 1 Then
                    MkDir ("\\Pooh\Metdata\" & stn)
           End If

            DosKeys = "zcat -d r:\data\" & stn & "\" & stn & "_" & yr & ".z >\\Pooh\Metdata\" & stn
& "\" & stn & "-" & yr & ".dat"


          progname = "Project1 - Microsoft Visual Basic [run]"
          Clipboard.Clear
          Clipboard.SetText DosKeys + Chr$(13)  ' Append a <CR>.
          DosTitle = "MS-DOS Prompt"

          AppActivate DosTitle, False
          SendKeys "% ep", 1

          AppActivate progname

          If k = 30 Then
            Input #1, x
            stn = x
          k = 0
        End If
    End Sub
```

```
Private Sub Form_Load()

    Open "D:\stn_3.txt" For Input As #1
    Input #1, stn
```

```
        Timer1.Interval = 5000
    End Sub
```

```
    Private Sub Timer1_Timer()
        Label1.Caption = Time
                'Ensure that progname is set to the titlebar of Visual Basic while running.
         yr = 60
        dummy (yr)
  End Sub
```

```
 Dim progname As String
  Public yr, k As Integer
  Public x As String
  Public stn As String
```

Project4.vbp (Form1.frm)

This program reads the extracted SAMSON files and creates files which are used as input for .MET files (generated by Paul and Chengwei).

```
Private Sub Command1_Click()

Dim h As String
Dim stn As String
Dim yer As Integer



   filez = "\\Pooh\MetData\"

   Open "d:\stn_3.txt" For Input As #1

Do While Not EOF(1)

   Input #1, stn

   MkDir (filez & stn & "\" & "MetFile")


   For yer = 61 To 90
      chanel = yer + 1
      filename = filez & stn & "\" & stn & "-" & yer & ".dat"
      outfile = filez & stn & "\" & "MetFile\" & stn & "-" & yer & ".dat"
      Open filename For Input As #yer
      Open outfile For Output As #chanel
      Input #yer, aa, bb
      Print #chanel, "~" & aa; Tab(8); bb; " "
      Print #chanel, "~YR MO DA HR I   7   8   11  12  13  15  16      21"

      Do While Not EOF(yer)
         Input #yer, yr, Mnth, dy, Hr, e, f, g, h

         Mnth = Trim(Mnth)
         dy = Trim(dy)
         Hr = Trim(Hr)
```

```
            If Mnth <= 9 Then
               Mnth = " " & Mnth
            End If

            If dy <= 9 And dy < 98 Then
               dy = "  " & dy
            ElseIf dy <= 99 Then
               dy = " " & dy
            End If

            If Hr <= 9 Then
               Hr = " " & Hr
            End If

            opaque = Mid(h, 23, 2)
            temp = Mid(h, 26, 5)
            pressure = Mid(h, 42, 4)
            wind_dir = Mid(h, 46, 4)
            wind_speed = Mid(h, 50, 5)
            ceiling_ht = Mid(h, 61, 6)
            precip_type = Mid(h, 69, 9)
            precip_amount = Mid(h, 101, 1)

            Print #chanel, yr; Mnth; dy; Tab(11); Hr; Tab(14); "0"; Tab(16); opaque; Tab(19); temp;
Tab(25); pressure; Tab(29); wind_dir; Tab(34); wind_speed; Tab(40); ceiling_ht; Tab(47);
precip_type; Tab(62); precip_amount


         Loop
            Close #yer
            Close #chanel

      Next yer

   Loop

   End
   End Sub
```

Daily1.Vbp (daily1.frm)

This program is to used to generate an intermediate file which reads data from .MET files.

Private Sub Command1_Click()

```
Dim temp As Double
Dim stn As String
Dim pr_stn As String

dir_name = "\\Pooh\MetData\"
Open "d:\samson\Stn_to_do.txt" For Input As #10

Do While Not EOF(10)
   Input #10, stn, pr_stn
   Open dir_name & stn & "\" & stn & "tp.dat" For Output As #2
```

```
    Open dir_name & "MetFiles\" & stn & ".met" For Input As #1

    Line Input #1, a

    Do While Not EOF(1)


        Line Input #1, a

        If a = "" Then
          If EOF(1) Then Exit Do
          Line Input #1, a
        ElseIf Mid(a, 1, 1) = " " Then
          zz = 0
        Else

          yr = Trim(Mid(a, 1, 2))
          mth = Trim(Mid(a, 3, 2))
          dy = Trim(Mid(a, 5, 2))
          hr = Trim(Mid(a, 7, 2))
          wind_sp = Mid(a, 19, 8)
          temp1 = Mid(a, 28, 5)
          temp = temp1 - 273.15
          temp = Format(temp, "#.000#")

          stab_class = Trim(Mid(a, 34, 1))
          precip = Mid(a, 83, 4)

          Print #2, yr, mth, dy, hr, wind_sp, temp, stab_class, precip
        End If
    Loop

    Close #2
    Close #1
  Loop
  End

  End Sub
```

Correct.vbp (Correct.frm)

Missing data for relative humidity and cloud cover are identified and corrected by this program. This program generates a corrected data file.

```
Private Sub Command1_Click()
Dim ccz(1000000), rhz(1000000), new_rh(1000000), new_cc(100000)
Dim rhx(100), ccp(1000), daily_rh(100), daily_cc(100)
Dim rh_diff As Double
Dim counter As Integer
Dim stn As String
Dim pr_stn As String

Filez = "\\Pooh\Metdata\"

Open "d:\samson\stn_to_do.txt" For Input As #11

Do While Not EOF(11)
```

```
        Input #11, stn, pr_stn

        Open Filez & "\MetFiles\" & stn & "y.OUT" For Input As #1

        Open Filez & stn & "\" & stn & "rc.dat" For Output As #3

        Input #1, yer, dy1
        rh_diff = 0

        For i = 61 To 90
           i = yer
           Open Filez & stn & "\" & stn & "-" & i & ".dat" For Input As #4
           Input #4, aa, dd
           Do While Not EOF(4)

              Input #4, yr, mnth, dy, Hr, e, f, g, h
              cc = Mid(h, 19, 3)
              cc1 = Trim(cc)
              rh = Mid(h, 37, 4)
              rh1 = Trim(rh)

