Fido's Operating Manual

This manual is new for version 10 of Fido. It covers all of the previous three manuals, and supercedes them. This is totally new documentation, though you will find occasional paragraphs the same. Hopefully, this new organization makes more sense.

1 Nov. 84

Fido Software
Tom Jennings
2269 Market St. #118
San Francisco CA 94114
Fido #1 415/864-1418 300/1200 24 hrs.

Fido supports only the following hardware, though others may be oneday supported, in the definitely unspecified future:

IBM PC, XT, AT, Junior
DEC Rainbow, 100, 100+, 100B
Otrona Attache 8:16

D.C. Hayes SmartModem 300, 1200 and clones
  EXCEPT Anchor Automation
Racal Vadic VA212
DEC DF03
Novation SmartCat series

Fido requires 256K of memory, and MSDOS 2.00 or higher, and at least two floppies. It is recommended that you use a hard disk.

```
                                                _
                           The World's First   /  \
                              BBS Network     /|oo \
                              * FidoNet *    (_|  /_)
                                              _`@/_ \    _
                                             |     | \   \\
                                             | (*) |  \   ))
                                ______       |__U__| /  \//
                               / FIDO \       _//|| _\   /
                              (________)     (_/(_|(_____/
                                    (j. madill)
```

The following people have contributed greatly to the design of Fido and FidoNet. I probably omitted many people from this list; apologies ahead of time. Thanks also to the many Fido sysops and callers that contributed ideas (and found the many bugs).

Special thanks to John Madill, who thought out many of the nicer Fido features, and did much of my debugging for me.

Fido and FidoNet Design Crew:

John Madill, Baltimore, MD	(Fido #2)
Tony Clark, St. Louis, MO	(Fido #4)
Danny Feinsmith, NYC, NY	(Fido #8)
Jim Ryan, Danbury, CT	(Fido #9)
Ben Baker, St. Louis, MO	(Fido #10)
Vern Crawford, San Jose, CA	(Fido #13)

FidoNet Scheduling and Routing:

The St. Louis DECUS Group:
Ken Kaplan, Mike Mellinger, Jon Wichman
(Fido's 16, 17, 22)

And general thanks to all of the other Fido sysops for suggestions.

Fido is a package of programs that allows an MSDOS computer and modem to be used as:

- A Bulletin Board
- A Private Mail System
- For Binary and text file transfers
- Has Extremely high system security

Fido supports up to 99 seperate message and file areas. There are "privelege levels" for each Fido command, each user, and each area. These can be intermixed to allow free access to all areas, limited access to some areas, or locking each user into a specific area.

Fido logs information about it's use continuously; each users access and their activity, mail systet activity, and statistics on overall system use.

FidoNet

Fido also contains a true electronic mail and networking system, (FidoNet) using ordinary phone lines. Up to 250 physically seperate Fido systems may be interconnected for message and file transfers, under automatic control. Each of the 250 Fido systems can be a host, with up to 250 connected to it.

Fidonet is a true, full featured electronic mail network. It supports host systems, message forwarding and routing, and cost accounting.

Routing and forwarding can be used to direct mail for one or more Fido systems to a different Fido system, which will later send it on to the correct destination.

FidoNet can be configured to use a centralized host, or an amorphous network of independent systems for higher reliability.

Cost accounting is done on a user by user basis, for each message entered. (By Jan 85 it will also have host to host charge reversals.)

Hardware Support

- All IBM PCs: PC, XT, Jr. AT and most clones

- DEC Rainbow 100, 100+, 100B
- Otrona Attache 8:16

- Hayes Smartmodems and clones
- Racal Vadic VA212
- Novation SmartCat series

Table of Contents

- Initial Installation -

- Final Installation -

- Hardware Installation -

- More Detailed Configuration -

- Internal Stuff -

- Using Fido's Commands -

- Sysop Commands and Features -

- FidoNet Electronic Mail -

Introduction to Fido

If you are really lucky, this manual will help you create and customize a Fido Bulletin Board system to your liking. If you are not lucky, you will become hopelessly lost and confused. This can be summed up as:

"Don't let you mouth write no check your tail can't cash."

Which means that I don't promise that these manuals are useful, never mind good, though what little information you find should be accurate, if a bit poorly organized and hard to find.

Or:

"These manuals stink!"

As one sysop succinctly put forth. This is the third and most extensive major rewrite of the Fido documentation, and may even be better than the previous attempts. To help things further, Fido itself seems to have stabilized, making the job easier.

Step By Step Installation

Fido is a software system, not just a single program. Besides the program itself, Fido requires a number of disk files, subdirectories or other disks to hold messages and downloadable files, and obviously a modem and cable.

While Fido requires many disk files to run properly, almost all of them are created and maintained by Fido itself. The exceptions are the batch file to actually run Fido, the welcome messages and other text messages that Fido can display, and various other lists and messages. These are covered in detail later.

You can think of Fido as having an inside and an outside. The inside is what you can change and see by running the Fido program. These are things such as setting up message and file areas, privelege levels for the various commands, scheduled events such as FidoNet, and the like. These are all accessed by Fido commands.

The outside consists of the MSDOS batch file, subdirectories for the message and file areas, signon and welcome messages that the system operator creates, and the modem system. While these are of course used by Fido (messages are displayed, modem is used to answer and dial, etc) they cannot be created or modified from within Fido.

Simply put, there are four stages to follow to get Fido properly installed. These are:

(1) The Environment:
Making subdirectories, formatting blank diskettes if necessary, determining how many and where to put the message and file areas.

(2) Inside:
Running Fido from the local keyboard (instead of the modem) in order to set up Fido to match the environment you created. Fido must be told about each message and file area subdirectory or diskette set up earlier.

(3) Outside:
Customizing the batch file to run Fido from; this contains the command line parameters for users time limits, and other operating details.

(4) Hardware:
Making the modem go. Up to this point, the modem is not needed at all, and can be ignored. If you have a modem already connected and running with a communications program, then this step is easy.

Step 1: Setting Up Message and File Areas

The first step is to figure out what you want your system to look like. This depends on your needs, your hardware (mainly, how much disk space do you have) and the neeeds of your users.

It is not possible to cover all the combinations that Fido can be configured in, and the one you end up with will probably change as your needs change. Therefore, this manual will cover a single simple configuration; others are set up in the same way, and once it is understood, is fairly straightforward.

If you follow this installation guide exactly, you will end up with a Fido Bulletin Board system that has the following characteristics:

- A public access system

- Two independent Message Areas

- Two independent File Areas

- An additional, special Message Area where users can find special messages from the system operator.

- Ready to be used with a DC Hayes SmartModem 1200B

Any and all of this can be modified, either now (if you are confident) or later, after you get this example one running. While this manual is arranged in the order covered in the introduction, you may actually perform any steps in any order; however, this order has been found, by trial and error, to be the simplest and easiest to follow.

Step 2: The Environment

Choose how you want your disk(s) laid out. If you have a hard disk, then use that; create a single subdirectory in the root, called FIDO (or whatever) log into it, and perform the steps below from within it.

```
CD \                           log into the root
MD FIDO                        make a FIDO directory
CD FIDO                        enter the FIDO directory
```

If you have two floppies, you should format two new diskettes, and make one bootable (format it with /S, to install the system on it). This will become your A: diskette. On it, put the minimum tools you will need: some sort of editor or word processor (for making welcome messages, etc) and the fido .EXE files. Put the other formatted diskette in B:.

For all systems, you should set the number of BUFFERS in CONFIG.SYS on the booted diskette. The performance increase will be enormous. If you do NOT do this, then your Fido system will run very slowly, and annoy all of your users. If you know how to use CONFIG.SYS, then just do it. If not, then on your boot disk do the following:

```
CD \                            log into the root
COPY CON CONFIG.SYS             copy keyboard to CONFIG.SYS
BUFFERS = 50                    type this exactly,
FILES = 20                      this too
^Z                              Enter Control-Z, then return

Copied 1 file(s)                Message from MSDOS
```

Next time you boot, MSDOS will read CONFIG.SYS and set the number of buffers to 50. (It will not do it until then.)

Anyways ...

Before even thinking about running any programs, the subdirectories and such need to be created. As mentioned above, we will end up with two message areas and two file areas.

Message areas:

GENERAL	Msg area #1
TECH	Msg area #2

File Areas:

IBMPC	File area #1
CPM80	File Area #2

Special Areas:

ANSWERS	Special Msg area

First, create these subdirectories using the MD or MKDIR command:

```
MD GENERAL
MD TECH
MD IBMPC
MD CPM80
MD ANSWERS
```

Now the subdirectories exist on your disk. These are where Fido will put messages and files that can be accessed by users. (Of course, there areen't any yet, don't worry about it.)

Now for a tedious part. Now you must log into each area, and make a file called DIR.BBS. This is a very small file, and contains a one line description of what the subdirectory is for. (No, Fido cannot read; this small description is merely displayed to a user when they log into an area.)

This can be done with your word processor, or more simply with the COPY command.

```
CD GENERAL                      enter the GENERAL subdir
COPY CON DIR.BBS                make DIR.BBS
General Interest Messages       one line descrption
^Z                              Control-Z, then ENTER

1 File(s) Copied                MSDOS message to you
CD ..                           log into previous subdir
```

Repeat this for each area above, entering the description for each, instead of the "General Interest Messages" in the example above.

At this point, you have made all the basic subdirectories for Fido. Changes or additions can be made later, following the same basic steps.

