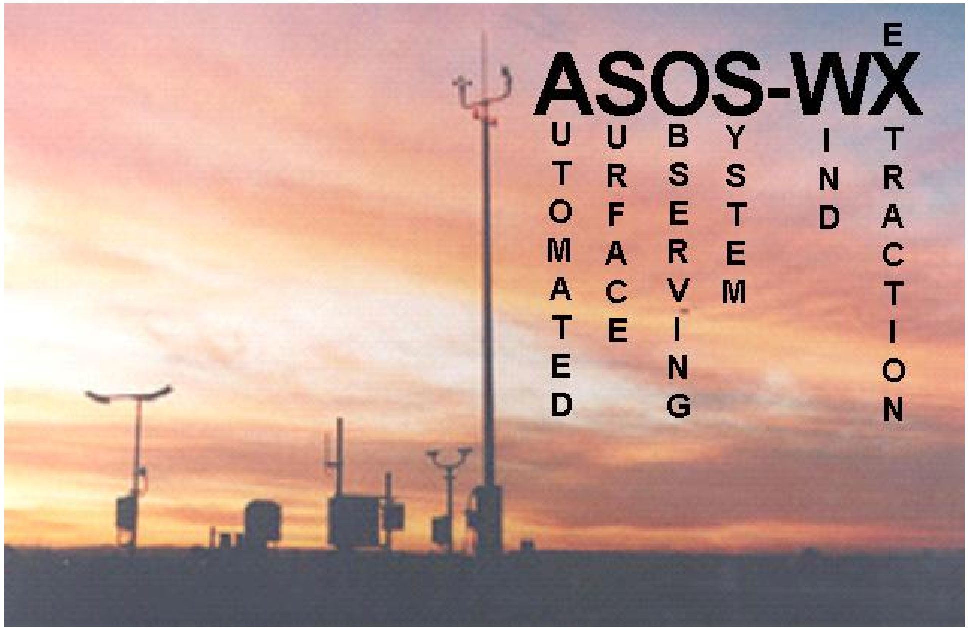
Documentation for **ASOS-WX** Software



developed by Franklin T. Lombardo and Joseph A. Main

Building and Fire Research Laboratory
National Institute of Standards and Technology

Updated February 20, 2008 (*Original version: January 9, 2006*)
Current version available at www.nist.gov/wind

Click the following hyperlinks to jump to the corresponding section of the documentation:

Overview
MATLAB m-files
History of program updates

Overview

ASOS-WX is a MATLAB-based[1,2] software tool for extraction of wind and thunderstorm data from archived ASOS (Automated Surface Observing System) weather reports for use in extreme value analyses. More information about ASOS itself can be found in the *ASOS User's Guide* from NOAA. **ASOS-WX** is designed to extract data from records in Data Set 9956 from the National Climatic Data Center (NCDC). ASCII files containing records for specific stations over specified date ranges can be purchased from NCDC as described in the *Documentation for Data Set 9956* from NCDC. Within **ASOS-WX**, one or more ASOS files can be selected for analysis, and the software can then perform the following actions for all selected files:

1. Extraction of peak gust wind data (speed, direction, and date/time)
2. Extraction of thunderstorm reports (beginning and end times, manual thunderstorm observations)
3. Classification of wind data as “Thunderstorm” or “Non-Thunderstorm” (to enable separate statistical analysis)
4. Construction of data sets separated by specified minimum time intervals (to ensure statistical independence)

The selected output results are written as comma-delimited text files (*.csv), which can be readily opened in standard spreadsheet software for further analysis and plotting.

The **ASOS-WX** software is available for download at www.nist.gov/wind, by following the link for “Extraction of wind data from ASOS records”. At this **ASOS-WX** website, the software is available either as MATLAB m-files, for execution within MATLAB, or as a stand-alone executable. The **ASOS-WX** software was developed and tested in the Microsoft Windows environment using version 7 of MATLAB. It is believed that the MATLAB m-files can be ported to non-Windows (e.g., Linux/Unix) environments with minimal changes. For execution within MATLAB, the folder containing the downloaded **ASOS-WX** m-files must be added to the MATLAB search path, and the software can then be launched by typing “asos_wx” at the MATLAB command prompt.