              Print #3, yr, mnth, dy, rh1, cc1
           Loop
           If EOF(1) Then Exit For
           Input #1, yer, dy1
           Close #4
        Next i
        Close #1
        Close #3
        Close #4

Loop

Close #11
Open "d:\samson\stn_to_do.txt" For Input As #11

Do While Not EOF(11)
        Input #11, stn, pr_stn

        i = 0
        Open Filez & stn & "\" & stn & "rc.dat" For Input As #1
        Open Filez & stn & "\" & stn & "zx.dat" For Output As #2

        Do While Not EOF(1)

           Input #1, yr, mth, dy, rhz(i), cc

           If rhz(i) = 999 Then
              j = i
              Do While rhz(i) = 999

              If EOF(1) Then
                 X1 = (rhz(i - 2) + rhz(i - 1)) / 2
                 X2 = 0
```

```
                rh_diff = X1 - X2
                If rh_diff = 0 Then rh_diff = 0
                Print #2, yr, mth, dy, counter, X1, X2, rh_diff, rhz(i - 1), rhz(j)
                Exit Do
            End If
                Input #1, yr, mth, dy, rhz(j), cc
                counter = counter + 1
                If rhz(j) <> 999 Then Exit Do
                j = j + 1
            Loop

                If EOF(1) Then Exit Do
                Input #1, yr, mth, dy, rhzx, cczx
                X1 = (rhz(i - 3) + rhz(i - 2)) / 2
                X2 = (rhz(j) + rhzx) / 2
                rh_diff = X1 - X2
                If rh_diff = 0 Then rh_diff = 0
                Print #2, yr, mth, dy, counter, X1, X2, rh_diff, rhz(i - 1), rhz(j)
                'Exit Do

                counter = 0


        End If
        'Print #2, yr, mth, dy, new_rh, cc
        i = i + 1

    Loop

    Close #1
    Close #2
Loop

Close #11
Open "d:\samson\stn_to_do.txt" For Input As #11

Do While Not EOF(11)

Input #11, stn, pr_stn
    i = 0
    j = 0
    Open Filez & stn & "\" & stn & "rc.dat" For Input As #1
    Open Filez & stn & "\" & stn & "zxc.dat" For Output As #2

    Do While Not EOF(1)

        Input #1, yr, mth, dy, rhzx, ccz(i)

        If ccz(i) = 99 Then
            Do While ccz(i) = 99
                If EOF(1) Then
                    xc1 = (ccz(i - 3) + ccz(i - 2)) / 2
                    Xc2 = 0
                    cc_diff = Xc2 - xc1
```

```
                    Print #2, yr, mth, dy, counter, cc_diff, Xc2, xc1, ccz(i - 1), ccz(j)
                    Exit Do
                End If
                j = i
                Input #1, yr, mth, dy, rhzx, ccz(j)
                counter = counter + 1
                If ccz(j) <> 99 Then Exit Do
                j = j + 1
            Loop
                If EOF(1) Then Exit Do
                Input #1, yr, mth, dy, rhzx, cczx

                xc1 = (ccz(i - 2) + ccz(i - 1)) / 2
                Xc2 = (ccz(j) + cczx) / 2
                cc_diff = Xc2 - xc1

                Print #2, yr, mth, dy, counter, cc_diff, Xc2, xc1, ccz(i - 1), ccz(j)
            counter = 0

            'If EOF(1) Then Exit Do
        End If
        'Print #2, yr, mth, dy, new_rh, cc
        i = i + 1
    Loop
    Close #1
    Close #2
Loop

Close #11
Open "d:\samson\stn_to_do.txt" For Input As #11

Do While Not EOF(11)
Input #11, stn, pr_stn
counter = 0
    Open Filez & stn & "\" & stn & "rc.dat" For Input As #1
    Open Filez & stn & "\" & stn & "zx.dat" For Input As #2
    Open Filez & stn & "\" & stn & "rc1.dat" For Output As #3
    rh_diff = 0

        Do While Not EOF(1)

            Input #1, yr, mth, dy, rhz1, cc

            If rhz1 <> 999 Then
                Print #3, yr, mth, dy, rhz1, cc
            ElseIf rhz1 = 999 Then


                Input #2, yr, mth, dy, counter, X1, X2, rh_diff, rhz1, rrx

                If counter <> 0 Then

                    For c = 1 To counter
                        If X1 < X2 And rh_diff < 0 Then
                            If c = 1 Then
```

```
                        new_rh(c) = X1 - (rh_diff / counter)

                    Else
                        new_rh(c) = new_rh(c - 1) - (rh_diff / counter)
                    End If

                ElseIf X1 > X2 And rh_diff < 0 Then
                    If c = 1 Then
                        new_rh(c) = X2 + (rh_diff / counter)
                    Else
                        new_rh(c) = new_rh(c - 1) + (rh_diff / counter)
                    End If
                ElseIf X1 > X2 And rh_diff > 0 Then
                    If c = 1 Then
                        new_rh(c) = X2 + (rh_diff / counter)
                    Else
                        new_rh(c) = new_rh(c - 1) + (rh_diff / counter)
                    End If
                ElseIf X1 < X2 And rh_diff > 0 Then
                    If c = 1 Then
                        new_rh(c) = X1 - (rh_diff / counter)
                    Else
                        new_rh(c) = new_rh(c - 1) - (rh_diff / counter)
                    End If
                ElseIf rh_diff = 0 Then
                    If c = 1 Then
                        new_rh(c) = X1
                    Else
                        new_rh(c) = (X1 + new_rh(c - 1)) / 2
                    End If
                End If

                If c = 1 Then
                    Print #3, yr, mth, dy, new_rh(c), cc
                Else
                    Input #1, yr, mth, dy, rh11, cc
                    Print #3, yr, mth, dy, new_rh(c), cc
                End If
            Next c
        ElseIf counter = 0 Then
            rh_diff1 = rh_diff / 2
            Print #3, yr, mth, dy, Abs(rh_diff1), cc
        End If
    End If

  Loop


Close #1
Close #2
Close #3
Loop

Close #11
Open "d:\samson\stn_to_do.txt" For Input As #11
```

```
Do While Not EOF(11)

  Input #11, stn, pr_stn

    Open Filez & stn & "\" & stn & "rc1.dat" For Input As #4
    Open Filez & stn & "\" & stn & "zxc.dat" For Input As #5
    Open Filez & stn & "\" & stn & "rhc.dat" For Output As #6
    rh_diff = 0