Step 3: Starting Fido

Now that the basic subdirectories have been made, you must run Fido and tell it where to find them. There is one last step before you do this; it is to make sure that Fido has the minimum files it needs to run.

While they are not absolutely necessary to run, the HELP files are handy, since you probably havent memorized all the Fido commands yet. All that needs to be done is to copy them into the HELP directory, created above.

While logged into the main FIDO directory, or the root for floppy systems, put the Fido distribution diskette into B:, and:

```
        COPY B:*.HLP
```

You also need as a minimum the initial user list, USER.BBS, and the initial system file, SYSTEM.BBS. You will also need the Fido program, of course. Copy these from the distribution diskette, to the Fido subdirectory.

```
        COPY B:USER.BBS
        COPY B:SYSTEM.BBS
        COPY B:FIDO_IBM.EXE     (or _DEC for DEC, etc)
```

Now, Fido has all it needs to be run. Fido has many command line options, which are covered in a later section. Here, you need only one:

```
        /T              Test Mode

FIDO_IBM /T             (or _DEC for DEC, etc)
```

After a few seconds, you will see the signon message, and a prompt for your name. You will NOT see any nice welcome messages and so on; these are created later.

At the prompt for your name, fill in as follows:

```
Your FIRST Name: SYSOP <enter>
Your  LAST Name: <enter>
Wait ...
Password: WHAT <enter>
```

When you enter the password, it will be displayed as dots, for security reasons. After some messages from Fido, you will be at the MAIN Command prompt.

There are a number of commands that do NOT show up in the prompt. These are the sysop commands, and are the digits 1 to 9. Only a user with SYSOP priveleges may use them. (The initial user list contains one user, named SYSOP, whose privelege level is also SYSOP.)

The sysop-only commands are covered in detail in elsewhere; however the minimum needed information will be provided here.

Fido keeps track of message and file subdirectories by storing them in small lists. Each list contains one Message Area subdirectory, one File Area subdirectory, and one HELP subdirectory (not used here). There can be up to 99 lists. When a user chooses a message or file area, they are really choosing one of the lists.

You must now store the names of the subdirectories in these lists, so that Fido will know where to find them, and so that users can select them.

The "1" command is used to maintain these lists. The lists are numbered 0 to 99; you select one by:

```
1 <n>           Where <n> is a number 0 - 99
```

First we set area #1, using the subdirectory names chosen before. (Only what you enter is shown here; Fido will display the Main Command prompt and other messages also.)

```
1 1             Select list 1
1 GENERAL\M     Set Message Area name
1 IBMPC\F       Set File Download Area name
1 IBMPC\U       Set File Upload Area
1 S             Save the changes
```

The first set of areas is now set. You can view your changes by entering "1" by itself.

Now repeat this process for the second set of areas.

```
1 2             Select list 2
1 TECH\M        Set Message Area
1 CPM80\F       Set File Download Area
1 CPM80\U       Set File Upload Area
1 S             Save changes
```

The third list will contain only the ANSWERS message area, there is no corresponding File Area. This is set in the same way as the other areas.

```
1 3             Select list 3
1 ANSWERS\M     Set Message Area
1 F             Set BLANK File Area
1 U             Set BLANK Upload area
1 S             Save the changes
```

The last list to set is the "special" list, number 0. Only the Message Area is used from this list. This area is where comments to the sysop go, using the G (GoodBye) command. Usually (and in this example) this area will be the same as one of the regular areas; ANSWERS in this instance.

Step 4: Setting Up Areas

This is so that the sysop can reply to messages left at GoodBye, and users can read them by choosing the ANSWER area.

```
1 0             Select list 0
1 ANSWERS\M     Set message area
1 S
```

At this point Fido is more or less usable in Test Mode (/T). You can now go into the various areas, enter messages and operate most of the commands. There will not be any files available yet, nor any messages. You should go through each area and make sure it is set up as you think it is.

Message Areas

From the MAIN Command prompt, use M to enter the Message Section. Check that you have the names of the message areas (subdirectories) set properly. The A command displays the message areas available, and lets you choose one. You should see three areas, and the small descriptions you entered previously. If not, something is wrong; go back over the previous steps and check. Probably just an incorrectly entered name. If all is well, pick one of the areas, for example "1" for area #1, GENERAL.

Use the E command to enter a sample message. After saving it, it should say "Saving your message". If it does not, but says "DISK FULL: Cannot save your message" it is either that you disk is really full (not likely at this point!) or the subdirectory name you entered is incorrect. You must manually create, with the MSDOS MKDIR command, the subdirectories where Fido expects to find them.

Return to the MAIN Command prompt, then enter the Files Section, and check the areas there, using the A command as before. You should see the area descriptions. Note however, that there will not be any files for downloading.

Completing Installation

The software is more or less ready to be used. At this point, there are a number of options. The sections you need are listed below; be sure to browse through the others for ideas and tips.

ALL SYSTEMS:

Fido Message Files

How to customize the welcome messages, bulletins, etc., in common with all Fido configurations.

Hints and Tips

Tips on speed, problems to watch out for, and ways to improve overall performance. Some things you can do and shouldnt do.

PICK ONE:

Public System

How to make a public access bulletin board, so that new users can log in for themselves, and how to handle special cases.

Semi-Public System

How to make a semi-private system, where new users can log themselves in, but have restricted priveleges.

Private System

Where new users are not allowed; the sysop has to manually enter new users. How to keep security absolute.

ALL SYSTEMS:

Modem Installation

Isntalling and configuring the modem. Many modems are listed as installed in actual installations. Tips on how to fix problems.

AND THEN ...

FidoNet Installation

Setting up your Fido as a node in the public domain electronic mail network, and what this means.

System Messages and Bulletins

There are a number of text files that Fido will display at various points; welcome messages, the initial signon, the bulletins and so forth. You will need to customize them to your taste, or delete them if they areen't needed. Fido does not require the presence of any of these, though you should probably use most of them.

These are the ones in common with all Fido setups. There is a section for each configuration: public access, private, and semiprivate.

WELCOME1.BBS

This is the first text that a user sees, after determining baud rate by hitting the CR or ENTER key. Most of the public Fidos use the "doggie", already provided on the distribution diskette.

WELCOME2.BBS

Presented after a user enters their name and password. Just a chance to give logged in users news or other information. Many systems run without WELCOME2.BBS.

BULLETIN.BBS

Usually where frequently changing information is posted, for all user to see. This bulletin can also be read with the B (Bulletin) command from the MAIN Command prompt.

EDTORIAL.BBS

Similar to BULLETIN.BBS, except Fido does not display it automatically. It can be accessed by the E command from the Main Section.

Making a Public System

When set up as in the previous sections, Fido is by default a public access system. However, there are special files that you can use to make life easier for you and your users.

NEWUSER1.BBS

This is displayed once for new users, just before Fido requests their city and state, and to choose a password. It should probably explain the need for a password, etc.

NEWUSER2.BBS

This is displayed right after a new user enters their city, state, and password. Usually, a simple tutorial of the system, what to expect, where to find things.

If a previously logged in user forgets their password, after three attempts at entering a password, and no match is found, then there are a few options.

The presence or absence of QNOPWD.BBS, a special questionaire, determines how Fido treats new users.

OPTION 1: LEAVE A MESSAGE TO THE SYSOP

Make sure that you do not have a file named QNOPWD.BBS; if it exists, then Option 2 will be used.

NOPWD.BBS is displayed, then they get to leave a message to the sysop, identical to when a logged in user uses the "G" command to leave a comment. After leaving a comment, they are logged off. They have 10 minutes (or the specified /S value) to complete the message.

OPTION 2: FILL OUT A QUESTIONAIRE
OPTION 3: MESSAGE THEN LOG OFF

For these two options, QNOPWD.BBS is used. This is a special file, a questionaire. Briefly, a questionaire causes Fido to ask the user questions contained in it, and records the answers (in this case) in ANOPWD.BBS. If used, you should look at this file daily to check for new users. You can delete it whenever necessary, Fido will create a new one.

Alternatively, QNOPWD.BBS can contain only text to be displayed, and no questions at all. Usually, this is the easiest thing to do; just tell them to log on again, but use a new name and password.

If QNOPWD.BBS exists, the user gets to fill it out (or not) and is then logged off; they do not get to leave a message to the sysop as in Option 1.

Semi-Private (Restricted) Public System

If you want to run a public board, but restrict new users, this can be easily done.

COMMAND LINE SWITCH:

```
        /P      New Users get DISGRACE privelege
```

When new users log on, they are assigned DISGRACE privelege. Then, by using the "3" command to set the privelege level for the Fido commands, you can restrict the commands available to new users. For instance, they can read messages, but not enter any, not be able to change their name or password, access the file area or the message area, or any other Fido command.

If you use this method, then new users have to be upgraded manually, using the SYSOP utility program.

Other than the new users privelege level, the is identical to a public system.

Making Fido a Private System

After following the previous steps to get Fido up as a public system, follow these to make it a private system.

A private system is one where only already existing callers can log on; new callers are prevented from entering the system. You can let new callers leave a message to the sysop, fill out a questionaire, or merely log them off, with or without a messaeg as to why.

When running a private system, callers must be entered manually, using the SYSOP utility program. Operation of SYSOP is covered elsewhere.

The presence or absence of QNOPWD.BBS, a special questionaire, determines how Fido treats new users.

COMMAND LINE SWITCH:

/M Private System

Add this switch to the command line in RUNBBS.BAT. It prevents new users from being added to the user list.