A “stand-alone” version of the software is also available at the **ASOS-WX** website, for users who do not have MATLAB version 7 and are running in the Windows 2000/XP environment. This stand-alone version requires installation of the MATLAB Component Runtime (MCR) Libraries provided by The MathWorks, Inc. The MCR Libraries are available for download[3] at the **ASOS-WX** website as a self-installing executable named “MCRInstaller.exe”. The stand-alone executable files are available for download at the **ASOS-WX** website as a single self-extracting zip file named “stand-alone.exe”. Execution of this file will create a folder named “stand-alone”, and the **ASOS-WX** software can then be launched by executing the application “ASOS_WX.exe” within this folder.

To enable users to evaluate this software, a sample ASOS record (HAS0000826915.op) is provided for download at the **ASOS-WX** website, containing data from two different airports in the New York City area:

- Station 725020: Newark International Airport (13 months of data)
- Station 725030: New York LaGuardia Airport (39 months of data)

Also provided for download at the **ASOS-WX** website is a file named WBAN-MSC.TXT, which was obtained from the Surface Inventories Page on the NOAA website. This file contains station identification numbers and descriptive names for weather stations around the world, including ASOS stations in the United States. This file is used by the **ASOS-WX** software for identification of stations encountered in ASOS files.

[1] MATLAB®. © 1984 – 2005 The Mathworks, Inc.

[2] Certain trade names or company products are mentioned in the text to specify adequately the procedure used. Such identification does not imply recommendation or endorsement by NIST, nor does it imply that the product is the best available for the purpose.

[3] The authors' limited rights to the deployment of this program are limited by a license agreement between NIST and The MathWorks. The license agreement can be found at www.mathworks.com/license/. The authors, NIST, and The MathWorks and its licensors are excluded from all liability for damages or any obligation to provide remedial actions.

MATLAB M-files

The **ASOS-WX** software consists of the following nine MATLAB functions, which are further described subsequently (the functions are listed in the order they are called when **ASOS-WX** is executed):

- asos_wx: The main function, which calls the other functions
- PK_WND: Extracts peak gust wind data from ASOS files
- case_time: Determines the date and time of a peak wind or thunderstorm beginning/ending report using relevant information from the current line of the ASOS file
- ASOS_id: Extracts ASOS station numbers and descriptive names from WBAN-MSC.TXT, a text file provided by NOAA
- MW: Extracts manual thunderstorm observations from ASOS files
- tsBE: Extracts thunderstorm beginning and end times from ASOS files
- matchBE: Matches thunderstorm beginning and end times from MW and tsBE functions
- set_ts_windows: Opens a dialog box in which users can specify lengths of time for extending thunderstorm windows, for classification of “Thunderstorm” and “Non-Thunderstorm” winds
- set_sep_intervals: Opens a dialog box in which users can specify time intervals by which “Thunderstorm” and “Non-Thunderstorm” winds must be separated to ensure statistical independence
- interval_sep: Constructs data sets separated by at least the specified time interval, to ensure statistical independence

Each function is available for download as a MATLAB m-file with the same name as the function and a “.m” file extension.

asos_wx

This is the main function of the **ASOS-WX** software, and it calls a number of other functions that are described subsequently. The flow of the analysis within the *asos_wx* function is as follows:

1. A dialog box is opened that prompts the user to select one or more ASOS files for analysis (to select multiple files, hold down the "Ctrl" key while selecting files).
2. The PK_WND function is called to extract peak wind speed observations from each of the selected ASOS files, along with their corresponding wind direction, date/time, and ASOS station code. The PK_WND function calls the case_time function to determine the date/time of each peak wind observation using relevant information on the current line.
3. A dialog box is opened that prompts the user to select one ASOS station from a list of stations (with station numbers, descriptive station names, and number of peak wind reports found) for which peak wind observations were identified in the selected files. Descriptive names corresponding to the reported ASOS station numbers are obtained using the ASOS_id function.
4. The user is prompted to select which of the following output files should be written:

(A) All winds	(filename: "station_######_ALL.csv")
(B) Non-thunderstorm winds	(filename: "station_######_NTS.csv")
(C) Thunderstorm winds	(filename: "station_######_TS.csv")
(D) Non-thunderstorm winds, separated by specified time interval	(filename: "station_######_NTS_sep.csv")
(E) Thunderstorm winds, separated by specified time interval	(filename: "station_######_TS_sep.csv")
(F) Manual thunderstorm observations	(filename: "station_######_MW.csv")
(G) Thunderstorm beginning and end times	(filename: "station_######_tsBE.csv")