    Do While Not EOF(4)

       Input #4, yr, mth, dy, rhz1, cc

      If cc <> 99 Then
        Print #6, yr, mth, dy, rhz1, cc
      ElseIf cc = 99 Then
        If EOF(5) Then
          Print #6, yr, mth, dy, rhz1, cc
          Exit Do
        End If
        Input #5, yr, mth, dy, counter, cc_diff, Xc2, xc1, cc1, cc2
        If counter <> 0 Then

          For c = 1 To counter
            If xc1 < Xc2 And cc_diff < 0 Then
              If c = 1 Then
                new_cc(c) = Abs(xc1 - (cc_diff / counter))
                'AVA = (new_rh(c) + rhz1) / 2
              Else
                new_cc(c) = Abs(new_cc(c - 1) - (cc_diff / counter))
              End If

            ElseIf xc1 > Xc2 And cc_diff < 0 Then
              If c = 1 Then
                new_cc(c) = Abs(Xc2 + (cc_diff / counter))
              Else
                new_cc(c) = Abs(new_cc(c - 1) + (cc_diff / counter))
              End If
            ElseIf xc1 > Xc2 And cc_diff > 0 Then
              If c = 1 Then
                new_cc(c) = Abs(Xc2 + (cc_diff / counter))
              Else
                new_cc(c) = Abs(new_cc(c - 1) + (cc_diff / counter))
              End If
            ElseIf xc1 < Xc2 And cc_diff > 0 Then
              If c = 1 Then
                new_cc(c) = Abs(xc1 - (cc_diff / counter))
              Else
                new_cc(c) = Abs(new_cc(c - 1) - (cc_diff / counter))
              End If
            ElseIf rh_diff = 0 Then
              If c = 1 Then
                new_cc(c) = Abs(xc1)
              Else
```

```
                new_cc(c) = Abs(new_cc(c - 1) + xc1) / 2
              End If

            End If

            If c = 1 Then
              Print #6, yr, mth, dy, rhz1, Int(new_cc(c))
            Else
              Input #4, yr, mth, dy, rh11, cc
              Print #6, yr, mth, dy, rh11, Int(new_cc(c))
            End If
          Next c

        ElseIf counter = 0 Then
          cc_diff1 = cc_diff / 2
          Print #6, yr, mth, dy, rhz1, Abs(Int(cc_diff1))
        End If
      End If

    Loop
    Close #5
    Close #4
    Close #6
Loop

Close #11
Open "d:\samson\stn_to_do.txt" For Input As #11

Do While Not EOF(11)

  Input #11, stn, pr_stn


    Open Filez & stn & "\" & stn & "rhc.dat" For Input As #7
    Open Filez & stn & "\" & stn & ".dat" For Output As #22
    Do While Not EOF(7)

      For i = 1 To 24
        Input #7, yr, mnth, dy, rhx(i), ccp(i)
        daily_rh(i) = daily_rh(i - 1) + rhx(i)
        daily_cc(i) = daily_cc(i - 1) + ccp(i)
      Next i

      dailyrh_av = (daily_rh(24) / 24)
      dailycc_av = CInt(daily_cc(24) / 24)

      dailyrh_av1 = Format(dailyrh_av, "#0.0")
      Print #22, yr, mnth, dy, dailyrh_av1, dailycc_av

    Loop

    Close #7
    Close 22
Loop
```

```
End
End Sub
```

5. Deadly.vbp (daily_f.frm)

This program combines Coop precipitation data files, intermediate files created from .MET files and corrected files to generate the daily data files from hourly data.

Private Sub Command1_Click()

```
Dim daily_wind_speed(100), wind_speed1(100), wind_spd(100)
Dim daily_temp(100), temp1(100), tot_pre(1000000), temp(100)
Dim daily_precip(100), precip1(100), vap(100)
Dim u2(100), step_1(100), u(100), u1(100), u4(100), u6(100), u8(100), u10(100)
Dim anemom, Rusle As Single
Dim nu1, nu2, nu4, nu6, nu8, nu10 As Single
Dim met_stn As Long
Dim stn As String
Dim pr_stn As String
Dim st1 As String

dir_name = "\\Pooh\MetData\"
dr_name = "\\tigger\cea1\Pollut~1\Projects\6720-06a\"
Open "d:\samson\stn_to_do.txt" For Input As #290
Do While Not EOF(290)

  Input #290, stn, pr_stn

    i = 0
    z = ","


      nu1 = 0
      nu2 = 0
      nu4 = 0
      nu6 = 0
      nu8 = 0
      nu10 = 0
    Open dr_name & "coop\process1\processed\" & stn & "p.PRP" For Input As #11
    Open "d:\samson\anemom.dat" For Input As #20
    Open "d:\samson\AnnRusle.prn" For Input As #40
    Open dir_name & stn & "\" & stn & "tp.dat" For Input As #60
    Open dir_name & stn & "\" & stn & "d.dat" For Output As #5
    Open dir_name & stn & "\" & stn & ".dat" For Input As #99

      Counter = 1
      j = 1
      Do While Not EOF(11)

        Input #11, yr1, mth1, preci
        tot_pre(j) = tot_pre(j - 1) + preci
        j = j + 1
      Loop

      Close #11
```

```
      Do While Not EOF(20)
          Input #20, met_stn, anemom_file, moist

          If met_stn = stn Then
            anemom = anemom_file
            Exit Do
          End If

      Loop



      Do While Not EOF(40)
          Input #40, met_stn, Rusle_file

          If met_stn = stn Then
            Rusle = Rusle_file
            rtot = Int((j - 1) / 365) * Rusle
            Exit Do
          End If

      Loop

      Open dr_name & "coop\Process1\Processed\" & stn & "p.PRP" For Input As #11

      Open dir_name & "MetFiles\" & stn & "y.OUT" For Input As #56

    Do While Not EOF(60)
      'If EOF(99) Then Exit Do
      Input #99, yr, m, dd, Rh, ccover

      For x = 1 To 24

        Open "d:\samson\wind.dat" For Input As #10

        Input #60, yr1, mth1, dy1, hr1, wind_speed1(x), temp1(x), stab_cat, precip1(x)

        wind_spd(x) = wind_speed1(x)
        daily_wind_speed(x) = daily_wind_speed(x - 1) + wind_speed1(x)
        daily_temp(x) = daily_temp(x - 1) + temp1(x)
        daily_precip(x) = (daily_precip(x - 1) + precip1(x))

