

ARES/Data UPDATE:
A PACKET RADIO DATABASE FOR EMERGENCY COMMUNICATIONS WITH CONFERENCE BRIDGE

W. E. Moerner, **WN6I**
1003 Belder Drive
San Jose, California 95120
WN6I @ KB6OWT

Sharon Moerner, **N6MWD**
1003 Belder Drive
San Jose, California 95120
N6MWD @ KB6OWT

David Palmer, **N6KL**
248 Omira Drive
San Jose, California 95123
N6KL @ KB6OWT, CIS: 73357,3157

INTRODUCTION

ARES/Data is a multiple-connect, multiple-port specialized bulletin board system with a conference bridge that is tailored to store and retrieve basic information about people, places, or things in an emergency or disaster. The current program (Version 1.0) contains several enhancements not included in ARES/Data Version 0.1 [see Moerner, W. E., & Palmer, D. (1988), ARES/Data: A Packet Radio Database for Emergency Communications, Proceedings of the Seventh ARRL Computer Networking Conference, 141-144]. [Note: For those interested in the history of ARES/Data, see the above mentioned article and also see Moerner, W. E., Moerner, S., & Palmer, D. (1987), Family Information Database for Emergency Responders, Proceedings of the Sixth ARRL Computer Networking Conference, 131-141.]

New features added to version 1.0 (beyond 0.1) include:

- Multiple ports (up to 4, with 8-32 simultaneous connects each)
- Support for DRSI **PC*PA** packet adapters
- Enhanced **config** file processing with startup files for each TNC
- Update of selected field in a record
- Import/Export facility
- List range in addition to list all
- Beacon facility
- Download file facility from a public directory
- Labels command, including a label for the message field
- Separate paths for database, index files, public directory
- Many enhancements for the sysop screen
- Improved error handling of disk full and disk errors

The key idea behind ARES/Data is that it allows tracking of any needed information that can be organized as four **20-character** fields plus an **80-character** message for each record.

ARES/Data is a system designed for management of information during a widespread emergency that overloads normal communications channels. The

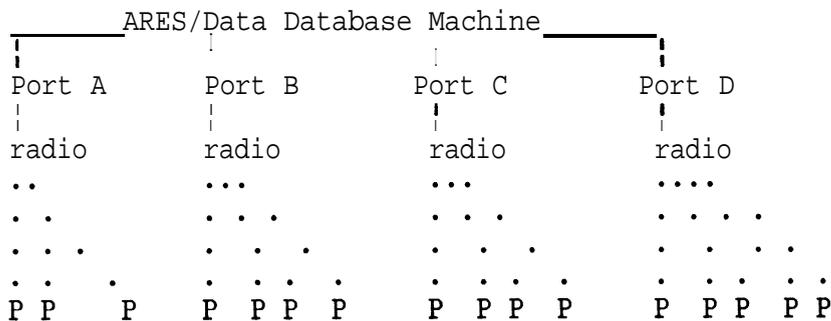
program is conceived to be flexible, so that it can be used without change for both small and large disasters to organize information about victims, evacuees, or even ham radio operators. Examples of situations in which ARES/Data could be used include:

- Track victims of a multiple casualty incident sent to hospitals
- Track ham manpower availability / assignments
- Record evacuees and shelter enrollment
- Track floats in a parade
- Short-message database: Field1 = To Field2 = From Msg = 80 char. message
- House-by-House Damage Assessment
- DX Spotting (!)
- Etc. !

ARES/Data SYSTEM OVERVIEW

Briefly, ARES/Data may be regarded as a specialized, multiple-port, multiple-connect database with a specific command set tailored to the handling of information input, search, listing, and summary requests. In addition, the system provides a full-featured conference bridge so that all connected stations may converse conveniently with one another.