OPTION 1: NEW USERS CAN LEAVE A MESSAGE TO THE SYSOP

Make sure that you do not have a file named QNOPWD.BBS; if it exists, then Option 2 will be used.

This is one way in which you can make new users request entry to your system; NOPWD.BBS is displayed, then they get to leave a message to the sysop, identical to when a logged in user uses the "G" command to leave a comment. After leaving a comment, they are logged off. They have 10 minutes (or the specified /S value) to complete the message.

NOPWD.BBS is just a text file, that is displayed to new users, or users that forgot their password. In it, you can explain that this is a private system, and request that they enter whatever information you require.

OPTION 2: NEW USERS FILL OUT A QUESTIONAIRE
OPTION 3: NEW USERS GET LOGGED OFF IMMEDIATELY

For these two options, QNOPWD.BBS is used. This is a special file, a questionaire. Briefly, a questionaire causes Fido to ask the user questions contained in it, and records the answers (in this case) in ANOPWD.BBS. You should look at this file daily to check for new users. You can delete it whenever necessary, Fido will create a new one.

Alternatively, QNOPWD.BBS can contain only text to be displayed, and no questions at all. This is frequently

used to tell users that the system is totally private, and access to it cannot be had.

If QNOPWD.BBS exists, the user gets to fill it out (or not) and is then logged off; they do not get to leave a message to the sysop as in Option 1.

Installing the Modem

The modem should be installed ONLY after the software is ready for it. There is no need for a modem for configuring the software.

In order to test Fido and the modem, you will need a second computer and modem. This may be difficult, but is the only way by which to test the modem. Any modem and computer or terminal may be used; all you require is some sort of communications program to call your Fido with.

For IBM PC versions, you can use a command line switch to select either COM1 or COM2. The default is COM1. If you need to use COM2, add to your command line in RUNBBS.BAT:

```
        /2              COM2
```

Later on there is a section of modem/computer combinations that have worked in other installations. If your system is in there, you're all set. Probably a good idea to read on anyways, so you'll know whats going on.

It is entirely possible you will have to modify your modem cable or replace it; even if you have a modem/comm/ program combination that works, it may not have all the lines needed to operate a bulletin board. The extra lines are Carrier Detect (CD) and Data Terminal Ready (DTR). These two are not normally needed by most communications programs; they are however needed by Bulletin Boards, since they need to be able to detect incoming calls and to force them to disconnect.

USING A STANDARD CABLE:

You can change the RS232 pin that Fido uses to determine the state of the Carrier Detect signal. This is done with the /V switch; see the section on switches, and the section on Modems, next.

No /V switch at all is the same as entering:

```
          16/V      default
```

If you have a DC Hayes Smartmodem, connected to an IBM Async Card via a one-to-one cable, then you should run Fido with no /V, or possibly:

```
             128/V         or        (RLSD)
              32/V                   (DSR)
```

When Fido is first run, and there is no caller connected, Fido will say:

(1) Waiting for a Call or Event

When a connection is made, within 2 seconds, you should see:

(2) Incoming call
Determining Baud Rate

If Fido does not sit there and wait for a carrier (1), but immediately says (2), then you have selected the wrong /V bit. If it says (1) like it's supposed to, but never acknowledges a connection, then you have it wrong also.

Legal values for /V are: 1,2,4,8,16,32,64,128.

Custom Cabling:

The idea here is to connect the cable such that you can not only communicate to the modem, but also get to the Carrier Detect signal from it. If you already have MINITEL, PC-TALK, etc working with the modem, thats a good start, but not enough. The Carrier Detect line is not needed by programs such as MINITEL and PC-TALK; Fido needs it to tell when a caller has made a connection, or when they have hung up. Just because PC-TALK or MINITEL work with the modem doesnt mean that Fido will.

The following are the pins that must be wired to an IBM Async Card. You will have to determine how to connect other computers using this as an example.

IBM PC	SMARTMODEM	USE
1	1	Ground,
2 Rx Data	2	Data, modem to PC
3 Tx Data	3	Data, PC to modem,
6 Carrier Detect	6	Carrier Detect
20 Data Terminal Ready	20	Cause Disconnect

The other pins are not needed by Fido, though may be needed by other comm. programs, and in any case can be left in.

Carrier Detect: CD

CD is generated by the modem, and tells the computer (Fido) that a call came in, and a connection made. Conversly, Fido can tell when a caller hangs up or loses connection, and knows to log that user off.

Data Terminal Ready: DTR

Fido uses this to disconnect users that have gone over their limit, and to prevent incoming calls while Fido is performing internal processing that may take a few minutes.

DTR is also of extreme importance for FidoNet for placing outgoing calls. Fido will not run properly if DTR is not connected and working properly.

Commanding the Modem

There are three things that control how the modem is used:

n/J	Modem Type
n/V	Bit Flag, for testing Carrier Detect
FIDOMDM.BBS	Command file for the modem

Between these three, Fido can accomodate almost any modem type. This section covers most popular modems, with specific settings for each of the above.

FIDOMDM.BBS is a special, optional, file that Fido uses to command the modem. If it does not exist, then it is not used. For some modems, it is not needed.

When Fido is first run, any commands in FIDOMDM.BBS are sent line by line, to put the modem into a state where it is useful for Fido and FidoNet. There is no limit to the length or contents of this file; the only thing Fido does is send out each line, and wait up to 3 seconds for a response from the modem. (To avoid sending them out too fast.)

For external Smartmodem type modems, it's usually easier to set the switches to make it autoanswer. For internal modems that do not have switches, you have to do it with FIDOMDM.BBS.

The /J modem type switch tells Fido how to handle the modem. If not specified, it defaults to 1/J, Hayes or compatible. Select other values for other type modems. If you do not set this right, Fido may still work, however, FidoNet will not when it tries to dial to other systems.

Most of this information was compiled by Jim Ryan, sysop of Fido #9.

If you do not see your modem here, try various settings. If you have a comm program such as PCTALK running, all you need to do is get it to autoanswer; see the D. C. Hayes 1200B Internal Modem. That should work for most Hayes type modems.

In General:

For Hayes type modems, the idea is to set the following:

Carrier Detect follows true Carrier
Data Terminal Ready enabled
Autoanswer on the first ring
Numeric Result codes
Turn loudspeaker off always (annoying)

D. C. Hayes Smartmodem 300 and 1200

This is the easiest modem to use; besides being "compatible" (being the one everyone copies) it also performs flawlessly. It's manual is a bit technical and terse, but is complete and bugfree. It performs well both because of it's software and it's high quality analog phone line circuitry.

If you can afford it, dollar for dollar, this is one of the best modem buys. You cannot go wrong with an external Hayes modem.

Modem Type: 1/J
Bit Flag: 16/V

Switch Settings (External Modem)

1 Up	Support DTR Lead
2 Down	Digit Result Codes
3 Down	Result codes sent
5 Up	Autoanswer first ring
6 Up	Supports Carrier Detect lead
8 Down	Recognize commands

Others as needed

FIDOMDM.BBS should contain:

```
ATM0                    turn speaker off
```

See also the D. C. Hayes 1200B Internal Modem for FIDOMDM.BBS commands that can be used instead of some of the switch settings.

D. C. Hayes 1200B Internal Modem

Modem Type: 1/J
Bit Flag: 128/V

Switch settings:

Dont have a 1200B manual, but you need to set the switch that enables Carrier Detect to follow the true Carrier Detect state.

Put the following commands into FIDOMDM.BBS. The items in (parenthesis) are comments; do not include those in the command file!

```
ATS0=1          (select auto answer)
ATV0            (numeric result codes)
ATQ0            (send result codes)
ATM0            (turn off speaker)
```

IBM PC Junior: General Notes

Modem Type: 4/J

Bit Flag: 128/V

You will need a Techmar or other expansion memory, for starters. Also, you need to use the modem command file as desribed in this section under Novation SmartCat.

Qubie Internal Modem

See D. C. Hayes 1200B.

IBM PC and U.S. Robotics Password

```
Modem Type: 1/J     (default)
Bit Flag : 128/V
Switches : All Up
```

You'll need either a female to female RS232 cable or a gender change cable/plug in order to use the US Robotics cable. Your FIDOMDM.BBS should read.....

```
+++
ATH0Z
ATE0M0V0Q1S0=1
```

IBM PC and Apple Modem 1200

```
Modem Type: 1/J
Bit Flag : 128/V
Switches : #1 -- UP
           #2 -- DOWN
           #3 -- UP
```

You'll need either a female to female RS232 cable or a gender change cable/plug in order to use the Apple Modem 1200 cable. Your FIDOMDM.BBS should read.....

```
+++
ATH0Z
ATE0M0V0Q1S0=1
```

IBM PC and Rixon R212A

Modem Type: 1/J (default)
Bit Flag : 128/V
Switches : None

It works OK, but there are still some problems. Your FIDOMDM.BBS should read.....

```
+++
ATB 10001110
ATZ
```

IBM PC and Novation Access 1-2-3

Commanding the Modem

Modem Type: 4/J
Bit Flag : 128\v
Switches : None

WARNING: you can NOT send messages using FidoNet and you WILL have trouble using Xmodem file xfers. Add /X to the command line. Your FIDOMDM.BBS should read.....

```
%F 4
%C 1
%U 0
```

Televideo 1605 PC and XT

Modem Type: 1/J
Bit Flag: 32/V
Switches: 3,5,8 Down
  Others: Up

The FIDOMDM.BBS command file should contain:

```
ATS0=1
```

IBM PC Junior
Novation SmartCat

```
Modem type: 4/J
Bit Flag: 128/V          (PC Junior)
```

In FIDOMDM.BBS, put in the line:

```
UNLISTEN
```

Anchor Automation Signalman

Fido will not work with the Anchor Automation, as it does not support the DTR control line. Fido cannot cause a caller to hang up, nor can it disconnect during FidoNet mail transfers. Since FidoNet would be unable to disconnect after placing a call, with a remote possibility of an extremely high phone bill, the decision has been made not to support this modem.