        Do While Not EOF(10)

          Input #10, id, stab_class, pi_file

          If stab_class = stab_cat Then
            pi = pi_file
            Exit Do
          End If

        Loop

        step_1(x) = (wind_speed1(x) * ((1 / anemom) ^ pi))
        u1(x) = u1(x - 1) + step_1(x)
```

```
    If step_1(x) > 5.4 Then nu1 = nu1 + 1

    step_1(x) = (wind_speed1(x) * ((2 / anemom) ^ pi))
    u2(x) = u2(x - 1) + step_1(x)

    If step_1(x) > 5.4 Then nu2 = nu2 + 1

    step_1(x) = (wind_speed1(x) * ((4 / anemom) ^ pi))
    u4(x) = u4(x - 1) + step_1(x)

    If step_1(x) > 5.4 Then nu4 = nu4 + 1

    step_1(x) = (wind_speed1(x) * ((6 / anemom) ^ pi))
    u6(x) = u6(x - 1) + step_1(x)

    If step_1(x) > 5.4 Then nu6 = nu6 + 1

    step_1(x) = (wind_speed1(x) * ((8 / anemom) ^ pi))
    u8(x) = u8(x - 1) + step_1(x)

    If step_1(x) > 5.4 Then nu8 = nu8 + 1

    step_1(x) = (wind_speed1(x) * ((10 / anemom) ^ pi))
    u10(x) = u10(x - 1) + step_1(x)

    If step_1(x) > 5.4 Then nu10 = nu10 + 1

    Close #10

Next x

daily_wind_speed_av = daily_wind_speed(24) / (24)

daily_temp_av = daily_temp(24) / 24


u2_final = u2(24) / 24
u1_final = u1(24) / 24
u4_final = u4(24) / 24
u6_final = u6(24) / 24
u8_final = u8(24) / 24
u10_final = u10(24) / 24

    Open "d:\samson\tempeva.dat" For Input As #30
    p = 1
Do While Not EOF(30)


        Input #30, temp(p), vap(p)

        If daily_temp_av > temp(p - 1) And daily_temp_av < temp(p) Then

            vap_diff = vap(p) - vap(p - 1)
            temp_diff = temp(p) - daily_temp_av
```

```
                temp10 = temp(p) - temp(p - 1)
                vap1 = (vap_diff * temp_diff) / temp10
                vap_pres = vap(p) - vap1
                Exit Do
            End If

            p = p + 1

    Loop
        Close #30

    Input #11, y, m, precipita
        ea = vap_pres * Rh / 100
        rh1 = (100 - Rh) / 100
        daily_evapo = ((0.013 + 0.00016 * 86.4 * u2_final) * ea * rh1) / 100
        per_ppt = Val(precipita) / tot_pre(j - 1)
        daily_R = per_ppt * rtot

        daily_precipitation = Val(precipita)
        daily_precip1 = Format(daily_precipitation, "##0.0000")
        daily_temp_av1 = Format(daily_temp_av, "##0.0")
        daily_wind_speed_av1 = Format(daily_wind_speed_av, "##.0")
        daily_evapo1 = Format(daily_evapo, "0.0000000#")
        daily_R1 = Format(daily_R, "0.000#")
        u1_final1 = Format(u1_final, "#0.0")
        u2_final1 = Format(u2_final, "#0.0")
        u4_final1 = Format(u4_final, "#0.0")
        u6_final1 = Format(u6_final, "#0.0")
        u8_final1 = Format(u8_final, "#0.0")
        u10_final1 = Format(u10_final, "#0.0")

    Print #5, Counter & z; daily_precip1 & z; daily_temp_av1 & z; daily_wind_speed_av1 & z;
ccover & z; daily_evapo1 & z; daily_R1 & z; u1_final1 & z; nu1 & z; u2_final1 & z; nu2 & z;
u4_final1 & z; nu4 & z; u6_final1 & z; nu6 & z; u8_final1 & z; nu8 & z; u10_final1 & z; nu10


    If Counter = 365 Then
        Input #56, yr, dys

        If Counter = dys Then
            Counter = 1
        Else
            Counter = 366
        End If
    ElseIf Counter = 366 Then
        Counter = 1
    Else
        Counter = Counter + 1
    End If

    nu1 = 0
    nu2 = 0
    nu4 = 0
    nu6 = 0
    nu8 = 0
```

```
            nu10 = 0

        Loop

        Close #5
        Close #10
        Close #20
        Close #99
        Close #40
        Close #50
        Close #60
        Close #11
        Close #56
    Loop
    Close #290
    End
    End Sub
```

Monthly.vbp (Monthly.frm)

This program reads daily file data to generate monthly data file.

```
Private Sub Command1_Click()
Dim stn As String
Dim pr_stn As String

Dim month_temp(100), month_wind(100), month_precip(100), month_evapo(100), month_cc(100),
temp(190)

dir_name = "\\Pooh\MetData\"
z = ","

Open "d:\samson\Stn_to_do.txt" For Input As #10

Do While Not EOF(10)

    Input #10, stn, pr_stn

    Open dir_name & stn & "\" & stn & "d.dat" For Input As #1
    Open dir_name & "MetFiles\" & stn & "y.OUT" For Input As #2
    Open dir_name & stn & "\" & stn & "m.dat" For Output As #4

    Do While Not EOF(2)
        Input #2, yer, dys

        comp = yer Mod 4

        For mth = 1 To 12

            If mth = 1 Or mth = 3 Or mth = 5 Or mth = 7 Or mth = 8 Or mth = 10 Or mth = 12 Then
                k = 31
            ElseIf mth = 4 Or mth = 6 Or mth = 9 Or mth = 11 Then
                k = 30
            ElseIf mth = 2 And comp = 0 Then
                k = 29
            Else: k = 28
```

```
        End If

        For i = 1 To k
          Input #1, jdy, precip, temp(i), wind, cc, evapo, erosivity, u1, nu1, u2, nu2, u4, nu4, u6,
nu6, u8, nu8, u10, nu10
          month_temp(i) = month_temp(i - 1) + temp(i)

