Section 20
The Weather Processing Function

The purpose of the *Weather Processing* function is to move weather data files from a source ETMS path to both local and remote ETMS destination paths. It also is responsible for processing the METAR and TAF weather data files.

The *Weather Processing* function takes the weather data files located at a specific ETMS source path and distributes them to a specified list of target ETMS paths. The *Weather Processing* function continuously monitors the weather data files at a source address and distributes these files, so that the client processes can receive the current weather information.

Processing Overview

The *Weather Processing* function is made up of several processes (See Figure 20-1.).

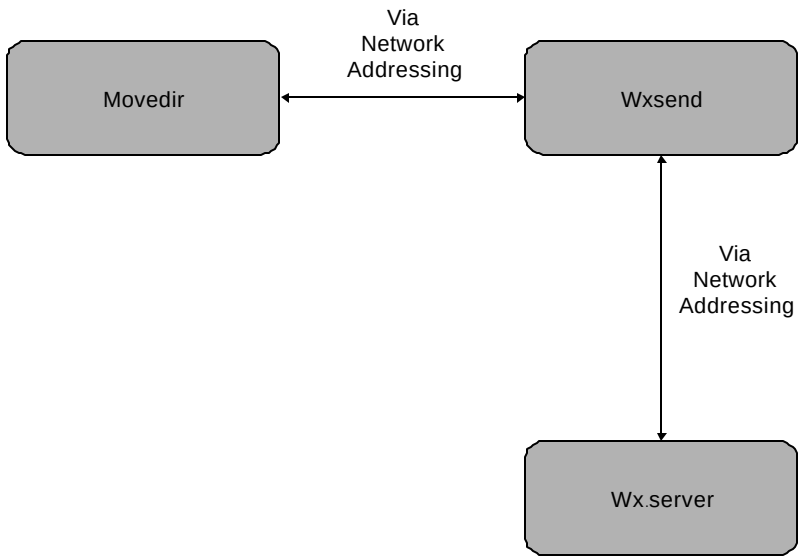

Figure 20-1. Weather Processing Function Processes

- Movedir – Moves all weather data files from a source path to one or more destination paths.

- *Wxsend* – Transfers weather data files delivered locally by the *movedir* process to remote ETMS sites.
- *Wx.server* – Processes METAR and TAF data files received on notification from the *wxsend* process. Processes weather data requests from local/remote sites.

The *Weather Processing* function receives the weather data files from sources outside of the ETMS System, that is WSI. The only types of weather data passed between the *Weather Processing* function and other ETMS components are data reports passed from the *wx.server* process to the *ASD* in reply to a request. Network messages are passed between the weather processes and other ETMS components.

20.1 The Movedir Process

Purpose

The *movedir* process moves all the files that appear in a specified source path to a number of destination paths. The destination paths can be at the same or different sites. All local file moves are performed using a copy command, and all remote file moves are performed using the *FTP* process.

Execution Control

The format to invoke the *movedir* process is shown below:

$ movedir configuration_file

The *movedir* process is normally started by the run-time support process called *nodescan*. If *movedir* fails, it will be restarted by *nodescan*.

Input

The *movedir* process requires one argument, a configuration file. The following is an example of a configuration file:

```
#Required fields include ! which specifies the sleep time
#Required at least two paths in form $site.//node
!1    #1 minute
$opensysd.//atl/to_tsc     #source path
$opensysd.//mci            #destination path - list of wxsends
```

A line that begins with a # is a comment line. An exclamation mark introduces sleep time, that is, how much time the application sleeps before going into the next cycle. *Movedir* examines the source directory every **sleep_time** minutes for new weather data files. There is only one source path and one or more destination paths in a configuration file.

Output

Movedir outputs the weather data files to the specified destinations via a copy or file transfer.

Processing

The *movedir* process runs continuously in the background, spending most of the time in a wait state expecting messages from the ETMS network. *Movedir* creates a directory listing all subdirectories of the source node directory. The source node directory is the directory listed in the configuration file. Periodically, it checks to see if new weather files have arrived in the subdirectories, contained in this listing.