DEC Rainbow 100 and DEC DF03 Modem

```
Modem Type:  2/J
Bit Flag:   16/V
```

Set the dip switches behind the front panel to accept commands at 1200 baud, and add to the command line (in RUNBBS.BAT):

```
     /R   Initialize at 1200 baud
```

DEC Rainbow and Racal Vadic VA212 Modem

```
Modem Type: 3/J
Bit Flag:  16/V
```

Bizcomp Modem

This so called "Hayes compatible" is one of the poorest modems ever made; it is barely compatible, and unfortunately, cannot be used for Fido, nor many other comm. type programs.

It will work OK at 1200 baud, but not at 300. For some unknown reason, the modem transmits all characters at 1200 baud, even when connected at 300. While it sounds like it simplifies things, by not having to switch baud rates, it also means that you must "throttle" data sent to it; you cannot just send data at 1200 baud to the modem, when it is leaving the modem at only 300 baud.

This has so many basic flaws that no attempt will be made at making Fido accomodate it.

Otrona Attache 8:16 and US Robotics Modem

Modem Type: 1/J
Bit Flag: 16/V (default)
Switch Settings:

- RNG: Off
- 8HI: On
- BAL: Off
- DTR: On

Cable connections:

- pins 2,3,4,7 & 20 - straight thru
- pin 8 on modem to pin 5 on Otrona

Prentice POPCOM Modem

Modem Type: 1/J
Bit Flag:

Except for S register 18, this can be treated the same as Hayes Smartmodems.

In FIDOMDM.BBS:

```
ATV0E0Q0
ATS18=0
```

Modem Problem Determination

SYMPTOM:

Fido says "Waiting for a call or event" but will not answer the phone.

You must tell the modem to autoanswer, either by a switch setting (if applicable) or via a command in FIDOMDM.BBS. See the Modem installation section.

SYMPTOM:

Fido says "Waiting for a call or event", immediately followed by "Incoming call, determining baud rate" when there is no incoming call.

You have the modem installed incorrectly. The problem is that Fido thinks that there is Carrier Detected when there isnt. You probably have the /V value wrong. If the one reccomended above does not work, try the others. There are eight possible values; 1,2,4,8,16,32,64,128.

For Hayes type modems, there may be a switch to cause CD to be "faked". This switch must be set for true carrier detect.

A third possibility is that you have a "non standard" cable, and pin 6 or 8 is not wired. If that is the case, you will have to repair or replace the cable.

SYMPTOM:

Fido rings the bell continuously, displaying the message "CANNOT DISCONNECT THE MODEM!".

Fido requires that it be able to cause a disconnect by using the DTR line. The Anchor Automation modem does not support DTR, and therefore cannot be used with Fido.

For modems with options switches, it may be merely a switch setting to enable the use of the DTR line.

This can also be caused by a modem cable that does not have all the connections made.

If you see this message, you can type Control-C to abort it. You should also make sure the modem is disconnected until you can repair this situation.

SYMPTOM:

Fido detects the wrong baud rate.
Fido gets (or gives) garbage to the modem

For Hayes type modems, you probably have WORD RESULT

CODES set, and Fido thinks that this is being typed by someone who just called in. See the modem section on commanding your modem to give numeric result codes.

A sure symptom of this is if you see a user name of CONNECT when you look at the screen.

This can also happen with poor modem designs, such as the Bizcomp modem.

Changing Command Privelege Levels

To further customize Fido, you can change the privelege level needed to operate each of the commands. This is done with the 3 command (covered in detail in *Fido's Operating Manual*).

Entering 3 by itself lists the current settings. Each command area has it's own privelege tables:

Main Section,	MAINPRIV.BBS
Message Sections,	MSGPRIV.BBS
Mail Sections,	MAILPRIV.BBS
File Sections,	FILEPRIV.BBS
Read Command Prompt	READPRIV.BBS
CHANGE Command Prompt	CHGPRIV.BBS

For example, you can change the Raw-Display command in the file areas to be NORMAL, and let everyone use it, or you can make the EnterMessage command in the Message areas PRIVEL, so that new callers (first time callers) cannot enter messages until you up their privelege level with the SYSOP.EXE program. (They can always leave Messages to the Sysop at Goodbye.)

Briefly, to change a command privelege, enter:

```
3 <x> <priv.>
```

Where:

<x>   Is the command letter, as entered normally to Fido. (E for Enter message, K for Kill, etc.)

<priv.> Is one of the Fido privelege levels below. You MUST spell it out fully.

TWIT	Lowest
DISGRACE	
NORMAL	
PRIVEL	
EXTRA	
SYSOP	Highest

Hints and Tips

There are a number of things you can do that make Fido faster. If you have enough memory for a memory disk, put the messages in it. This will speed things up by a factor of ten or so. Dont worry about the .BBS files or .HLP files; they are small and infrequently accessed. The downloadable files are generally too large, and again, areent accessed in a way that they speed up from a memory disk.

Also, you should use a large number of FILES and BUFFERS in CONFIG.SYS. This is probably the limiting speed factor in most systems without memory disks. Use the maximum FILES = 20, (each "file" takes only 36 bytes, so dont worry about that) and lots of buffers. The following is an example:

```
FILES = 20
BUFFERS = 50
```

More than 70 buffers or so starts to actually slow the system down, as DOS spends all of its time fiddling around with the buffers.

If you run out of memory, then lower the number of buffers. Buffers are 512 (1/2 K) bytes each on IBM PC type machines.

IMPORTANT NOTE:

MSDOS is not too smart about checking for enough memory when you use BUFFERS = . The actual buffers arent allocated until they are needed; if at that time there isnt enough memory, then you get the message:

```
        "Insufficient Memory"
```

And the system halts, making you reboot. Make sure you have enough memory. 40 to 50 buffers is plenty, and will work fine with 256K of memory.

Use GOOD diskettes. If you get a disk error, the user will just see the system hung, and the "... error" "Retry Ignore Abort" message on the local console. Not good.

Initially, use the RUNBBS.BAT batch file from the distribution set of files. Modify it to include any switches needed. Later on, you can include other things for system maintenance, using the scheduler. See the scheduler section for that.

When Fido terminates to DOS, it returns an ERRORLEVEL, that can be detected from within RUNBBS.BAT. Some of the ERRORLEVEL values are built into Fido; others can be set by the sysop. The fixed ones are:

1	Control-C from console or stack overflow
2	Fido Disk File error
3	Other fatal errors

If there is some fatal error, you'd probably like to see it, and in any case there's not any sense in running again. Also, you have to type Control-C on the local console to terminate Fido (for maintainance, etc) and this causes it to terminate the batch file.

The Fido software is not prone to crashing. Most troubles will be hardware ones, or power failures and brownouts. The system cannot be crashed by callers hanging up at random points, and there is no way in which debuggers can be run, etc.

WARNING:

Beware of other programs such as NEWKEY, pop up alarm clocks, SIDE KICK, automatic screen blank programs, and anything like them. None of them are compatible with any program that makes extensive use of interrupts, such as Fido.

Most of the programs do illegal things with the hardware, and will bomb Fido, and other systems. You may never have problems; probably you will, but it may take hours or days to show up. One symptom is that the system just "stops". If in doubt, start the system up again; if it happens again, reboot again, this time leaving out any such program, and see if that cures it. It may take several days to discover.

It is recommended that you do not use such programs while running Fido.

Files Section Organization

There are a number of ways you can run your Files Section, if you have one at all. You can of course combine the features any way you want, but here are some common ones:

OPTION 1: FREE ACCESS UPLOAD AND DOWNLOAD

Users can upload and download files at will. New files are uploaded into the same area and added to the files list, so that they are available immediately.

OPTION 2: FREE ACCESS DOWNLOAD, UPLOAD BY SUBMISSION

Users can download any file, but newly uploaded files cannot be downloaded until the sysop allows it. There are two ways to do this.

These are accomplished with three things:

File Download Path	In the list of areas
File Upload Path	In the list of areas
FILES.BBS	The files list

The upload and download paths are accessed by the "1" command, as mentioned before. FILES.BBS is a special file in each file area, that contains the list of available files.

Before describing the Options above, here is some information:

FILES.BBS is the list in each Files area of the files available for downloading. You must create the first one, but it can be automatically updated in a number of ways later.

FILES.BBS is basically just a text file, that contains the names and a description. Fido puts in the file size, and aligns the columns neatly. For instance:

(Sample Files display)

```
AFILE.EXE      4545 Just a sample file.
```

```
PROG.BIN  323432434 A very large file
FOO.FIZ     MISSING Another sample
```

(The actual FILES.BBS contents)

```
AFILE.EXE Just a sample file.
PROG.BIN A very large file
FOO.FIZ Another sample.
```

You must enter the names as shown; the DOS DIR format (FILE   EXT) will not work, it must be (FILE.EXT).

Note that if a file is really missing (not in the directory) the file size will say MISSING instead.