          If temp(i) > max_temp Then
            max_temp = temp(i)
          End If

          If i = 1 Then
            min_temp = temp(i)
          ElseIf temp(i) < temp(i - 1) Then
            If min_temp > temp(i) Then
              min_temp = temp(i)
            End If
          End If

          month_wind(i) = month_wind(i - 1) + wind
          month_precip(i) = month_precip(i - 1) + precip
          month_evapo(i) = month_evapo(i - 1) + evapo
          month_cc(i) = month_cc(i - 1) + cc

        Next i

        month_temp_av = month_temp(k) / k
        month_wind_av = month_wind(k) / k
        month_precip_av = (month_precip(k) / k) / 100
        month_evapo_av = (month_evapo(k) / k)
        month_cc_av = Int(month_cc(k) / k) + 1

        month_temp_av1 = Format(month_temp_av, "#.0#")
        month_wind_av1 = Format(month_wind_av, "#.0#")
        month_precip_av1 = Format(month_precip_av, "#.0000#")
        month_evapo_av1 = Format(month_evapo_av, "0.0000#")

        Print #4, mth & yer & z; month_temp_av1 & z; month_wind_av1 & z; month_precip_av1 &
z; month_evapo_av1 & z; month_cc_av & z; max_temp & z; min_temp

        min_temp = 0
        max_temp = 0

      Next mth

    Loop

    Close #1
    Close #2
    Close #4
Loop
Close #10
End
```

```
End Sub
```

Annual.vbp (Annual.frm)

This program reads several files to generate annual data file.

```
Private Sub Command1_Click()
Dim wind(100), av_fast(1000), fast_wind(1000), annual_av_wind(1000)
Dim annual_nu1(1000), annual_nu2(1000), annual_nu4(1000), annual_nu6(1000)
Dim annual_nu10(1000), annual_nu8(1000), month_precip_av(1000), month_evapo_av(1000)
Dim month_evapo_av1 As Double
Dim daily_temp(1000), daily_precip(1000), daily_evapo(1000)
Dim stn As String
Dim pr_stn As String

dir_name = "\\Pooh\Metdata\"
Open "d:\samson\Stn_to_do.txt" For Input As #10

Do While Not EOF(10)

    Input #10, stn, pr_stn

    Open dir_name & "Metfiles\" & stn & "y.OUT" For Input As #1
    Open dir_name & stn & "\" & stn & "tp.dat" For Input As #2
    Open dir_name & stn & "\" & stn & "i.dat" For Output As #3

    'yer = 61
    i = 61
    Do While Not EOF(2)

      'Input #1, yer

      'Do While yer = i
        For z = 1 To 24

          Input #2, yer, mth, dy, hr, wind(z), x, y, zzz

          If wind(z) > fastest_wind Then
            fastest_wind = wind(z)
          End If

        Next z

        'co = co + 1
        Print #3, yer, mth, dy, fastest_wind
        fastest_wind = 0
      'Loop
     'If yer = 90 Then Exit Do
     'Print i, co
     'm = yer - i
     'i = i + m

    'co = 0
    Loop

    Close #3
```

```
yer = 61
Open dir_name & stn & "\" & stn & "i.dat" For Input As #5
Open dir_name & stn & "\" & stn & "w.dat" For Output As #99
c = 1
i = 61

Do While Not EOF(1)
   Input #1, yr, pp
      For c = 1 To pp

         Input #5, yer, mth, dy, fastest_wind
         fast_wind(c) = fast_wind(c - 1) + fastest_wind
      Next c

      fast_av = fast_wind(pp) / pp

      Print #99, yr, fast_av
Loop

Close #5
Close #99
Close #1

Open dir_name & stn & "\" & stn & "d.dat" For Input As #5
Open dir_name & stn & "\" & stn & "m.dat" For Input As #55

Open dir_name & stn & "\" & stn & "w.dat" For Input As #99
Open dir_name & "Metfiles\" & stn & "y.OUT" For Input As #1
Open dir_name & stn & "\" & stn & "a.dat" For Output As #23

pr_days = 0
Do While Not EOF(1)

   Input #1, yr, pp

   For c = 1 To pp
      Input #5, Counter, daily_precip1, daily_temp_av1, daily_wind_speed_av1, ccover,
daily_evapo1, daily_R1, u1_final1, nu1, u2_final1, nu2, u4_final1, nu4, u6_final1, nu6, u8_final1,
nu8, u10_final1, nu10

      annual_av_wind(c) = annual_av_wind(c - 1) + daily_wind_speed_av1
      annual_nu1(c) = annual_nu1(c - 1) + nu1
      annual_nu2(c) = annual_nu2(c - 1) + nu2
      annual_nu4(c) = annual_nu4(c - 1) + nu4
      annual_nu6(c) = annual_nu6(c - 1) + nu6
      annual_nu8(c) = annual_nu8(c - 1) + nu8
      annual_nu10(c) = annual_nu10(c - 1) + nu10
      daily_precipx = daily_precip1 * 0.3937
      daily_precip(c) = daily_precip1 + daily_precip(c - 1)
      daily_evapo(c) = daily_evapo(c - 1) + daily_evapo1
      daily_temp(c) = daily_temp(c - 1) + daily_temp_av1
      If daily_precipx > 0.01 Then
         pr_days = pr_days + 1
      End If
```

```
        Next c



        For jj = 1 To 12

            Input #55, zz, month_temp_av1, month_wind_av1, month_precip_av1, month_evapo_av1,
month_cc_av

                month_precip_av(jj) = month_precip_av(jj - 1) + month_precip_av1
                month_evapo_av(jj) = month_evapo_av(jj - 1) + month_evapo_av1
        Next jj