5. A dialog box is opened that prompts the user to select a location for writing the specified output files.
6. If any of output files (B) through (G) are selected, then the MW function is called to extract manual thunderstorm observations for the specified ASOS station from the selected files. The date/time of each manual thunderstorm observation is returned along with a code that indicates the type of thunderstorm in progress (descriptions for the thunderstorm codes are given in a table in the description of the MW function). The MW function also returns date/times of thunderstorm beginnings and ends estimated from the manual thunderstorm observations.
7. If any of output files (B) through (E) or (G) are selected, then the tsBE function is called to extract the date/times of reported thunderstorm beginnings and ends for the specified ASOS station from the selected files. The tsBE function calls the case_time function to determine the date/time of each reported thunderstorm beginning and end using relevant information on the current line. These reported beginning and ending times are more precise than those estimated by the MW function in step 6; however, the beginning and ending times extracted from the tsBE function do not always match (e.g., two consecutive beginning times are sometimes encountered without an intervening ending time). Therefore, the matchBE function is called to create lists of matching beginning and ending times using the estimated beginning and ending times from the MW function in conjunction with those from the tsBE function.
8. If any of output files (B) through (E) are selected, then the following steps are performed:
   a. The function set_ts_windows is called, which opens a dialog box prompting the user to specify lengths of time to extend thunderstorm windows before and after the matching beginning and ending times, for identification of thunderstorm winds.
   b. The times of the peak wind observations are used in conjunction with the reported thunderstorm beginning and end times to classify the peak winds as "Thunderstorm" or "Non-Thunderstorm". Peak winds are classified as "Thunderstorm" if they fall in the interval between a matching thunderstorm beginning and ending or within the specified lengths of time before a thunderstorm beginning or after a thunderstorm ending. Peak winds that do not fall within these windows are classified as "Non-Thunderstorm."
9. If output file (D) or (E) are selected, then the following steps are performed:

a. The function set_sep_intervals is called, which opens a dialog box and prompts the user to specify minimum time intervals by which “Thunderstorm” and “Non-Thunderstorm” winds should be separated in order to ensure statistical independence. Different separation intervals can be specified for “Thunderstorm” and “Non-Thunderstorm” winds.
b. The function interval_sep is called separately for “Thunderstorm” and/or “Non-Thunderstorm” winds to obtain peak wind speed, wind direction, and date/time values separated by at least the time intervals specified in the previous step.

10. The specified output files are written in comma-delimited format (*.csv), and a dialog box is opened to inform the user of the names of the files that have been written. Output files (A) through (E) each contain three columns of data, giving the date/time, wind speed, and wind direction for the selected type of winds. Output file (F) contains data from step 6 above (output from the MW function) arranged into two different sections. The first section contains two columns of data giving the date/time and code of each manual thunderstorm observation, and the second section contains three columns of data, giving an estimated beginning date/time, estimated end date/time, and estimated duration for each thunderstorm that was reported by one or more consecutive manual thunderstorm observations. Output file (G) contains data from step 7 above, arranged in two different sections. The first section presents the output of the matchBE function and contains three columns, giving the beginning date/time, ending date/time, and computed duration of each thunderstorm. The second section presents the output of the tsBE function and contains two columns, one giving the list of reported beginning date/times and the other a list of reported ending date/times. These two columns generally have different lengths because of the coding errors noted above.

PK_WND

Syntax: `[ws, wd, date_time, station_id] = PK_WND(full_filename)`

Description:

This function extracts peak gust wind data from the selected ASOS file. The standard format for reporting peak wind gusts in ASOS files is as follows: “PK WND **dddss(s)/(hh)mm**”, where **ddd** represents three numeric characters giving the wind direction in degrees, **ss(s)** represents either two or three numeric characters giving the wind speed in knots (three characters are used only if the wind speed exceeds 99 knot), and **(hh)mm** represents either two or four numeric characters giving the time of the report. Prior to the stations becoming fully automated, peak winds were reported as “PK WND **D(D)ss(s)/(hh)mm**”, where “**D(D)**” represents either one or two numeric characters giving the wind direction in tens of degrees; e.g., 270° would be indicated as “27” while 90° would be indicated as “9”. The existence of these various possible coding formats for wind speed and direction presents a challenge in the interpretation of peak wind reports. The total number of digits preceding the slash can range from three to six, and the following table summarizes the rules that are applied in interpreting the wind speed and direction depending on the number of digits encountered:

Number of Digits	Interpretation
<3	*ambiguous* → *rejected*
3	**Dss**
4	if 1st digit >3 or 3rd digit ≤ 1 (see note 1) → **Dsss** otherwise (see note 2) → **DDss**