There are special lines within FILES.BBS, that control how it is listed by the various commands. The special lines are determined by the first character. They are:

```
- This is a comment only line
 So is this (starts with a space)
@ This is the "end" of the files list.
```

The first two are merely comments; you can use them for whatever you wish. The third is a special marker, and tells Fido to not list anything after this line, including this line.

When a file is uploaded, Fido asks for a one line description for that file. The filename and description is added to the END of the files list, above.

If there is a line beginning with @, then the new files are added after it. Since nothing is listed beyond this point, these files are not listed, and are esentially unavailable until the sysop edits FILES.BBS and deletes the line containing the @, or moves it past the last file in the list. This lets the sysop check out uploaded files before making them available.

Uploading and Downloading

As mentioned before, Fido has an upload and a download path. The "F" command lists the files in the download path, using the FILES.BBS list there.

Uploaded files go into the upload path, and their names and descriptions are added to the FILES.BBS list in the upload path.

If the upload and download paths are the SAME, then Fido will not let an uploaded file replace (or destroy ...) an already existing downloadable file. This prevents good files from accidentally being destroyed or changed.

If the upload and download paths are DIFFERENT, then Fido will let existing files be overwritten. Usually this means there is a seperate upload area, not accessible to users.

You can use the above characteristics to your advantage. Instead of trying to describe them all, here are three examples.

OPTION 1: FREE ACCESS UPLOAD AND DOWNLOAD

Users can upload and download files at will. New files are uploaded into the same area and added to the files list, so that they are available immediately.

Make the upload and download paths the same. New files go into the download area. Make sure there is no @ line in FILES.BBS; when a new file is uploaded, it's name and description is added to the end of the files list, where it is immediately available for downloading.

OPTION 2: FREE ACCESS DOWNLOAD, UPLOAD BY SUBMISSION

Users can download any file, but newly uploaded files cannot be downloaded until the sysop allows it. There are two ways to do this.

VARIATION A:

As in Option 1, make the paths the same, but put a line beginning with @ in FILES.BBS. When a file is uploaded, it is added to the download directory, but it's name and description will not be displayed with the "F" command.

This is the easiest to maintain way to do things; to make new files available, all you have to do is delete the line beginning with @, or move it to the end of the list.

The disadvantage is that the file really is accessible; it just does not show. If a user knows it's there, it can be downloaded.

VARIATION B:

Make the upload and download paths different, where the upload path is a disk or directory not available as a File Area. It doesnt matter if there is an @ line in FILES.BBS.

Since new files are uploaded into a totally seperate area, they cannot be accessed by users. (You can put the path in a separate file area, with SYSOP privelege level, so that you can get in there to look.)

The disadvantage is that in order to make these files available, you must use the DOS COPY command, and update the FILES.BBS in the download area manually.

This is the most secure method, is is reccomended only you require absolute security on uploaded files.

Command Line Switches and Setting Time Limits

There are many command line options that control Fido operation. You will have to edit these into the batchfile that runs Fido. The switches are:

<n>/J Selects the type of modem you are using. <n> is a number that specifies the modem type:

1/J	Hayes
2/J	DEC DF03
3/J	Racal Vadic VA212
4/J	Novation/ IBM PC Junior

See the section on modems for details.

/T Test mode. Described early in this manual, not used for a remote Fido system. Not a good thing to put into the batch file.

/B Not used anymore.

/M Private message system. Please read the chapter on private systems.

/P Set the default privelege level for new users to DISGRACE. See the chapter on semiprivate systems for details.

(number)/S   Set the time limit for logging on. This limit starts as soon as carrier is detected, and stops after a valid password is entered. At that point, the normal user or first time user time limit starts. The minimum is 5 minutes; the default is 10.

(number)/L   Set the time limit per call to (number), where (number) is the number of minutes. 0 means unlimited use, i.e. no time limit. Must be 0 - 65535. Read a later section on time limits. If not specified, then the limit defaults to 60 minutes.

(number)/F   Set the time limit for new users. The default here is 60 minutes also. You can give first time callers a low limit.

(number)/D   Set the total time limit for a 24 hour period. This is usually 4 or 5 times the /L limit. This is the maximum number of minutes a caller is allowed in a 24 hour period, and prevents long term hogging, and keeps them from hanging up and calling right back when the per-connect limit is reached. The default is 300 minutes.

(number)/K   Limit the number of K bytes a caller can download in a 24 hr. period. Along with /F, /L, and /D, allows full control over call-in-once-download-everything and-never-call-back type users.

<number>/V   Define the bit used to determine the state of the Carrier Detect signal from the modem. The default is CTS, or 16. See the modem section in this manual for details.

/U   Disables the file transfer information displayed when a file upload or download is happening. Normally, Fido will display status messages about each block being transferred, and all errors that happen. This is very useful when a caller is having trouble. Enter /U to disable this display.

/?   Debug feature enable. The use of this varies from revision to revision. Frequently, it enables new functions that are not fully debugged, or may not be liked by users or sysops.

/R   Initialize the modem (with the command file FIDOMDM.BBS, if present) at 1200 baud. If /R is not present, it is done at 300 baud. Used mostly for DF03 modems.

<n>/G   Sets the number of attempts with connect for FidoNet. The default is one. This is the number of times that Fido/FidoNet will call a particular number, if the modems connect but no mail is transferred.

/I   Disable the local console. This is pretty much useless for most installations. Usually, Fido displays what a user sees, on the local console. /I disables this; it can also be turned on and off with the local console commands ^O and ^N. This is used mostly with multitasking programs such as XAP or Multilink.

These switches are entered thusly:

```
Fido 90/l 30/f 200/k 180/d
```

The above sets:

90 minute time limit per call (normal users)
30 minute time limit (first time callers)
200K bytes/48 hrs download limit
180 minutes time limit per user per 48 hrs

/1
/2

Select which physical IO device to use. The default is /1, COM1 on the IBM PC. /2 Selects COM2. Not used on machines other than the IBM PC, etc yet.

HELP Files

Fido has a number of HELP files, that are displayed by entering "?" at almost any point within Fido. There is a help file for each place ? can be entered. The HELP path is used to find the file needed. The standard Fido setup is a single subdirectory called HELP, which contains all the help files.

All the help files have the extention .HLP without exception. These are:

MAIN.HLP	Main Section command prompt
C.HLP	Within the C command, Main Section
MSG.HLP	Message Section command prompt
MQUERY.HLP	Within the R command, Message, Mail Section

ENTERCMD.HLP	Within the E command, Message Section
MAIL.HLP	FidoNet Mail Area command prompt
FILES.HLP	File Section command prompt
XFERTYPE.HLP	Within the D and U commands, File Section

You can customize these if necessary, but at least initially don't worry about them.

Maintenaince and Backup

Once the system is up and running, you will need to look at messages, put files in the download display, etc. This should be done daily or so.

TIMELOG.BBS and *.TLG FILES:

These record the number of calls to your Fido. Please read the section on the TimeLog.

SYSOP.LOG

This is a record of callers and any messages they leave at the Goodbye command. Also in here go error messages (missing files, disk full, etc) These can be printed or whatever, then deleted. Fido will create a new one if it does not exist, otherwise it adds new info to the end. This is the most useful way to watch system activity.

MAILER.LOG

This is a record of all FidoNet activity.

USER.BBS

This is your list of users. It grows with every new caller, and the SYSOP utility program can be used to maintain it.

NEW FILES

If you use a separate upload directory, look in it once in a while. Copy desireable files into a download directory, and update FILES.BBS accordingly. Deleting them afterwards makes it easy to keep track of uploaded files.

MESSAGES

Log on to the system (via test mode or calling in) and read and prune as necessary. SYSOP level users can read all messages, private or not, and can delete anyones messages.

Fido's Internal Files

This is a brief accounting of some of the internal only files that Fido can create. These are managed automatically;

you do not need to worry about them. They are included here for the curious.

NODELIST.SYS

This file is created every time that NODELIST.BBS has changed since the last time Fido was run, or if NODELIST.SYS does not exist.

ROUTE.BBS

This file controls where and how FidoNet mail is sent. It is described in the FidoNet section.

MAIL.SYS

Created and maintained by the 4 (sysop only) command, this is where FidoNet related info is kept. If you delete this, you will need to reenter the node number and mail paths.

MSGPRIV.SYS
MAILPRIV.SYS
MAINPRIV.SYS
FILEPRIV.SYS
READPRIV.BBS
CHGPRIV.BBS

Each of these contain the names of all the normal user commands, and the privelege levels necessary to access them. One or more of these files may not exist; they are created only if a command privelege level is ever changed.

Deleting these files causes the command priveleges to revert to their default settings.

*.IN
*.OUT
*.FLO
*.FLI

These are working files created and usually deleted by FidoNet. You will never see these files, except when you reset or reboot in the middle of FidoNet operation.

The .IN files are packets received from other FidoNet systems; they contain messages which are unpacked. The .OUT files are packets sent to other remote FidoNet systems. .FLO files are the list of files to be attached to that node; it may be empty. .FLI is the list of files received from a remote system, and may be empty or non-existent.

Fido's Internal Structures

These are the data structures used by Fido for various things. Each is described seperately. This information is included for informational purposes.

The structures are fairly stable; if additions are made, they are always made in a backwards compatible fashion, usually with a version flag or some other method to indicate the format change. If you write utilities to work with these structures, you will almost always be able to accomodate future changes.

All text strings: null terminated, left justified, lower case, first letter upper case. Only one space between words.

Integers are all 16 bit twos complement, except double word (32 bit) integers as noted.