        PE1 = 10 * (month_precip_av(12) / month_evapo_av(12))
        annualnu1 = Int(100 * (annual_nu1(pp) / (365 * 24)))
        annualnu2 = Int(100 * (annual_nu2(pp) / (365 * 24)))
        annualnu4 = Int(100 * (annual_nu4(pp) / (365 * 24)))
        annualnu6 = Int(100 * (annual_nu6(pp) / (365 * 24)))
        annualnu8 = Int(100 * (annual_nu8(pp) / (365 * 24)))
        annualnu10 = Int(100 * (annual_nu10(pp) / (365 * 24)))
        annual_wind = annual_av_wind(pp) / pp
        Input #99, years, max_wind
        annual_wind = Format(annual_wind, "#.00#")
        max_wind = Format(max_wind, "#.0##")
        PE1 = Format(PE1, "0.000#")
        annual_temp1 = daily_temp(pp) / pp
        annual_precip2 = daily_precip(pp)
        annual_evapo3 = daily_evapo(pp) * 100
        annual_temp = Format(annual_temp1, "#.00#")
        annual_precip = Format(annual_precip2, "#.0000#")
        annual_evapo = Format(annual_evapo3, "#.0000#")
        Print #23, yr & ","; annual_wind & ","; annualnu1 & ","; annualnu2 & ","; annualnu4 & ",";
annualnu6 & ","; annualnu8 & ","; annualnu10 & ","; max_wind & ","; pr_days & ","; PE1 & "," &
annual_precip & "," & annual_evapo & "," & annual_temp
        pr_days = 0
    Loop
    Close #5
    Close #55
    Close #2
    Close #99
    Close #1
    Close #23
Loop

Close #10
End
End Sub
```

Long.vbp (Longterm.frm)

This program reads several files to generate long term file.

Private Sub Command1_Click()

```
Dim temp(262968), tot_pre(262968), temp0(100), wind(262968), cco(100000)
Dim daily_precip(100), precip1(100), evapo(262968), pr_days(1000), fastest_wi(100)
Dim anemom, Rusle As Single
```

```
Dim stn As String
Dim pr_stn As String
Open "d:\Samson\Stn_to_do.txt" For Input As #10
dir_name = "\\Pooh\MetData\"
Do While Not EOF(10)

    Input #10, stn, pr_stn

    Open "d:\samson\anemom.dat" For Input As #20
    Open "d:\samson\AnnRusle.prn" For Input As #40
    Open dir_name & stn & "\" & stn & "tp.dat" For Input As #60
    j = 1
    Do While Not EOF(60)

        Input #60, yr1, mth1, dy1, hr1, wind_spee, tem, stab_cat, preci

        temp(j) = temp(j - 1) + tem
        wind(j) = wind(j - 1) + wind_spee
        j = j + 1
       ' If j = 999 Or j = 9999 Or j = 99999 Or j = 200000 Then MsgBox "hello"
    Loop

      yy1 = Int((j - 1) / (24 * 365))



    temera = temp(j - 1) / (j - 1)
    wind_a = wind(j - 1) / (j - 1)

    temera = Format(temera, "0.0#")
    wind_a = Format(wind_a, "0.00#")

    Do While Not EOF(40)
          Input #40, met_stn, Rusle_file

          If met_stn = stn Then
            Rusle = Rusle_file
            Exit Do
          End If

    Loop
    i = 1
    Open dir_name & stn & "\" & stn & "d.dat" For Input As #77

    Do While Not EOF(77)
      Input #77, jday, dailyppt, tmp, ud, ccover, Ed, daiR, u1, Nu1, u2, Nu2, u4, Nu4, u6, Nu6, u8,
Nu8, u10, Nu10
      cco(i) = cco(i - 1) + ccover
      i = i + 1
    Loop

    cc_av = Int(cco(i - 1) / (i - 1))
    Close #77
```

```
        'Rusle = CInt(Rusle)
        Open dir_name & stn & "\" & stn & "a.dat" For Input As #1
        i = 1
        Do While Not EOF(1)
            Input #1, yr, annual_wind, annualnu1, annualnu2, annualnu4, annualnu6, annualnu8,
annualnu10, max_wind, pr_day, PE1, an_ppt, an_evp, an_temp
            pr_days(i) = pr_days(i - 1) + pr_day

            fastest_wi(i) = max_wind + fastest_wi(i - 1)

            i = i + 1
        Loop

        prd = Int(pr_days(i - 1) / (i - 1))
        fastest_win = fastest_wi(i - 1) / (i - 1)
        Open dir_name & stn & "\" & stn & "d.dat" For Input As #44

        bb = 1
        Do While Not EOF(44)
            Input #44, jda, pr, tm, w, ccd, evaporation, ero, u1, Nu1, u2, Nu2, u4, Nu4, u6, Nu6, u8, Nu8,
u10, Nu10
            evapo(bb) = evapo(bb - 1) + evaporation
            tot_pre(bb) = tot_pre(bb - 1) + pr
            bb = bb + 1
        Loop

        evapo_a = (evapo(bb - 1) / yy1) * 100
        prec = tot_pre(bb - 1) / yy1
        evapo_a = Format(evapo_a, "#0.000")
        prec1 = Format(prec, "#0.0#")

        Open dir_name & stn & "\" & stn & "l.dat" For Output As #25
        Print #25, fastest_win & ","; wind_a & ","; prd & ","; prec1 & ","; evapo_a & ","; Rusle & ",";
temera & ","; temera & ","; cc_av

        Close #20
        Close #40
        Close #60
        Close #77
        Close #1
        Close #44
        Close #25