The ARES/Data network is a star network with the ARES/Data database machine at the hub. It looks something like this:



Each "P" represents a remotely-connected packet station which is connected to the ARES/Data database machine. All the remotely-connected stations have shared access to the data in the system. In particular, the packet operators can utilize two groups of functions provided by ARES/Data which are described in detail in this file:

- I. send/receive status requests and current information to/from the ARES/Data database station
- II. send/receive short messages to/from other remotely connected stations or the sysop (Conference Bridge)

As can be seen, there are three major elements to the ARES/Data system:

- o ARES/Data software and database (at one centrally located computer)
- o Remote packet stations connected to the central node
- o Conference Bridge

The central element of the ARES/Data system is the ARES/Data database program running on an IBM Personal Computer or compatible. The ARES/Data program collects and collates current information about people (or other items) according to the needs of the incident, and maintains the database on floppy disk or hard disk at this central computer. The sysop at this computer can execute all database functions.

If remote access is desired to the information in the database, this computer may be connected to a packet radio channel or channels through up to four ports, each of which can be either a TNC with **WA8DED** firmware or a DRSI **PC*PA** packet radio adapter. In this manner, other remote packet stations can connect to the ARES/Data machine and thus gain access to the stored information. This information can be updated or queried by the sysop or any of the remote packet stations that are all simultaneously connected to the main database station.

At any time, the sysop and any of the connected stations can communicate with each other by using a simple "tell" command. This "conference bridge" actually implements a star-shaped network in which the central database station relays all of the message traffic. (As noted, the central element of the ARES/Data system is the computer on which the ARES/Data program is running.)

Hardware Requirements

The ARES/Data program will run on any IBM Personal Computer or **IBM-**compatible system running DOS V. 3.2 or later. Although it can be run on a computer having about 400 KB of memory or more with at least one floppy disk, a hard disk is recommended as it increases the allowable size of the database and improves performance. [IBM is a registered trademark of International Business Machines Corporation.]

To enable the remote packet access features, the sysop also needs at least a (i) DRSI **PC*PA** packet adapter, cable, and transceiver or (ii) RS-232 serial port, TNC containing **WA8DED** firmware (TNC-1, TNC-2, AEA PK-87, AEA **PK-88**), cable, and transceiver. NOTE: the remotely connecting packet stations may use ANY AX.25 TNC.

DESCRIPTION and OPERATION of the ARES/Data SYSTEM v. 1.0

The ARES/Data Program

The ARES/Data software was written by W. E. Moerner, **WN6I**, and David Palmer, **N6KL**, with the ideas and support of a committee of hams from the Santa Clara County Amateur Radio Emergency Service (ARES). It is a copyrighted program, available without charge to anyone interested. The authors cannot and will not accept remuneration for this program which is provided as a public service only. The ARES/Data program is written in Turbo Pascal Version 5.5, and uses Turbo Database Toolbox for management and indexing of its B-plus structured tree. [Turbo Pascal and Turbo Database Toolbox are trademarks of Borland International, Inc.]

ARES/Data may be run in either of two modes: (i) stand-alone with no TNC support and no remote access, or (ii) by changing the configuration file, the program will control one or more TNCs to allow multiple, simultaneous remote connections. Use of WA8DED firmware with the central computer is necessary because WA8DED host mode is used for communication **between** the computer and the TNC. WA8DED firmware is currently available for the TAPR TNC-1 and TNC-2 as well as the AEA PK-87. We emphasize that NO REQUIREMENT IS PLACED ON THE OTHER TNC'S CONNECTED TO THE ARES/Data DATABASE MACHINE, except that they use AX.25 link-layer protocol.

General Rules for Current Information Input/Search Requests

All basic commands can be entered either at the main ARES/Data keyboard or at any one of the remotely connected packet stations. The operator at the main ARES/Data keyboard (the "sysop") has an additional set of commands that allow direct communication with the TNC, the printing of a log, backups, and disk report files.