Flags are implemented as bit masks in a 16 bit unsigned integer. All are positive true.

Flag type items (tab expansion on/off, "More?" on/off, etc) are all full integer flags; Fido keeps them as 1 == on, 0 == off, though any non-zero value means on. Please keep them as 1 == on for consistency.

This is the format for each record within the users record. There is no explicit start or end of file markers. This is a very stable structure.

```
struct  _usr {
     char name[36];       /* users ASCII name, */
     char city[36];       /* city and state, */
     int date[20];        /* initial date, */
     char pwd[16];        /* what else, */
     int times;           /* # times called, */
     int help;            /* last help setting, */
     int tabs;            /* 1 == expand tabs, */
     int nulls;           /* number of nulls */
     int msg;             /* last message area, */
     int more;            /* last MORE setting, */
     int priv;            /* user privelege level, */
     char ldate[20];      /* last time called, */
     int time;            /* time on sys in 1 day */
     unsigned flag;       /* various bit flags, */
     unsigned upld;       /* total K byte uploaded, */
     unsigned dnld;       /* total K bytes downldd, */
     unsigned dnldl;      /* download limiting, */
     int files;           /* last file area, */
     char width;          /* screen width, */
     char len;            /* screen length, */
     int credit;          /* credit, in cents, */
     int debit;           /* debit, in cents, */
};
```

```
/* User privelege levels */

#define TWIT -2           /* total XXXXXXX */
#define DISGRACE 0        /* disgraced user, */
#define NORMAL 2          /* normal user, */
#define PRIVEL 4          /* priveleged user, */
#define EXTRA 6           /* extra priveleges, */
#define SYSOP 10          /* SYSOP proveleges, */

/* User help levels */

#define EXPERT 2          /* expert */
#define REGULAR 4         /* experienced */
#define NOVICE 6          /* new user */

/* Structure for the time log: first access,
last access, and an integer counter for each hour
of each day of the week. */

struct _tlog {
     int calls;           /* # calls in log, */
     char fdate[20];      /* logging started, */
     char ldate[20];      /* logging ended, */
     int log[7][24];      /* the counters, */
};

/* Structure for each virtual bulletin board.
(SYSTEM<n>.BBS) */

struct _sys {
     int caller;          /* caller number, */
     int priv;            /* min. privelege */
     char msgpath[40];    /* path for message base, */
     char bbspath[40];    /* path for .BBS files, */
     char hlppath[40];    /* path for HLP files, */
     char uppath[40];     /* path for uploads, */
     char filepath[40];   /* path for file area, */
     int attrib;          /* attributes */
};

/* System file attributes */

#define SYSMAIL 1         /* is a mail area */

/* Tables of command names and privelege levels for each. */

struct _cmd {
     char name[20];
     int priv;
};
```

```
/* Message header structure. The message text is
just a long string following the structure. */

struct _msg {
```

```
     char from[36];      /* who from, */
     char to[36];        /* who to, */
     char subj[72];      /* message subject, */
     char date[20];      /* creation date, */
     int times;          /* number of times read, */
     int dest;           /* destination node, */
     int orig;           /* originating node */
     int cost;           /* actual cost this msg */
     int caca[6];        /* extra space, */
     unsigned reply;     /* thread to previous msg */
     int attr;           /* message type, below */
     int up;             /* thread to next msg */
};

#define MSGPRIVATE 1     /* private message, */
#define MSGBROAD 2       /* broad cast message, */
#define MSGREAD 4        /* read by addressee */
#define MSGSENT 8        /* sent OK (remote) */
#define MSGFILE 16       /* file attached to msg */
#define MSGFWD 32        /* being/was forwarded */
#define MSGORPHAN 64     /* unknown dest node */

/* Scheduler Structures and Definitions (SCHED.BBS) */

/* Scheduled events, tag definitions. */

#define MIN_EVENT 'A'         /* minimum FidoNet tag */
#define MAX_EVENT 'W'         /* maximum */
#define EXT_EVENT 'X'         /* return to DOS event */
#define YELL_EVENT 'Y'        /* Yell command enable */
#define DTR_EVENT 'Z'         /* return to DOS DTR high */

#define SCHEDS (5 * DAYS_WK)  /* size of time table */

/* Standard time of day structure. */

struct _time {
     int year,month,day,daywk;
     int hour,mins,sec;

/* (daywk == 7 == every day. Month, day, year,
sec not used by sched) */
};

/* Scheduler time structure. */
```

```
struct _sched {
     struct _time time;
     int len;
     int enable;         /* 1 == enabled -1 ==
                            disabled 0 == deleted */
     int trigger;        /* not used (yet) */
     int result;         /* returned value */
     char tag;           /* schedule tag */
     int a,b,c,d,e;      /* extra garbage */
} sched[SCHEDS];
```

```
/* Message packet header. */

#define PKTVER 1         /* current version */

struct _hdr {
     int orig;           /* originating Node # */
     int dest;           /* destination node */

/* This should be converted to the new time struct, but
oh well, someday */
     int year,month,day,hour,minute,second;
     int rate;           /* baud rate */
     int ver;            /* packet version */
     int extra[19];
} pkthdr;

/* FidoNet Mail definitions. */

/* MAIL.SYS file structure */

struct _mail {
int node;           /* local node number, */
float fudge;        /* cost fudge factor, */
int rate;           /* baud rate */
char mailpath[80];  /* path to find mail in */
char filepath[80];  /* mail file path */
} mail;

/* Node descriptor. This is created from the routing
list at run time. This is the in memory version,
and also whats written out in NODELIST.SYS. */

struct _node {
int number;         /* node number, */
int cost;           /* cost per minute to call */
int rate;           /* baud rate */
char sched;         /* schedule tag */
char name[14];      /* node name */
char phone[40];     /* phone number */
char city[40];      /* city and state, */
} node;
```

Fido's Users Manual: Main Section

Edited by Ben Baker

FidoBBS is a Bulletin Board system that takes full advantage of MSDOS version 2. Most of the features were modeled after the usual CBBS and RBBS software, since the command system is so popular and familiar.

There are three help levels; the default for new callers is NOVICE. This provides a small menu of commands, and usually gives further description as commands are executed.

REGULAR is the command letter prompts, minus the full words, and does not tell you what command you have just executed. Useful for after you understand the structure, but have not memorized all the command mnemonics.

EXPERT is just a very short command prompt; there are no menus or other things; prompts are very small and fast. Once you are very familiar with Fido, this will make using it very fast.

Regardless of help level, you can enter ? at almost any point for a list of the options at that point.

From the Main Section, all of Fido's available commands can be accessed. The actual commands are described in detail below. Fido is arranged into three sections:

```
        +-----------------------+ Change User name, etc
        |                       | Bulletins, Questionaire,
        |     Main Section      | Yell at Sysop, access to
        |                       | other Sections.
        +-----------------------+
        |                       |
        |                       |
        |                       |
+------------------+   +--------------------+
|                  |   |                    |
| Message Section  |   |    File Section    |
|                  |   |                    |
+------------------+   +--------------------+
Read, enter, delete     List, download,
messages, message       upload files, change
areas, etc.             file areas, etc.
```

Commands

All commands in all areas are one letter, though some require further arguments or options. Most command allow entering the further arguments right after the command key; if not, or if not supplied, each necessary component is prompted for, usually with help available at each point.

For instance, to download a file you must enter the D command, the download method (one of seven) then the filename. If you entered D, then Fido would ask for the download method. After that, it asks for the filename. Once you become familiar with it, you can enter it all as:

```
D X FILENAME.EXE
```

All on one line, skipping the prompts. This works for most commands.

All commands can be aborted with a Q or a blank line.

All Sections

The following are commands that are available in all Sections.

G -- GoodBye

Logoff the system, optionally leave a private message to the sysop. If a message is left, it goes into the special message area (in system area 0) that is accessable only by the sysop. The usual message editor is available for comments.

S -- Statistics

This command lists the time on the system, time remaining, etc. Also, depending on the area, it will list further information.

Files Area

Statistics also lists the free space left on the default drive.

Message Area

Optionally lists messages to or from you.

FidoNet Area

As in Message areas, and also lists FidoNet statistics.

Main Menu

This is the section you are in after signon, right after the Bulletins and quotations. From here, the "system" type commands are accessed, such as Change user settings, etc. The commands accessible from here are described below.

M	Message Section
F	File Section
Y	Yell at the Sysop
A	Answer a Questionaire
B	Get Bulletin again
E	Get the Editorial
C	Change User settings
S	Statistics
U	Display User List
G	Goodbye, logoff

M -- Message-Area

Enter the message section. From there, the message areas can be accessed.

F -- Files-Area

Similar to Message-Area, but where the upload and download commands are.

A -- Answer-Questionaire

If there is a questionaire in the system, you get to fill one out with this command. A questionaire is an text pseudo-language file, described in the Questionaire section.

C -- Change User

A small sub-menu of commands to change various user settings, such as signon name, password, screen dimensions, etc.

U	Users name
C	City and State
P	Password
W	Screen Width
L	Screen Length
H	Help level
M	"More?" on and off
T	Tabs On and Off
N	Nulls, 0 - 20

You can change your name, password, city and state, and anything else at any time. (Unless the options were removed by the sysop.)

NOTE: For security reasons, when your name is changed, the entire users list is searched to make sure the name is unique. This is to prevent having two or more people with the same name.

The screen dimensions are used by the page pauser ("More?") and the message system. Messages are listed to fit within the set screen boundaries, and message entry is done via a word wrap editor. If "More?" is ON, then Fido will pause after each screenful. All of the command prompts are made to fit within this.