5	if 3rd digit = 0 (see note 3) → **dddss**
	if 3rd digit = 1 → **DDsss**
	otherwise (see note 4) → *ambiguous* → *rejected*

6	**dddsss**
>6	*ambiguous* → *rejected*

Notes:

1. The wind direction cannot be greater than 360°, and wind speeds of less than 25 knot are not reported.
2. The case of 4 digits is actually ambiguous if the 1st digit ≤ 3, the 3rd digit > 1, and the 2nd digit ≤ 1; e.g., “3145” could be interpreted as a speed of 45 knot at 310° (**DDss**) or as a speed of 145 knot at 30° (**Dsss**). However, the latter interpretation is much less plausible, and such an interpretation is only possible for a wind speed exceeding 119 knot with a direction between 0° and 30°. Such an occurrence is considered sufficiently improbably that the **DDss** interpretation is adopted for all 4-digit cases in which the 1st digit ≤ 3 and the 3rd digit > 1.
3. Wind directions are reported in increments of 10°.
4. Wind speeds in excess of 199 knot are not considered a realistic possibility and are rejected.

Two digits are generally used to represent the time (**mm**) if the peak gust occurred within the hour previous to the time reported at the start of the data line, while four characters are used (**hhmm**) if the peak gust occurred more than an hour previously. The case_time function is called to determine the date and time of each peak gust from the two- or four-digit time code and the date and time reported near the start of the data line.

Variants of the “PK WND” coding (e.g., “PKWND” or “PK WNDS”) are sometimes encountered in ASOS data files. Since the “PK” code is not used for reporting any other information in ASOS files, the PK_WND function searches for only the string “PK”, followed within 30 characters on the same line by a slash ("/"). The consecutive numerical characters preceding the slash are taken to represent the wind speed and direction, interpreted as indicated in the table above, and the numerical characters following slash are taken to represent the time.

A fairly commonly encountered error in the reporting of peak gust winds in ASOS files is the repetition of a "PK WND" report on subsequent lines without switching from the **mm** time format to the **hhmm** format. Because the time at the start of the line is used to determine the hour of each report, such repetitions would be misinterpreted as a new event occurring at a later hour. To avoid such errors, the PK_WND function ignores reports in which the same wind speed and time code are repeated within an interval of two hours, unless a different wind speed is reported between the two. Another error that is sometimes encountered is a time code having other than two or four digits. The time cannot be interpreted unambiguously in such cases, so the time reported at the start of the line is used for the peak wind. However, if the ambiguous time code is reported within two hours of a previous "PK WND" report having the same wind speed, then the ambiguous report is assumed to be a repetition and is ignored.

Input Arguments:

`full_filename` A string containing the entire pathname and filename of the ASOS record to be analyzed (e.g., "C:\ASOSfiles\HAS0000826915.op").

Output Arguments:

`ws` A vector of peak wind speed values reported in the selected file.

`wd` A corresponding vector of wind direction values reported in the selected file.

`date_time` A corresponding vector of serial date numbers (returned by the MATLAB *datenum* function) giving the date and time of each peak wind report.

`station_id` A corresponding vector of numerical values (six-digit codes used by NOAA) that defines the ASOS station for each peak wind report. This output argument enables handling of ASOS files containing data from more than one station.

case_time

Syntax: `date_time = case_time(time_str, line_datestr)`

Description:

This function is called by both the PK_WND and tsBE functions, and it determines the date and time of a peak wind or thunderstorm beginning/end observation using relevant information from the current line of the ASOS file. Relevant information includes the date and time reported near the start of the current data line (input argument `line_datestr`) and the time reported for the peak wind or thunderstorm beginning/end observation itself (input argument `time_str`). Most commonly `time_str` contains only two characters, representing the minute of the observation; in some cases `time_str` may contain four characters, representing both the hour and the minute. The date (and in most cases the hour as well) must be inferred from the information contained in `line_datestr`. This inference is made by noting that the time represented by `time_str` must precede the time represented by `line_datestr`, because `line_datestr` represents the time that the current weather report was written. If the number of characters in `time_str` does not equal two or four, then its meaning is ambiguous. In such cases the date and time in `line_datestr` are returned in the output argument `date_time`. The following table illustrates several different scenarios in which the output argument `date_time` is determined from different values of the input arguments:

line_datestr	**time_str**	**date_time** (converted to ‘datestr’ format)
199911221051	37	22-Nov-1999 10:37:00
199911221051	57	22-Nov-1999 09:57:00 (previous hour)
199911221051	0937	22-Nov-1999 09:37:00
199911220042	53	21-Nov-1999 23:53:00 (previous day)
199911220042	123 (ambiguous)	21-Nov-1999 00:42:00 (only in PK WND)

Input Arguments:

`time_str` A string representing the time of the observation. Either 2 or 4 numeric characters are expected, representing the minute of the report as **mm** or the hour and minute of the report as **hhmm**.