ARES/Data organizes the incoming data into "records", which can be viewed as a group of pieces of information about some particular person, place, or thing. Each record is given a unique "record number" by the program. Each record consists of four main items or "fields" plus a message item. The information in the four main fields can be used as keywords for searching as required.

Syntax for Current Information Input

field1,field2,field3,field4,message<cr>

(<cr> means carriage return)

To add a record to the database, the operator simply enters (in order) the four fields and any message, separating each field with a comma. Within a field, leading and trailing blanks are ignored, but imbedded blanks ARE significant. If no value is desired for a particular field, just skip the field by adding an extra comma. The database will fill that field with ten blank characters.

Fields 1 through 4

The four main fields are totally general. Each can have up to 20 characters, with imbedded blanks. Entries can be in upper or lower case, or a mixture, but are converted to UPPER case before being stored in the database. The meaning of each field is defined at the beginning of the event depending upon the nature of the event. The sysop can issue a "labels" command that will give specific names to each of the four fields to help the operators remember what they mean. Similarly, the remote packet operator can type "labels<cr>" to see the current label definitions.

Message

MESSAGE is an optional, free-form field that can be up to 80 characters in length. It could contain a message, a phone number, an address, or other information deemed useful for the incident. The distinction between

the "fields" and the "message" is that the "fields" are organized internally by the program so that the packet operator can request searches and summaries on the information in any one of the four fields. Searches and summaries cannot be performed on the information in the message field.

Examples of Data Input with Sample Responses from ARES/Data

```
85553195,joe,12,sj34<cr>
response-> 1040: data input accepted, #234.
```

```
Johnson,Mary,93445,sj13, home 2333 Alsace Ln SJ 617-555-2368<cr>
response-> 2134: data input accepted, #114.
```

All of the input information is stored in the database as a record of the status and location of a particular person, place, or thing at a particular time and date. The time and date are added automatically by the ARES/Data program. Further STATUS INPUT packets for the same person, place, or thing will also be saved in the database. The time and date identifies the most recent information.

Correcting and/or Updating Information

Two options are available if incorrect information is entered into the database or if the information in a particular record has changed.

Deleting an Entire Record

You can ask the sysop to delete the bad entry by typing a message to him/her, such as:

```
tell sysop oops, typo. pse delete #234.<cr>
```

OR, if the sysop has enabled the remote delete feature, a remotely connected packet station can delete this by using the delete command:

```
d nnnn<cr> (where nnnn = record number)
```

This function is always enabled at the sysop keyboard. Its use by remotely connected packet stations is controlled initially by the configuration file during program startup. Thereafter, the sysop can disable or enable this function as necessary.

Changing or Updating a Particular Field of a Particular Record

The syntax for updating fieldm of record nnnn is:

```
#nnnn,m=new text for fieldm<cr>
```

For example, suppose it was desired to change the value in field3 of record 235 to "shelter 9". This is done by typing:

```
#235,3=shelter 9<cr>
```

Note that when correcting or updating a single field like this, the time and date for the record are not changed. ARES/Data responds by showing the old values for record 235, along with the newly updated values,

Requesting Information from the ARES/Data Database

The packet operator can request several types of searches of the ARES/Data database. S/he can request a search for a specific value of any one of the four main fields. In this case, the ARES/Data program sends back to the packet operator a status report listing all entries in the database having the specified value for the selected field. In addition, the operator can request a "wildcard" search, which looks for any entries in a specific field that START with a particular string. The packet op can also request a summary for any one of the four fields, which is a list of the number of entries in the database for each distinct value of that field (see below). The operator can list single records in the database by specifying the record number.