Help level is either NOVICE, REGULAR or EXPERT. NOVICE is the default; it causes all prompts to be displayed fully, and each command to be described with a one liner.

REGULAR is similar, but does not list the entire wordy prompt. EXPERT is almost no prompt at all.

"More?" on and off controls whenther Fido asks "More?" every screenful.

Tab expansion converts all tabs sent into spaces, for those systems that cannot handle tabs.

Fido can be set to send nulls after each linefeed. It defaults to 0 nulls.

B -- Bulletin

This just displays the signon bulletin, BULLETIN.BBS again.

E -- Editorial

Similar to the Bulletin, (but displaying file EDTORIAL.BBS) but not displayed automatically.

Y -- Yell at the Sysop

By yelling, the caller can attempt to contact the sysop. It beeps for 30 seconds, and if the sysop does not respond by then, displays "Sysop not available".

The sysop can chat with the user at any time.

U -- Users List

Lists the names of all current callers to the system. Not very useful.

G -- Goodbye

Terminates the call. It is not really necessary; however, callers seem to get upset when they cant figure out how to log off, so there it is. (Its OK to just hang up.) It also allows entering a private message to the sysop. After (optionally) entering a message, it causes a disconnect, exactly what would have happened if the caller just hung up.

Fido's Users Manual: Message and Mail Section

The Message Section is accessed by the M command from the Main Menu. All message commands are available here. Return to the Main Menu is via the M command.

M	Return to Main Section	G	Goodbye
A	Area-Change	R	Read messages
E	Enter a Message	K	Kill a message
L	List Messages	S	Statistics

I   Search for an item

A -- Area-Change

Change the current message area. If an area number is entered after the A, that area is selected if available. If not, a list of areas is displayed, from which you may choose one.

If a new area is selected, then the messages there are counted (why it says "Wait ...") and if it's the first time you've selected that area, asks if you want to list mail addressed to you. Nothing happens if you select the same area, or blank line or Q.

The mail search searches all messages, newest to oldest, and lists: New message to you (ones you haven't read yet) Old messages to you (ones you have read) and messages you have entered. Typing Control-C or Control-K aborts the search. If there are lots of messages in any of the above catagories, it reminds you to Kill some.

The message area numbers displayed may not be sequential. Each area has an associated privelege level and if the your privelege is not high enough, that area will not be displayed nor available.

L -- List Messages

Only the message headers are listed. A starting number is requested; this can be:

An ordinary number
A number followed by + or -
- itself.

If just a number is entered, messages are listed starting at that one, towards the highest message.

A number followed by + or - controls the direction messages are listed in. + is redundant. - means list backwards; 100- then means start listing at 100, list towards 1.

- means list from the highest message towards the lowest. This is the most useful, and for some reason the least used. If you have called before, this is the easiest

way to see any new messages.

R -- Read

Actually a small menu by itself. From here messages can be read, entered, killed, replied to, etc. See the descriptions of the Kill and Enter commands for details on those.

Entering a message number reads that message. N (Next) reads the next highest message, and P (Previous) reads the next lowest message. Hitting CR repeats the previous N or P; if neither was used, then it defaults to P.

* is a special message number. It means the highest message number.

+ and - are little used keys, but one of the most powerful message features. Messages can have, and be, replies. (See R command below.) Messages created with R are linked to other messages. For example, a message about an item for sale may have a reply. If you come across the original for sale message, it will say "See also #nnn, use + key" if it has a reply. Entering a + at this point will take you directly to that message. This message will say "Reply to #xxx, use - key". Entering - will take you back to the original message.

There is no limit to the number of messages that can be linked this way. Further replies are added to the end, and if one in the middle (or either end) is Killed, the string of messages is "patched" to maintain continuity.

This is my favorite feature (dreamed up by John Madill) and I am slightly annoyed that no one uses it.

R Replies to the current message. It acts basically like Entering a message, but automatically fills in the "To:" field, and links it into the original message.

You can enter "RK" to reply to a message, and kill it afterwards. This is extremely useful while reading your mail.

If you are replying to a message in the Mail area, it also fills in the destination system. The message will be mailed out that night.

Please read about Entering messages, further below.

E enters a message in the normal fashion.

E -- Enter a Message

You fill in the To:, From:, Subject: fields, etc, then the main body of the message. After terminating the message, commands can be used to save, abort, edit, etc.

If the area is a FidoNet Mail area, then in order to enter a message, you must select the system to send it to at the prompt. If you do not have enough credit, then Fido will not let you send a message to that system. If you do, then Fido will deduct the cost of the message from your credit.

Message text entry is done with a word wrap editor. If the screen width is set properly (C command in Main Menu) then this is about as close to a full screen editor as you can get through a modem (on current DOSs anyways.)

Like WordStar, text is entered continuously without carriage returns. Words that would go past the left edge are wrapped to the next line. Carriage returns can be used to form paragraphs as in WordStar. When entered this way, later reading of a message will conform to the callers screen width.

At the Enter Message command prompt, the options are:

C for continue. Add more to the message, adding to the end.

L lists the message so far.
T lets you change the To: field.
J lets you change the subject.
D Deletes a line
I Inserts a line

E edit a line. You enter the line number, then you are prompted for the Old string (what to change) then the New string (what to change it to). Both old string and new string can be anything that you can enter, except a carriage return.

NOTE: it is possible to edit a line such that it goes beyond the edge of the screen. It will list correctly after it is saved.

A aborts the message. You will be asked yes or no.

S saves the message. The message is written out to disk, and the highest message number, etc is updated.

K -- Kill

Deletes the message. Will not let you delete messages that are not To: you or From: you, unless you are the sysop. If the message you killed was mail to a remote system and it was not sent yet, you get your credit back; you are not charged for that message.

I -- Index

Terrible name, but too bad. Prompts you for a string. It then searches all messages, newest to oldest for the thing you enter. It looks in the "To:", "From:", or "Subject:" fields. If the string you enter is in any one of those parts of the message, the header of the message will be listed.

Typing Control-C aborts the search. An insensitive sliding match is done; ON matches TONY or PITON, etc. Each message header matching is listed, then a list of all matching message numbers.

? will match any character. For example, if there are the following messages in the system:

#100 From: Joe Bungpop To: All Subject: Musical Instruments

#102 From: No Body To: W.S. Burroughs Subject: Guns

#200 From: All To: Board Subject: Music

If you enter MUS, it will list messages 100 and 200, as MUS is contained in the Subject: fields of both. If you had entered BU, then it would list messages 100 and 102, as it would match "Bungpop" and Burroughs". Entering B??R would cause #102 and #200 to be listed, as B??R matches BURRoughs and BOARd.

Fido's Users Manual: File Section

The file Section contain all the file oriented commands for downloading, uploading, etc. The M command returns to the Main Menu.

M	Main Section
G	Goodbye
A	Area-Change
F	List Files
L	Locate specific Files
D	Download
U	Upload
R	Raw-List
K	Kill a file
T	Type a file

A -- Area-Change

Similar to the Message Area-Change. If a valid area number is entered, it is selected directly, otherwise a list is displayed.

Like the message areas, there may be numbers missing from the sequence. These are either blank paths or high privelege areas.

L -- Locate

Given a file specifier (wildcards reccommended) it searches through all file directories for the filename, and lists all occurences of it. L;*.ASM lists all .ASM files in all areas, etc. L;*.* lists all files everywhere.

This is not the general purpose list-files command.

F --- Files

Lists the files in the current directory. If no specifier is given, it lists all files, otherwise only matching ones. The size of the file and the description is displayed after the filename.

If "MISSING" is displayed instead of the file size, then the file is not there, even though it is listed.

T -- Type

Very simple. Displays an ASCII file on the screen. It statistically checks the file to make sure it is a text file.

U -- Upload

Upload a file or files to Fido. I will not even attempt to document all the upload methods. Suffice to say you select the upload method and do it.

The total uploads made is kept in the user record. These count towards further downloads, by effectively raising the download limit on a 1 for 1 basis. It will not give you credit however. There is no limit on uploads.

After all files are uploaded, you are asked to describe each one; this is added to the list of files. (You may not be able to see the newly added files though.)

D -- Download

Again, pick the method and do it. Before the download starts, Fido checks time limits and K byte limits, and prevents too long downloads. Uploading will remove some of the download limit.

Fido's Users Manual: Control Characters

Control characters are used by Fido for editing command lines, pausing and aborting commands. Control- C and Control- K always are identical and have the same effect:

Control-C Aborts anything
Control-K Aborts anything

Control-S pauses the display. Any key except a second Control-S will let it continue.

When not using the word wrap message editor, Fido allows simple editing of command lines. It assumes the your or computer terminal can support at least backspace. It supports:

^C, ^X, ^U, ^Y

Delete the entire line.

^R

Restore the previous line, from last use or deletion. (Most useful.)

^S, ^G, ^H, DEL

Delete one character to the left.

^D

Restore one character.

ESC, CR, LF

Enter the line.

Sysop Only Commands

The sysop commands are all single digits, and do not show up in any prompts nor in any of the supplied help files. Most are "dangerous", in the sense that they set and change Fidos environment, and are not for general use.

With one exception, all sysop commands are available at all prompts. This exception is at the "Read Command" prompt; entering numbers specifies a message number, not a command! Therefore, the 3 command (see below) is accessed only here as "#".

1 -- Paths

Sets paths, select and create new system files (SYSTEMn.BBS) and change the privelege level for each area.