`line_datestr` A string of 12 numeric characters representing the year, month, day, hour, and minute of the current weather report as **YYYYMMDDhhmm**. For example, November 22, 1999 at 04:00 appears as “199911220400”.

Output Arguments:

`date_time` A serial date number (returned by the MATLAB *datenum* function) giving the date and time inferred for the observation

ASOS_id

Syntax: `[station_id, station_dn] = ASOS_id`

Description:

The purpose of this function is to obtain descriptive names corresponding to the station numbers that are reported in ASOS files. These descriptive names are used by the asos_wx function to notify the user of which stations were found in the selected ASOS files. *ASOS_id* obtains a list of ASOS station numbers and corresponding descriptive names by reading the file WBAN-MSC.TXT, which is available for download at the **ASOS-WX** website and can also be obtained through the Surface Inventories Page on the NOAA website. A dialog box prompts the user to select this file. If the file selection is cancelled or if errors are encountered in reading the file, then the user is warned that descriptive station names will be unavailable, and `ASOS_id` returns with empty output arguments. If the file is read successfully, then the output arguments `station_id` and `station_dn` are saved in a MATLAB MAT-file named “ASOS_stations.mat” in the default directory, in addition to being returned by `ASOS_id`. In subsequent analyses, `station_id` and `station_dn` can then be accessed more efficiently by simply loading “ASOS_stations.mat” rather than using this function. Before calling *ASOS_id*, asos_wx checks whether the file “ASOS_stations.mat’” exists and loads this file if it does.

Input Arguments: *None*

Output Arguments:

`station_id` A vector of ASOS station numbers (six-digit codes used by NOAA)

`station_dn` A corresponding cell array of strings containing descriptive names for each ASOS station

MW

Syntax: `[MW_B_date_time, MW_E_date_time, MW_date_time, MW_code] = MW(sel_filename, station_id)`

Description:

This function extracts manual thunderstorm observations for the specified ASOS station from the specified files. Manual weather observations are indicated in ASOS data files by the characters “MW” followed by three numeric characters. The first of these numeric characters is simply a counter to indicate the number of manual weather observations on the current data line (e.g., the first manual weather observation on a given line begins with “MW1”, the second with “MW2”, and so on). The second and third of these numeric characters are a code that represents the type of weather in progress at the time of the current report. As shown in the following table, seven different codes are available for indicating a thunderstorm in progress, and this function searches for all occurrences of any of these codes.

Code	Description
17	Thunderstorm, but no precipitation at time of observation
29	Thunderstorm (with or without precipitation)
95	Thunderstorm, slight or moderate, without hail, but with rain and/or snow at time of observation
96	Thunderstorm, slight or moderate, with hail at time of observation
97	Thunderstorm, heavy, without hail, but with rain and/or snow at time of observation
98	Thunderstorm, combined with duststorm or sandstorm at time of observation
99	Thunderstorm, heavy, with hail at time of observation

The date and time of each report are obtained from the information provided near the start of the data line, and this function returns the date/time of each manual thunderstorm observation (output argument `MW_date_time`) along with the code from the table above used for each observation (output argument `MW_code`).

In addition to extracting the date/time and code of each manual thunderstorm observation, the date and time of thunderstorm beginnings and ends are also estimated from the manual thunderstorm observations and are returned in the output arguments `MW_B_date_time` and `MW_E_date_time`. These estimated beginning and end date/times are used by the matchBE function to resolve ambiguities in the reported beginning and end times extracted by the tsBE function. In estimating thunderstorm beginnings and ends, manual thunderstorm observations occurring in consecutive weather reports are assumed to represent a single thunderstorm. The thunderstorm beginning is then estimated as the date/time associated with the first in a set of consecutive manual thunderstorm observations. The thunderstorm end is estimated as the date/time of the first subsequent weather report that does not contain a manual thunderstorm observation and is 15 min or more after the estimated beginning. A thunderstorm ending within 15 min of the estimated beginning is not accepted because the definition of a thunderstorm ending in the *Federal Meteorological Handbook No. 1* implies a minimum thunderstorm duration of 15 min. Weather reports should occur at least once an hour, but in cases in which a reporting gap of more than two hours is encountered following a manual thunderstorm observation, the thunderstorm is assumed to have ended one hour after the last manual thunderstorm observation. (A reporting gap may occur, for example, if non-consecutive ASOS files are selected for analysis.)