If a new weather data file exists at any of the source subdirectories, *movedir* copies it to the same subdirectory on its local node. The subdirectories at the source must exist in advance, local to the node of the *movedir* process. Next, *movedir* sends a request to the *FTP* process to transfer the file to all destination node locations on its list. If all the replies received from the *FTP* process are *good*, then *movedir* assumes that the transfers were successful and it deletes the file from the source directory. A *bad* reply message from the *FTP* process means that the directory at one of the destination nodes does not exist. Then the *movedir* process sends a message to the *wxsend* process on that particular node to have the directory and subdirectories created. The *movedir* process assumes that the *wxsend* process was successful at creating the directories on its node and it re-transmits the request to the *FTP* process to have the file transferred. *Movedir* notifies all *wxsend* processes on its list of successful file transmissions to the destination locations.

The *movedir* process also responds to requests sent through net.mail. The *movedir* process responds to stats level S0, S1, S2 and S3, through the ETMS net.mail utility. S0 and S1 requests information about weather data files that were moved, the number of messages sent and received, and information on the name of the adaptation file. S2 provides information on the outstanding weather data files in the *movedir* transmit queue. S3 displays a short list of weather data files that were moved. Stats level S4 through S9 are not supported by *movedir.* The *movedir* process also supports the reconfiguration command through the net.mail utility. The format of the command is as shown below:

reconfigure process_address configuration file

Error Conditions and Handling

The error messages that the *movedir* process may generate are:

- ERROR GETTING ADAPT FILENAME – *movedir* could not get the configuration file name (required argument).
- COULD NOT OPEN ADAPT FILE – Invalid configuration file specified.
- MUST PROVIDE SLEEP TIME IN *configuration_file* – No sleep time was provided in the configuration file.

- INSUFFICIENT NUMBER OF PATHS OR TOO MANY FOUND IN *configuration_file* – At least one source and one or more destination paths must be specified in the configuration file. The total number of paths, sources, and destinations, must not exceed (max_entries + 1), currently 15.
- COULD NOT OPEN *configuration_file* – *movedir* was unable to open the configuration file.
- ERROR READING *configuration_file* – *movedir* encountered an error while reading the configuration file.
- UNKNOWN MSG CODE *message code* – *movedir* received message with an unexpected type.
- *source path name* ONLY GOT TO *number* DESTINATION(S) – *movedir* was not successful in the transfer of *source path name* to all the destinations. Only *number* destination transfers were successful.
- *destination path name* WAS BAD DESTINATION – Bad *destination path name* specified.
- INVALID SOURCE PATH NAME – No dot (.) is found in the source path name in the configuration file.
- COULD NOT GET *source path name* TO ANY DESTINATION. *number* TRIED – *movedir* was unable to send a weather data file to any of the destinations on the list.
- COULD NOT COPY *source path name* TO *destination. path name error message* – *movedir* notifies that the transfer/copy of the file from *source path name* to *destination path name* did not succeed; *error message* explains the reason.
- THERE IS NO DIRECTORY STRUCTURE SPECIFIED, GET_ETMS_PATH FAILED - SUICIDE. – *movedir* could not find ETMS_PATH directory, required by the process.
- TIMEOUT: COULD NOT SEND *source path name* TO *node name* – *movedir* could not send file *source path name* to node name even after trying to resend it and after exceeding the expiration time.
- TIMEOUT: RESEND *source path name* TO *node name* – *movedir* tries to resend file *source path name* to node *name*
- RETURN REASON: *error code* TO – *movedir* receives messages on unsuccessful sends to other processes.

20.2 The wxsend Process

Purpose

The *wxsend* process transfers the files delivered to the local ETMS site by the *movedir* process to remote ETMS sites. The *wxsend* process accepts client processes such as the *wx.server* process and *FDB* processes. It notifies each client process of file transfers.

Execution Control

The format to invoke the *wxsend* process is shown below:

$ wxsend configuration_file

The *wxsend* process is normally started by the run-time support process called *nodescan*. If *wxsend* fails, it will be restarted by *nodescan*.

Input

The *wxsend* process requires one argument, a configuration file. The following is an example of a configuration file :

```
#Required fields include * which specifies the default string id
#Required ! to specify home path.  Under = appears list of destination
#pathnames.
*D
!$opensysd.//mci  #source path
=A
#A string
=B
#B string
=C
#

=D
$opensysc.//jfk jet_stream lightning metar taf
=E
```

A line that begins with a # is a comment line. An asterisk place before a string identifier specifies the default string used for the destination path(s). The configuration file must have at least one default string. In the sample file above, *D* is set as the default string. An equal sign followed by a string identifier, informs *wxsend* that zero or more destination paths follows. In the sample file above, the destination paths listed below *=D* are used in the configuration. The format for the destination path is the site name followed by the node name and a list of source subdirectories.