Syntax for Search Requests

/1,value<cr>	or	?1,value<cr>	Searches for "value" in field 1
/2,value<cr>	or	?2,value<cr>	Searches for "value" in field 2
/3,value<cr>	or	?3,value<cr>	Searches for "value" in field 3
/4,value<cr>	or	?4,value<cr>	Searches for "value" in field 4

This type of packet instructs the database to look for ALL entries with the same entry as "value" in the specified field. The string "value" must exactly match what was originally typed in for the selected field, with leading and trailing blanks removed, and without regard for case. The initial character of the search request can be "/" or "?" - use whichever is most convenient. The two formats are handled identically. A status report listing all information for each match is sent back to the requesting packet station. The first line gives the search value and the field number. At the end of the report, the line:

"ARES/Data Search done at 1534, nn hits."

is sent, which signifies no more information coming, and that "nn" matches (or hits) were found in the database.

Wildcard Search or Partial Search

The syntax for a **wildcard** or partial search is:

/n,val*<cr>

where "n" is the field number, and "val*" indicates a search for all entries in fieldn that start with the characters "val". The response from the system is identical to that for an exact search request. This is very useful if a particular field has been defined to hold more than one piece of information. For example, suppose field 1 is defined to be "Lastname-Firstname" so that Bill Jones would be entered by the line:

Jones-Bill, shelter3, OK, 444-555-1212, message<cr>

Without knowing Mr. Jones' first name, or whether his data was entered as "Bill" or "William," one can still search for him in the database by typing:

?1,Jone*<cr>

The database would retrieve all records whose first field began with the characters "JONE".

Syntax for Summary Requests

\$1<cr>	produces a list of all distinct values for field1, with the number of entries in the database for each
\$2<cr>	similar, except summarize on field 2
\$3<cr>	similar, except summarize on field 3
\$4<cr>	similar, except summarize on field 4

A Summary Command is provided that prints a breakdown of the number of like-named items for any particular field. For example, suppose field 3 were defined to be the shelter name. After the packet operator types "**\$3<cr>**", ARES/Data sends a summary on field 3, which may be interpreted as a list of shelters, with the number of people that have checked in to each shelter.

Sample Output from a Summary Request

```
Database summary for Shelter at 1455 on 23
OAK GROVE          3
PIONEERHS         20
EASTVIEW         66
SHLTR5          37
ARES/Data done at 1456, found 4 distinct values, entire DB has 153
records.
```

Listing Specific Entries (Records) in the Database

1 nnn<cr> Lists record nnn

Each record is automatically assigned a unique record number for identification purposes. The response will be a short header showing the labels for the various fields, and then the complete information for record nnn.

1 nnn,mmm<cr> Lists records numbered from nnn to mmm

1 all<cr> List ALL records in the database.

WARNING: be careful with this command, as it may cause a large number of packets to be sent on the channel. Stop an undesired "list all" by simply disconnecting from the ARES/Data machine. This will cause no harm. Wait a bit, then just reconnect.

Downloading Files

To download a file from the database machine, type:

```
get filename<cr>
```

This facility is intended to be used and controlled by the sysop in the sense that s/he should tell the remote users what filename to download. There is no directory facility or anything like that, because such functions are handled better by a standard mailbox program such as BB (by AA4RE) or WORLI. One file that is provided with ARES/Data is an "info" file, which gives more information of interest to general users. If the sysop has copied this file to the public subdirectory, a station can download it by typing:

```
get info<cr>
```

Conference Bridge (Roundtable)

This feature allows any connected station to send messages to other connected stations or to the sysop. The conference bridge illustrates how the ARES/Data system operates as a hub-oriented network, with all transactions passing through the central database station.

Users Command

The users command in the form "users<cr>" or "u<cr>" returns a list of the callsigns of packet stations currently connected to ARES/Data. The response is of the form:

```
Users at WN6I-1 (AXO): N6KL W6BB-3 W6XYZ WB6MRQ-7
Users at WN6I-4 (DRO): 0:N6KL-3 1:N5BZK 3:AA4RE-12
```

Note that there is one line for each port defined in the ARES/Data system; who is using which port can be seen. The callsigns used by ARES/Data for the various defined ports do not have to be identical. After the database callsign, the port name defined by the sysop during startup is shown in parentheses. Note also that for the DRSI packet adapter, several radios and even several boards can be attached to the database machine. All the users connecting to the DRSI adapters are treated as being on one port of the ARES/Data network. To refer specifically to the user on DRSI sub-port 1, put a "1:" in front of the callsign: "1:N5BZK". In general, however, this is not really necessary, since as far as ARES/Data is concerned, "N5BZK" or "BZK" will do just as well (see below).