Input Arguments:

`sel_filename` A cell array of strings containing the entire pathname and filename of each ASOS file to be analyzed (e.g., “C:\ASOSfiles\HAS0000826915.op”) `station_id` A numerical value (six-digit code used by NOAA) that defines the ASOS station for which manual thunderstorm observations are to be extracted from the selected file. This input argument enables handling of ASOS files containing data from more than one station.

Output Arguments:

`MW_B_date_time` A vector of serial date numbers (returned by the MATLAB *datenum* function) giving the estimated date and time of thunderstorm beginnings

`MW_E_date_time` A corresponding vector of serial date numbers giving the estimated date and time of thunderstorm ends. The length of `MW_E_date_time` is the same as the length of `MW_B_date_time`. and the duration of the kth thunderstorm (in days) is given by the following MATLAB expression: `MW_E_date_time(k) - MW_B_date_time(k)`

`MW_date_time` A vector of serial date numbers giving the date and time of each manual thunderstorm observation

`MW_code` A corresponding vector of numerical values giving the code (from the table above) for each thunderstorm observation

tsBE

Syntax: `[B_date_time, E_date_time] = tsBE(sel_filename, station_id)`

Description:

This function extracts reported thunderstorm beginning and end date/times for the specified ASOS station from the selected file. Thunderstorm beginning times are reported in ASOS files using the codes “TB” or “TSB” followed by a two-digit or four-digit time code, which represents the minute of the report as **mm** or the hour and minute of the report as **hhmm**. Thunderstorm end times are reported in a similar manner using the codes “TE” of “TSE” followed by a two- or four-digit time code. Four digits are generally used when the time of occurrence was more than an hour before the time of the current report. The date and time of the current report are given near the start of each line in an ASOS file, and the function case_time is called to determine the time of each beginning or end report from the two- or four-digit time code and the date and time reported at the start of the line. Thunderstorm beginning and end times are also sometimes reported in alternation by using a “B” and an “E” to denote subsequent beginning and end times (e.g., “TSB12E57”, “TSE1245B13”, or “TSB2135E11B47”). In some cases a space may appear between the thunderstorm beginning or end code and the numerical characters, and the variants “T B” and “T E” are also sometimes encountered to denote a thunderstorm beginning and end, respectively. This function extracts beginning and end times reported in any of these possible formats.

A fairly commonly encountered error in the reporting of thunderstorm beginning and end times in ASOS files is the repetition of a time code on subsequent lines without switching from the **mm** time format to the **hhmm** format. Because the time at the start of the line is used to determine the hour of each report, such repetitions would be misinterpreted as a new event occurring at a later hour. To avoid such errors, the tsBE function ignores reports in which the same time code is repeated for a beginning (or end) within two hours of a previous beginning (end) report, unless a different beginning (end) time code occurs between the two. Another error that is sometimes encountered is a time code having other than two or four digits (e.g., “TSB123” or “TE7”). Because the time cannot be interpreted unambiguously in such cases, such reports are ignored.

Input Arguments:

`sel_filename` A cell array of strings containing the entire pathname and filename of each ASOS file to be analyzed (e.g., “C:\ASOSfiles\HAS0000826915.op”)

`station_id` A numerical value (six-digit code used by NOAA) that defines the ASOS station for which thunderstorm beginning and ending times are to be extracted from the selected file. This input argument enables handling of ASOS files containing data from more than one station

Output Arguments:

`B_date_time` A vector of serial date numbers (returned by the MATLAB *datenum* function) giving the date and time of each thunderstorm beginning report

`E_date_time` A vector of serial date numbers giving the date and time of each thunderstorm end report. Because of reporting errors, the length of `E_date_time` is generally not the same as the length of `B_date_time`, and the function matchBE is used to obtain lists of matching thunderstorm beginning and ending times.

matchBE

Syntax: `[B, E] = matchBE(B_date_time, E_date_time, MW_B_date_time, MW_E_date_time)`

Description:

The purpose of this function is to assemble lists of matching thunderstorm beginning and ending times using the outputs from the functions MW and tsBE. Intervals of thunderstorm occurrence can be defined more precisely by using reported thunderstorm beginning and ending times from tsBE (input arguments `B_date_time` and `E_date_time`) rather than by using estimated beginning and ending times from MW (input arguments `MW_B_date_time` and `MW_E_date_time`). This is because tsBE can extract the precise hour and minute of each reported thunderstorm beginning and ending, while MW must estimate beginnings and endings from the times of the weather reports. However, as a consequence of coding errors in ASOS weather reports, the vectors `B_date_time` and `E_date_time` from tsBE generally do not have the same number of elements, and for any given index `k`, the ending time `E_date_time(k)` may not be associated with the same thunderstorm as the beginning time `B_date_time(k)`. To address this problem, the function *matchBE* uses the outputs from the functions MW and tsBE in conjunction to assemble vectors of matching thunderstorm beginning and ending times (output arguments `B` and `E`). The procedure for assembly of `B` and `E` involves looping through the elements of the vectors `MW_B_date_time` and `MW_E_date_time` (which have the same number of elements, denoted `M`) and searching for corresponding elements in the vectors `B_date_time` and `E_date_time`. Beginning with the index `k=1`, the procedure is as follows:

1. Search for reported thunderstorm beginning times in `B_date_time` that fall within the smaller of the following intervals: (a) the hour preceding `MW_B_date_time(k)` or (b) the interval between `E(k-1)` and `MW_B_date_time(k)` (for $k>1$). (The time of a reported thunderstorm beginning must precede the time of the weather report in which it was indicated, but a beginning time should not precede the previous ending time.) If one or more reported beginning times are found in this interval, then set `B(k)` equal to the earliest of these reported beginning times. If no reported beginning times are found in this interval, then set `B(k)` equal to `MW_B_date_time(k)`.
2. Search for reported ending times in `E_date_time` that fall in the interval between `B(k)` and `MW_E_date_time(k)`. (The time of a reported thunderstorm ending must precede the time of the weather report in which it was indicated, but an ending time should not precede its corresponding beginning time.) If one or more reported ending times are found in this interval, then set `E(k)` equal to the latest of these reported ending times. If no reported ending times are found in this interval, then set `E(k)` equal to `MW_E_date_time(k)`. If `E(k)-B(k)<D_min` for the resulting value of `E(k)`, where `D_min` is a minimum thunderstorm duration, then set `E(k)` equal to `B(k)+D_min`.
3. If `k=M`, then terminate the procedure; otherwise increment the index `k` and repeat from 1.

The resulting vectors `B` and `E` are considered the best available estimates of the thunderstorm beginning and ending times. Both of the resulting vectors will have the same number of elements, and the indices of the beginning and ending times in `B` and `E` will correspond, so that the duration of the kth thunderstorm (in days) is given by the following MATLAB expression: `E(k) - B(k)`. The minimum thunderstorm duration `D_min` is set to 15 min, because the definition of a thunderstorm ending in the *Federal Meteorological Handbook No. 1* implies a minimum thunderstorm duration of 15 min.

Input Arguments:

`B_date_time` A vector of serial date numbers giving the date and time of each thunderstorm beginning report (returned by tsBE)

`E_date_time` A vector of serial date numbers giving the date and time of each thunderstorm ending report (returned by tsBE)

`MW_B_date_time` A vector of serial date numbers giving the date and time of each thunderstorm beginning estimated from manual weather observations (returned by MW)

`MW_E_date_time` A corresponding vector of serial date numbers giving the date and time of each estimated thunderstorm ending (returned by MW)

Output Arguments:

`B` A vector of serial date numbers giving the date and time of each *matching* thunderstorm beginning

`E` A corresponding vector of serial date numbers giving the date and time of each *matching* thunderstorm ending. The duration of the kth thunderstorm (in days) is given by the following MATLAB expression: `E(k) - B(k)`

set_ts_windows

Syntax: `[wind_neg, wind_pos, ok] = set_ts_windows`

Description:

This function opens a dialog box in which the user can specify the lengths of time to extend thunderstorm windows before and after the recorded beginning and end times. Peak winds reported within these extended thunderstorm windows are classified as "Thunderstorm" winds by the asos_wx function.