Output

Wxsend outputs the weather data file to the specified remote destinations via file transfers. It also notifies registered clients of weather file transmissions.

Processing

The *wxsend* process runs continuously in the background spending most of the time in a wait state, expecting messages from the ETMS network. It registers with the local site switch as a provider of the *WXSEND* class of services. The *wxsend* process may receive messages from other processes requesting client status, for example, *FDB* and *wxserver*. If the *wxsend* process receives notification from the *movedir* process that a weather data file was successfully transmitted to the node local to *wxsend*, it notifies client processes of these file transmissions. Not all client processes are notified of each file. For example, the *FDB* client process is notified of *winds-aloft* files transmitted and the *wxserver* client process is notified of *METAR/TAF* files transmitted. If the *wxsend* process receives a message from the *movedir* process that the file *transfer/copy* to the *wxsend* node was unsuccessful, it assumes that the local subdirectories do not exists and it creates them. The *wxsend* process also initiates sending the weather data file to a list of destination paths at remote sites. This task is accomplished through requests to the *FTP* process. If any of the destination paths do not exist at the remote sites, the *wxsend* process sends a request to the *CMD* process to have the directories created. After successful creation of the directories, the *wxsend* process re-transmits the file through the *FTP* process. The *wxsend* process abandons the re-transmission of the file, if the second attempt is unsuccessful.

The *wxsend* process also responds to requests sent through net.mail. The *wxsend* process responds to stats level S0, S1, and S2, through the net.mail utility. S0 and S1 requests return the same information. The S0 and S1 conveys information about messages sent and received, the number of clients connected to the *wxsend* process, and general statistics on the success/failure of the file transfers. S2 gives information about the client statistics. S3 displays the number of weather data files waiting for re-transmission. Stats level S4 through S9 are not supported by *wxsend*. The *wxsend* process also supports the reconfiguration command through the net.mail utility. The format of the command is as shown below:

reconfigure process_address configuration file

Error Conditions and Handling

The error messages that the *wxsend* process may generate are:

- COULD NOT OPEN *configuration_file* – *wxsend* could not open the configuration file.
- ERROR READING ADAPT FILE! – An error while reading the configuration file.
- MUST PROVIDE HOME PATH IN *source path* – The configuration file does not have the *source path*, that is, home path
- NUMBER OF STRINGS DOES NOT MATCH NUMBER FOUND IN configuration_file – The number of default strings do not match the number of strings defined in the configuration file.

- THERE IS NO DIRECTORY STRUCTURE SPECIFIED, GET_ETMS_PATH FAILED - SUICIDE – wxsend could not find ETMS directory.

20.3 The wx.server Process

Purpose

The *wx.server* process functions differently from that of the *movedir* and *wxsend* processes. It continuously check the node switch mailbox for any messages. The expected messages to *wx.server* are stats, weather requests, data messages from the *wxsend* process containing METAR/TAF file names and requests for reports from users or *ASD*s. If the *wx.server* process receives notification of a METAR /TAF weather data file, it processes the file into reports and logs the data to database files. It also receives weather data requests from local/remote sites, processes the requests, and returns the information to the requesting process.

Execution Control

The format to invoke the *wx.server* process is shown below:

$ wx.server configuration_file locid_file

The *wx.server* process is normally started by the run-time support process called *nodescan*. If *wx.server* fails, it will be restarted by *nodescan*.

Input

The *wx.server* process requires two arguments, a configuration file and a weather location identifier file (*locid_file*). The following is an example of a configuration file :

```
#WEATHER ADAPTATION FILE
#WXSEND CLASS NAME
!WXSEND
#PRIMARY SITE
$opensysd
#SECONDARY SITE
$opensysd
```

A line that begins with a # is a comment line. An exclamation mark precedes a *wxsend* class name. If the class name is not provided, the *wx.server* process uses a default class name. The current default class name is *WXSEND*. The configuration file must include a primary and (an optional) secondary site name, which are used by the *wxserver* process for locating the *wxsend* process it will register with. The location identifier file provides 3 characters to 4 characters for weather location mapping. The following is an example of a location identifier file:

```
#LOCATION IDENTIFIER TABLE
#5/24/96
#SID   ICAO
#
```

AED	PARD
AFE	PAFE
AFM	PAFM
AHP	PAAP
AJB	PAJB
AJC	PAJC
AKW	PAKW
AQC	PAQC
AWI	PAWI
AZK	PAZK
BVK	PABL
CFK	PACK
DDT	PADT
IBN	PABN
IEC	PAEC
IEM	PAWR
IGT	PAGT

Output

The METAR and TAF weather reports are written to the database files in:

/etms5/wx.server/data/metar_data
/etms5/wx.server/data/taf_data

Wx.server also sends messages containing weather reports in response to weather data requests.