Tell Command

The Tell command allows connected packet stations to use ARES/Data as a conference bridge, or roundtable. The general format is:

```
tell callsign message<cr> OR t callsign message<cr>
```

For example:

```
tell w6bb-3 The food truck just arrived at SJ12<cr>
```

The message "We have lots of people here at SJ12" is sent to the connected station **W6BB-3** prefaced by a time stamp and the call of the station originating the tell command. In this case, if the tell command was sent by **W6XYX**, **W6BB-3** sees:

1230 **W6XYZ**> The food truck just arrived at SJ12

It is not necessary to enter the entire callsign-just the suffix or some other substring will do. If several connected callsigns contain the substring, each station will get the message. The special **callsign "*" or "all"** is used to send a message to all connected stations. The special **callsign "sysop"** sends the message to the sysop at the ARES/Data database station.

EXAMPLES OF HOW TO USE ARES/Data IN SPECIFIC DISASTER SCENARIOS

One of the major virtues of this system is that the meaning of the stored information is not defined in advance. In this manner, an ARES/Data system can be left in unattended operation 24 hours a day, and then be put to use instantly, in a variety of ways, depending upon the particular disaster or emergency at hand. In a given event, the sysop can issue a "labels" command that gives particular meaning to each of the fields and the message, so that all know how ARES/Data is being used for that event.

In an evacuation, you may want to keep track of evacuees at shelters. Then you may want the fields to mean:

Name, Shelter, Status, **PhoneNumber**, Contact person.

In a multiple-casualty incident, you may want to keep track of victim transportation. You may then define the fields to mean:

Triage number, Sex/Age, Ambulance, Hospital, Condition.

If you need to maintain a roster for a ham radio staffing situation, the fields could be:

Call, Name, Location, Shift, phone number for cancellation.

If you were in a disaster situation where damage assessment and damage reports were needed, consider having the following fields:

Coded type of damage, Location, Number of injuries, Callsign, comment.

There are many more possibilities, of course. This is why the exact definitions of the various fields are not defined in advance. In any given situation, more information than will fit into four fields and a comment field might be needed. However, on today's 1200 baud packet radio networks, not much more information per record can be accommodated without restricting the total number of records that can be handled in a reasonable time.

HOW TO OBTAIN YOUR COPY OF **THE** ARES/Data PROGRAM

The ARES/Data program, a relative of and successor to the FINDER program, is available free of charge. The current version is 1.0, which operates as described in this paper. It is possible to download the program via CompuServe's **HamNET** in data library 9 and it is also available for anonymous FTP from the **SanJose** TCP/IP gateway as file **ARESDA10.ARC**. In addition, a copy of the program along with the documentation is available for non-commercial, non-profit use from **WN6I** or **N6KL** by sending a blank, formatted 5 1/4" (360 kB) or 3 1/2" (720 kB) floppy in a mailer with return postage stamps. The cost to you is the cost of the diskette and postage. No other compensation can or will be accepted--please do not send money. We have included a configuration file facility so that you can tailor many parameters to your specific system. Of course, you may also obtain ARES/Data from anyone who already has a copy, and you're encouraged to give the program to other interested radio amateurs.

ACKNOWLEDGEMENTS

Since the ARES/Data concept is a generalization of the FINDER system, the deliberations of the FINDER committee in the Santa Clara Valley Section of the ARRL have contributed to the present form of ARES/Data. In addition, our thanks also go to all the ham radio operators in the Santa Clara Valley Section that have participated in the various alpha and beta tests of the FINDER and ARES/Data systems.