Input Arguments: *None*

Output Arguments:

`wind_pos`	Length of time (in days) to extend thunderstorm windows after the recorded thunderstorm end
`wind_neg`	Length of time (in days) to extend thunderstorm windows before a recorded thunderstorm beginning
`ok`	A flag that indicates whether the user pressed "OK" (`ok` = 1) or "Cancel" (`ok` = 0).

set_sep_intervals

Syntax: `[interval_nts, interval_ts, ok] = set_sep_intervals`

Description:

This function opens a dialog box in which the user can specify the minimum time intervals by which non-thunderstorm and thunderstorm wind speeds should be separated to ensure statistical independence. The intervals specified here are used as input arguments for the interval_sep function.

Input Arguments: *None*

Output Arguments:

`interval_nts`	Time interval for separation of non-thunderstorm winds
`interval_ts`	Time interval for separation of thunderstorm winds
`ok`	A flag that indicates whether the user pressed "OK" (`ok` = 1) or "Cancel" (`ok` = 0).

interval_sep

Syntax: [ws_sep, date_time_sep, ind_sep] = interval_sep(ws, date_time, interval)

Description:

This function constructs data sets in which all values are separated by at least the specified time interval, in an effort to ensure statistical independence. Values that are less than or equal to neighboring values are eliminated as needed, in order to achieve the specified minimum separation interval. The function is called separately for "Thunderstorm" and "Non-Thunderstorm" winds by the asos_wx function, and the function set_sep_intervals prompts the user to independently specify values of the separation interval (input argument interval) for these distinct types of winds.

The separation procedure works as follows: The time of the first peak wind speed in the list, date_time(1), is checked against the time of the second peak wind speed, date_time(2). If date_time(2) – date_time(1) • interval, then the interval_sep function moves forward to check date_time(3) – date_time(2), and so on. If date_time(k+1) – date_time(k) < interval, then the lesser of the two wind speeds corresponding to those times is eliminated, and the greater is retained. If ws(k+1) = ws(k), then ws(k+1) is eliminated and ws(k) is retained, because ws(k) has a larger separation interval from subsequent peak winds. When a wind speed value is eliminated, the corresponding date_time value is eliminated as well, and the *interval_sep* function then compares the surviving date_time value with the next value in the list. This procedure is continued through the entire time history, to ensure that all of the data points are separated by at least the specified time interval. The indices of the surviving data points are also returned by the function, so that a corresponding vector of wind direction values can be assembled if required.

Input Arguments:

ws A vector of peak wind speed values

date_time A corresponding vector of serial date numbers (returned by the MATLAB function *datenum*) giving the date and time of each peak wind speed value in the vector ws

interval Minimum time interval (in days) by which output data must be separated (does not need to be an integer)

Output Arguments:

ws_sep A vector of peak wind speed values separated from neighboring values by at least the specified interval

date_time_sep A corresponding vector of serial date numbers giving the date and time of each peak wind speed in the vector ws_sep

ind_sep A vector giving the indices of the surviving data points; ws and ws_sep are related by the following MATLAB expression: ws_sep = ws(ind_sep). If wd represents a vector of wind direction values corresponding to ws, then a vector of separated wind direction values corresponding to ws_sep can be obtained using the following MATLAB expression: wd_sep = wd(ind_sep).

History of Program Updates

The following is a log of significant updates to the ASOS-WX software since the initial version was made publicly available on February 9, 2006:

- June 8, 2006: A major update to the software was completed. Some of the significant changes are described in the following. Matching of thunderstorm beginning and ending times is no longer performed within the tsBE function. Rather, tsBE now returns all of the reported beginning and ending times extracted from the selected file(s), which generally do not all match. A new function, matchBE, performs the matching of beginning and ending times using the output from tsBE in conjunction with thunderstorm beginning and ending times estimated from manual weather observations, which are now returned by the MW function. The contents of output files created by asos_wx have also been modified to reflect the change in these procedures. The order of the output arguments in set_ts_windows has been changed so that `wind_neg` is now the first output argument and `wind_pos` is the second.
- February 20, 2008: A bug has been fixed in asos_wx, in which the function matchBE was called regardless of what output files were requested, which resulted in an error if the functions MW and tsBE were not required for the selected output file. Another bug has been fixed in the MW function in which an error resulted if no reporting gaps were encountered. (Thanks to Craig Miller at the University of Western Ontario for identifying these bugs!) Because manual thunderstorm observations are required to classify thunderstorm and non-thunderstorm winds, a check has also been added in asos_wx so that only the first output file (“All winds”) will be created if no manual thunderstorm observations are found.