Processing

The *wx.server* process runs continuously in the background. Most of the time it spends in the wait state expecting messages through the ETMS network. The *wx.server* process registers with the *wxsend* process as a client on the current site and registers with the site switch as a provider of the KCW class of services. If the *wx.server* has not received a message from the *wxsend* process within a certain time interval, or it is not currently registered with a *wxsend* process for a certain time interval, it switches from a primary to a secondary site or vice versa and registers with a new *wxsend* process. Registration with the *wxsend* process may be rejected, if the *wx.server* process tries to register with the wrong *wxsend* process.

The *wx.server* process receives notification messages from the *wxsend* process indicating that a METAR or a TAF file has been transmitted to the local node. The *wx.server* process extracts the file name from the message and processes the file as specified in the functional section of this report. The *wx.server* process also receives messages from users and/or *ASD*s requesting weather data reports. The report type is contained in the message and can be one of the following:

- SA – METAR data reports
- FT – TAF data reports
- SF – METAR and TAF data reports

The *wx.server* process also responds to requests sent through net.mail. The *wx.server* process responds to stats level S0 and S1 through the net.mail utility. S0 provides information on the *wx.server*'s primary and secondary site names, METAR/TAF reports, METAR/TAF file names, and METAR/TAF database names. S1 provides general node switch statistics. Stats level S2 through S9 are not supported by the *wx.server*. The *wx.server* process also supports the reconfiguration command through the net.mail utility. The format of the command is as shown below:

process_address (configuration file) OR (primary and secondary site names)

If the configuration file is provided it will read the new configuration file or if the primary/secondary site names are provided, the *wx.server* process will disconnect from the *old wxsend* process and reconnect to the *wxsend* process running on the new site. If neither a file name of sites names are provided, the *wx.server* process will assume that the weather location identifier file has been modified. It will reopen it and read in the location identifiers.

Error Conditions and Handling

The error messages the *wxserver* process may generate are:

- INITIALIZE: NO ADAPTATION FILE SPECIFIED – The adaptation file was not specified during process invocation.
- INITIALIZE: NO LOCID FILE NAME SPECIFIED - USE DEFAULT! - LOCID FILE – The second argument was not specified, so it uses the default for the file *locid_file*.
- ERROR OPENING ADAPTATION FILE: *configuration file* – Error opening the configuration file.
- ERROR READING ADAPTATION FILE: *configuration file* – Error while reading the configuration file.
- ERROR OPENING LOCID FILE: *locid_file* – *wxserver* is unable to open the weather location identifier file: *locid_file*.
- NO PRIMARY SITE NAME IN ADAPT FILE! – *wxserver* requires the primary site name for wxsend in the configuration file.
- INVALID PRIMARY/SECONDARY SITE NAME *site* IN ADAPT FILE! – The primary/secondary site name *site* supplied to *wxserver* in the configuration file is not valid.
- INVALID WXSEND CLASS NAME: *class name* IN ADAPT FILE! – The wxsend class name supplied is not valid.
- INVALID OR UNKNOWN PRIMARY SITE NAME: *site* IN RECONFIGURE! – Reconfigure request contained an invalid primary site name *site*.

- INVALID OR UNKNOWN SECONDARY SITE NAME: *site_name* IN RECONFIGURE! – Reconfigure request contained an invalid secondary site name *site_name*.
- INVALID SECURITY PASSWORD IN RECONFIGURE COMMAND TO ASD – *wxserver* rejects the request to reconfigurebecause the process making the request did not match the *wxserver* security code.
- PROCESS_FILE: ERROR OPENING FILE! – *wxserver* could not open a weather data file.
- INITIALIZE: COULD NOT CREATE WEATHER ERROR FILE – The weather error file could be created.
- THERE IS NO DIRECTORY STRUCTURE SPECIFIED, GET_ETMS_PATH FAILED - SUICIDE. - WXSERVER – Could not find ETMS directory required for the process