Loop
End
End Sub
```

Appendix 4F. Meteorological Station Assignments for 201 HWIR Industrial D Sites

SiteID	City	State	Met Station	City and State
0114001	PITTSBURGH	PA	94823	Pittsburgh,PA
0130207	MUSCATINE	IA	14923	Moline,IL
0131104	JANESVILLE	WI	94822	Rockford,IL
0131207	NIAGARA FALLS	NY	14733	Buffalo,NY
0131508	KINGSPORT	TN	13877	Bristol,TN
0136703	TUCUMCARI	NM	23047	Amarillo,TX
0220102	CHARLESTON	SC	13880	Charleston,SC
0221207	SIDNEY	OH	93815	Dayton,OH
0223504	ORRVILLE	OH	14895	Akron/Canton,OH
0224002	LA CROSSE	WI	14923	Moline,IL
0231002	WINCHESTER	IN	93815	Dayton,OH
0231106	FLORENCE	NJ	13739	Philadelphia,PA
0231407	MIDLOTHIAN	TX	03927	Fort Worth,TX
0231610	CHICAGO	IL	94846	Chicago,IL
0231911	SAGINAW	MI	14826	Flint,MI
0231914	DETROIT	MI	94847	Detroit,MI
0232305	RANDOLPH	VT	14742	Burlington,VT
0232313	TYLER	TX	13957	Shreveport,LA
0232402	COATESVILLE	PA	13781	Wilmington,DE
0232415	CLEVELAND	OH	14820	Cleveland,OH
0232501	SHARPSVILLE	PA	14852	Youngstown,OH

(continued)

Appendix 4F. (continued)

SiteID	City	State	Met Station	City and State
0232705	CARTERSVILLE	GA	13874	Atlanta,GA
0233601	WAYNESBORO	VA	13741	Roanoke,VA
0233603	LEEDS	AL	03856	Huntsville,AL
0234904	E STROUDSBURG	PA	14737	Allentown,PA
0235301	NEW CASTLE	IN	93819	Indianapolis,IN
0312301	DAYTON	VA	13741	Roanoke,VA
0314202	TROY	NY	14735	Albany,NY
0321802	PORTLAND	OR	24229	Portland,OR
0331006	WEST HELENA	AR	13893	Memphis,TN
0331902	WOODWARD	OK	13985	Dodge City,KS
0332104	BORGER	TX	23047	Amarillo,TX
0332707	MULBERRY	FL	12842	Tampa,FL
0332811	MARION	OH	14821	Columbus,OH
0430108	BROWNVILLE	NE	03947	Kansas City,MO
0430412	PERRY	OH	14820	Cleveland,OH
0431912	LA CYGNE	KS	03947	Kansas City,MO
0432011	ATHENS	AL	03856	Huntsville,AL
0432106	MARTINS CREEK	PA	14737	Allentown,PA
0432716	COHASSET	MN	14913	Duluth,MN
0433201	WHEATFIELD	IN	14848	South Bend,IN
0433204	GEORGETOWN	SC	13880	Charleston,SC
0433404	WHEATLAND	WY	24018	Cheyenne,WY
0433408	COLSTRIP	MT	24037	Miles City,MT
0434505	IMPERIAL	CA	23183	Phoenix,AZ
0434804	GOLDSBORO	NC	13722	Raleigh-Durham,NC
0435510	HAVANA	IL	14842	Peoria,IL
0436007	MONCURE	NC	13722	Raleigh-Durham,NC

(continued)

Appendix 4F. (continued)

SiteID	City	State	Met Station	City and State
0436108	BELMONT	NC	13881	Charlotte,NC
0530901	HURON	OH	14820	Cleveland,OH
0531301	ILLIOPOLIS	IL	93822	Springfield,IL
0531502	JACKSON	OH	03860	Huntington,WV
0531702	BORGER	TX	23047	Amarillo,TX
0531902	PASADENA	TX	12960	Houston,TX
0534504	TUCKER	GA	13874	Atlanta,GA
0613402	EAST ST LOUIS	IL	13994	St. Louis,MO
0620401	HOUSTON	TX	12960	Houston,TX
0620604	SALT LAKE CY	UT	24127	Salt Lake City,UT
0621603	LA PORTE	TX	12960	Houston,TX
0621902	HUTCHINSON	KS	03928	Wichita,KS
0622902	SAHUARITA	AZ	23160	Tucson,AZ
0625002	SAVANNAH	GA	03822	Savannah,GA
0625501	DAYTON	TX	12960	Houston,TX
0631701	LA PORTE	TX	12960	Houston,TX
0631903	LAKE CHARLES	LA	03937	Lake Charles,LA
0632003	LONGVIEW	TX	13957	Shreveport,LA
0632606	PAMPA	TX	23047	Amarillo,TX
0632608	PORTLAND	OR	24229	Portland,OR
0634001	CARTERSVILLE	GA	13874	Atlanta,GA
0635301	HENDERSON	NV	23169	Las Vegas,NV
0713618	ROSEBURG	OR	24225	Medford,OR
0713705	CASA GRANDE	AZ	23183	Phoenix,AZ
0715007	NORRISTOWN	PA	13739	Philadelphia,PA
0715216	YORBA LINDA	CA	23174	Los Angeles,CA
0716701	BOONVILLE	IN	93817	Evansville,IN

(continued)

Appendix 4F. (continued)

SiteID	City	State	Met Station	City and State
0720506	COLUMBIA FLS	ME	14764	Portland,ME
0720803	VENICE	FL	12842	Tampa,FL
0721305	DETROIT	MI	94847	Detroit,MI
0722107	PUNTA GORDA	FL	12842	Tampa,FL
0722503	DENVER	CO	94018	Boulder,CO
0722505	HOLLYWOOD	FL	12839	Miami,FL
0722705	BINGHAMTON	NY	04725	Binghamton,NY
0723607	LLANO	TX	13958	Austin,TX
0724206	WOODLEAF	NC	13723	Greensboro,NC
0724301	HOLLY SPRINGS	MS	13893	Memphis,TN
0724804	TELFORD	PA	14737	Allentown,PA
0724909	WILLISTON	ND	24037	Miles City,MT
0730407	BUCKHANNON	WV	94823	Pittsburgh,PA
0730502	ROBINSON	IL	93817	Evansville,IN
0730914	ELGIN	TX	13958	Austin,TX
0731111	CORNING	NY	04725	Binghamton,NY
0731405	MARIETTA	PA	13781	Wilmington,DE
0731411	CHOWCHILLA	CA	93193	Fresno,CA
0731412	LITTLETON	MA	14739	Boston,MA
0731501	CLIFFWOOD	NJ	14734	Newark,NJ
0731507	LAURINBURG	NC	13722	Raleigh-Durham,NC
0731514	WAUKEGAN	IL	94846	Chicago,IL
0731703	ROYERSFORD	PA	13739	Philadelphia,PA
0732110	DURYEA	PA	14777	Wilkes-Barre/S,PA
0732405	MODESTO	CA	23232	Sacramento,CA
0732510	PLAINFIELD	IL	94846	Chicago,IL
0733203	TRACY	CA	23232	Sacramento,CA

(continued)

Appendix 4F. (continued)

SiteID	City	State	Met Station	City and State
0733210	SUMMITVILLE	OH	14895	Akron/Canton,OH
0733302	DANVILLE	KY	93820	Lexington,KY
0733404	ST MARYS	PA	14778	Williamsport,PA
0733501	MOUNTAIN HOME	NC	03812	Asheville,NC
0733606	PLYMOUTH MTG	PA	13739	Philadelphia,PA
0734604	BERLIN	NJ	13739	Philadelphia,PA
0735309	INDEPENDENCE	KS	13968	Tulsa,OK
0826707	WINSTON-SALEM	NC	13723	Greensboro,NC
0830601	COOSA PINES	AL	03856	Huntsville,AL
0830903	NEWTON FALLS	NY	14742	Burlington,VT
0831102	WINSLOW	ME	14764	Portland,ME
0831406	DEMOPOLIS	AL	13865	Meridian,MS
0831904	VALDOSTA	GA	03813	Macon,GA
0832304	GREEN BAY	WI	14898	Green Bay,WI
0832510	RANSOM	PA	14777	Wilkes-Barre/S,PA
0832903	ST FRANCISVL	LA	13970	Baton Rouge,LA
0832904	WINDER	GA	13873	Athens,GA
0832909	MIDDLETOWN	OH	93815	Dayton,OH
0833001	CANTON	NC	03812	Asheville,NC
0833007	PESHTIGO	WI	14898	Green Bay,WI
0834009	MIQUON	PA	13739	Philadelphia,PA
0923004	GRAND RAPIDS	MI	94860	Grnd Rapids,MI
0930205	BADIN	NC	13723	Greensboro,NC
0930301	GOLDENDALE	WA	24243	Yakima,WA
0930702	ROCKDALE	TX	13958	Austin,TX
0932103	GASTON	SC	13883	Columbia,SC
0932507	CRESSONA	PA	14737	Allentown,PA

(continued)

Appendix 4F. (continued)

SiteID	City	State	Met Station	City and State
0932509	ADDY	WA	24157	Spokane,WA
0932903	POINT COMFORT	TX	12960	Houston,TX
0933704	ABINGDON	VA	13877	Bristol,TN
1010805	WALLA WALLA	WA	24155	Pendleton,OR
1012203	AQUEBOGUE	NY	94728	New York,NY
1013209	WOODLAKE	CA	93193	Fresno,CA
1014805	BAINBRIDGE	GA	93805	Tallahassee,FL
1015510	HASKELL	TX	23042	Lubbock,TX
1023705	GENEVA	NY	14768	Rochester,NY
1031503	SALINA	KS	03928	Wichita,KS
1031507	MADERA	CA	93193	Fresno,CA
1032715	GADSDEN	AL	03856	Huntsville,AL
1032802	PITTSBORO	NC	13722	Raleigh-Durham,NC
1033107	TRAVERSE CITY	MI	14840	Muskegon,MI
1033114	HARBESON	DE	93730	Atlantic City,NJ
1033202	VAN BUREN	AR	13968	Tulsa,OK
1033602	PLAINWELL	MI	94860	Grnd Rapids,MI
1034005	FLOWERY BR	GA	13873	Athens,GA
1034210	HAMMONDSPORT	NY	04725	Binghamton,NY
1034406	GREENVILLE	MS	03940	Jackson,MS
1034805	BATESVILLE	AR	13995	Springfield,MO
1035117	LIBERTY	MO	03947	Kansas City,MO
1035405	NAMPA	ID	24131	Boise,ID
1035508	RICHMOND	UT	24156	Pocatello,ID
1120904	BURLINGTON	VT	14742	Burlington,VT
1122705	HAYSI	VA	13877	Bristol,TN
1131103	NEW ORLEANS	LA	12916	New Orleans,LA

(continued)

Appendix 4F. (continued)

SiteID	City	State	Met Station	City and State
1131802	PLACERVILLE	CA	23232	Sacramento,CA
1133902	LAURENS	SC	03870	Greenville,SC
1134405	HANNIBAL	MO	03945	Columbia,MO
1212301	STOUGHTON	MA	14739	Boston,MA
1221704	LARGO	FL	12842	Tampa,FL
1223404	CONSHOHOCKEN	PA	13739	Philadelphia,PA
1230111	LAUREL	MT	24033	Billings,MT
1230206	MEMPHIS	TN	13893	Memphis,TN
1230517	FINKSBURG	MD	93721	Baltimore,MD
1230919	ALBANY	NY	14735	Albany,NY
1231101	MEMPHIS	TN	13893	Memphis,TN
1231705	WANDO	SC	13880	Charleston,SC
1233101	WILMINGTON	CA	23174	Los Angeles,CA
1235205	ADA	OK	13967	Oklahoma City,OK
1236637	WEST NYACK	NY	94728	New York,NY
1236652	MERIDEN	CT	14740	Hartford,CT
1236732	BECKLEY	WV	13741	Roanoke,VA
1236810	SANTA FE SPGS	CA	23174	Los Angeles,CA
1236820	TOWANDA	PA	04725	Binghamton,NY
1331103	GADSDEN	AL	03856	Huntsville,AL
1333001	FLORENCE	SC	13883	Columbia,SC
1333701	WARREN	OH	14852	Youngstown,OH
1415407	PHOENIX	AZ	23183	Phoenix,AZ
1421506	RENTON	WA	24233	Seattle,WA
1430107	WICHITA	KS	03928	Wichita,KS
1430404	INDIANAPOLIS	IN	93819	Indianapolis,IN
1430602	SUNNYVALE	CA	23234	San Francisco,CA

(continued)

Appendix 4F. (continued)

SiteID	City	State	Met Station	City and State
1431515	MAGNA	UT	24127	Salt Lake City,UT
1434022	TORRANCE	CA	23174	Los Angeles,CA
1434802	COLUMBIA	SC	13883	Columbia,SC
1435317	BEAUMONT	TX	12917	Port Arthur,TX
1522504	BEAR	DE	13781	Wilmington,DE
1530605	BEAUMONT	TX	12917	Port Arthur,TX
1530808	TRENTON	MI	94847	Detroit,MI
1532401	MIDDLEBURY	VT	14742	Burlington,VT
1621808	SENOIA	GA	13874	Atlanta,GA
1630106	RABUN GAP	GA	03812	Asheville,NC
1630401	KNOXVILLE	TN	13891	Knoxville,TN
1631701	WASHINGTON	NC	13722	Raleigh-Durham,NC
1632106	MESILLA PARK	NM	23044	El Paso,TX
1632703	WHITMIRE	SC	03870	Greenville,SC
1633404	LITCHFIELD PK	AZ	23183	Phoenix,AZ
1633405	NEW MILFORD	CT	14740	Hartford,CT
1635404	LIBERTY	SC	03812	Asheville,NC
1721603	SHERWOOD	OR	24229	Portland,OR

* Addresses for the HWIR sites are from the Industrial D Screening Survey