Entering 1 by itself lists the contents of the current system file. Changing and setting system file parameters is done by entering them after the 1, in a slightly peculiar way. The 1 commands are:

(a)	1 <number>	Select or create a system file.
(b)	1 <pathname>\M	Set the MSG Path
(c)	1 <pathname>\F	Set the FILE Path
(d)	1 <pathname>\U	Set the UPLOAD Path
(e)	1 <pathname>\H	Set the HELP Path

(g)	1 <privelege>\V	Set the Area Privelege level
(h)	1 <number>\A	Set Attributes
(i)	1 S	Save the changes
(j)	1 ?	Get HELP! With the 1 command

If (a) is not done, then the system file is the one last selected when in a Message Section or File Section. You can always find out by just entering 1 by itself; it lists which system file and all its contents.

To create a new system file, enter 1 <number>, where <number> is one higher than the highest one. Fido will not allow creating SYSTEM99.BBS, for instance, when the current highest is only 8. When a new one is created, it copies the contents of the current one into the new one.

For (b) through (e), the specified path is selected. You can check your changes by entering 1 by itself.

(g) sets the privelege level for the Message and File areas in this system file. The default is DISGRACE. (This is not copied from the previous one; it is always set to DISGRACE.) To change it, enter as shown. The privelege levels are:

TWIT
DISGRACE
NORMAL
PRIVEL

EXTRA
SYSOP

You must spell fully the privelege level name.

Each system file has an attribute word. Currently, only one bit of it is used, so the method of changing it is crude. The 1 bit determines whether the MSG Path is a regular Message area or a FidoNet Mail area.

1 0\A	Set normal Message Area
1 1\A	Set Fidonet Area

This should be set on only one MSG Path; the one you use for Fidonet mail.

Any changes you make are not permanent until you save them with (i). If you make changes, then select another area with (a), then the changes are lost.

3 -- Command Privelege Levels

# -- READ COMMAND ONLY

The 3 command is used to change and list the privelege levels for each command in each Section. 3 by itself lists the command names and the privelege level needed to be able to use it. If the users privelege level isn't high enough, then the command will not show in the prompt.

To change a privelege level, you enter:

```
3 <cmd> <privelege>
```

Where <cmd> is the command letter as normally entered (for instance, in a Message Section, K for Kill message, etc) and <privelege> is the name of the privelege level, as listed under the 1 command. The change is stored immediately.

The following are considered separate areas, as far as the 3 command goes:

Main Section,	MAINPRIV.BBS
Message Sections,	MSGPRIV.BBS
Mail Sections,	MAILPRIV.BBS
File Sections,	FILEPRIV.BBS
Read Command Prompt	READPRIV.BBS
CHANGE Command Prompt	CHGPRIV.BBS

Though FidoNet message sections are almost identical to regular Message Sections, it has it's own set of privelege levels for the commands.

4 -- Fidonet Parameters

The 4 command sets and displays FidoNet parameters. These parameters are used mainly by FidoNet.

1 <number>\N	Set Node Number
1 <pathname>\P	Set Mail Path
1 <pathname>\F	Set Mail File Path

The Mail Path is the message directory you want to use for the FidoNet mail area. It must be the same as one of the message paths set with the 1 command, and the one that has the MAIL attribute set with "1 1\A".

The Mail File Path is where files received via FidoNet are placed. Generally a good place is the same as where ou put uploaded files, or a seperate directory only for mail files.

The node number is the way that your Fido system is identified to others on the Net. These numbers must be unique, and known to everyone else in the Net. If you bought the Fido package, then your node number is on the cover of the FidoNet manual. Otherwise, contact Tom Jennings at Fido #1 to get one.

5 -- List scheduled events

Lists all of the scheduled events that Fido can execute. Events are added, changed, enabled, disabled, etc with the 6 and 7 commands. The default, with a new Fido version 9, is one event enabling the Yell command.

The actual scheduler operation is covered in the Scheduler section of Fido's FidoNet Manual. Basically, these events are things like when to execute FidoNet, days and times when the Yell command is allowed, and with additional work, when external programs are to be executed.

6 -- Enter/change events

This command allows entering and changing events. There is room for up to 35 events. You are asked for an event number, then you are asked for each of the components that make up an event. These are:

Event # [0 - 35]:

Which event to work on. Events can be in any order; Fido executes them according to the day and time. If you enter a number that is already used, you will replace it with the new one you are entering. Hitting RETURN will exit the 6 command.

Day of week [All, 0=Sun, 1=Mon ...]:

The day of the week that this event should be run on.

Entering A (for ALL) means run it every day. For example, FidoNet gets run every day; enter A. An alternative (just for an example) would be to enter seven events, one for each day of the week. This is just for convenience.

Start Time: [hh:mm]:

The time this event is to start. Enter as 10:44, etc, or use a space instead of the colon. Entering just 10, for instance, is the same as entering 10:00. 00:00 (or just 0) is midnight; 23:59 is one minute til midnight.

Window width, Minutes:

How long the specified event should be run. This can be from 1 to 1440, the number of minutes in a day. If you had entered A for the day of the week, 10:00 for the start time, and 60 for window width, then the event would be run from 10:00AM to 11:00AM.

For external events, you should enter one minute; Fido cannot "call back" an external program when it's time is over, it is up to that program (and your batch file) to return to Fido.

Schedule Tag [A - W, X, Y, Z]:

This tells Fido what event it is you are running. A - W are FidoNet events; Fido's FidoNet Manual describes this in detail. Tag A is the normal, nation wide FidoNet time slot. If in doubt, or for new systems, enter A. This will then be compatible with older FidoNets.

Y is the yell command. The Yell command (which rings the bell annoyingly on the console) can be enabled for any time of the day, by setting it up as an "event". Fido will not terminate or do anything different; it will just allow the yell command to be used while the event is running. The default is to allow the yell command from 10:00AM til 10:00PM.

Tag X and tag Z are external events; with these, you can run other DOS programs through a specially modified batch file. X terminates with DTR low (the modem hung up, and auto-answer disabled) and Z leaves the modem connected, so that you can run programs that "talk" to the modem.

ERRORLEVEL [4 - 255]:

Asked ONLY if Tag X or Z is entered, above. This is the ERRORLEVEL code passed back to DOS, where it can be detected by the batch file, for selectively running programs.

7 -- Enable/disable/remove an event

This is used to enable, disable and remove set events.

You are asked for an event number (as entered above) and then what to do with it. If you enter an event number that does not exist, it will tell you so.

Enable "turns on" that event, that is, allows it to run. Disable causes it to be ignored, though it will still exist, so you can enable it later if you want. Remove deletes it entirely.

8 -- Renumber Messages

The 8 command performs the same function as the RENUM.EXE utility program, but from within Fido. This will renumber all the messages in the current message area, so that they start at 1, and increase by 1. It will preserve Replies and See Alsos.

It has the limitation that it will not renumber if there are more than 200 messages; either Kill some messages, or use RENUM.EXE, which will handle up to 1000 messages per area.

Renumbering causes temporary great upheaval to the messages themselves; interrupting the 8 command (by powering off or resetting) is not reccomended. To prevent losing the connection (and disconnecting) from aborting the renum, nothing is displayed while the command is working.

0 -- Terminate to DOS

This is a highly specialized command, that works only when a /W value is specified on the command line.

When set, it causes a immediate termination to DOS with the ERRORLEVEL specified by the /W switch. It is assumed that your RUNBBS batch file will trap this errorlevel, and do something appropriate.

Note that it is NOT adequate to just use CTTY to redirect the console; you must have a special "watchdog" program to monitor the Carrier Detect line from the modem, and reboot the system if carrier is lost. Ohterwise, is the connection is lost, the next person that calls in has control of your system, from the DOS level.

Special Local Console Features

There are a number of commands that can be used from the system's local console while a user is on. This is sometimes called "spy mode", as the user does not have any indication that you are spying on them. The commands are:

?

Displays the name of the current user, and how long he has been on. It lists a small menu of other things that can be done, described here.

C

Chat with the caller. (Also see the Yell command.) The user will get an appropriate message, and you will then be in terminal mode with the caller, allowing you to type back and forth. The chat mode is terminated by typing Control-Z, at which point the user is back in normal Fido operation.

Z

Clear the current users limits. This clears the accumulated download, time on system, and accumulated time on system. (same as user calling back in 48 hours.) This is useful for extending someones time limits temporarily.

^A (Control-A)

Simultaneous keyboards. All keystrokes typed are treated as if they were from the callers modem. Terminated by ^Z, Control-Z, below. This can be used for intensive handholding of terrified users.

^Z (Control-Z)

Terminates both Chat mode (or the Yell command) and simultaneous keyboards.

^X (Control-X)

Immediately terminate a call. The user is logged off. Ignored during file transfers. To be used sparingly. If you need to get into the system while a caller is on, refer to ^T.

^O (Control-O)
^N (Control-N)

These two are for special purposes only. They turn off (^O) and on (^N) the local screen output. Normally, Fido displays what the user sees, on the local screen.

For use with a multi-DOS type program (XAP, Multilink, etc) this console activity greatly slows down the system. It can be disabled with ^O, or via the /I command line switch. These two keys override and /I command line switch.

^T (Control-T)

When entered while a user is on, after the user hangs up it causes Fido to disable the modem, and alert the operator that the system is free. Fido rings the bell once a second for five minutes.

The sysop types ^C to quit to DOS, or ^Z to stop the terminate and resume normal operation. If there is no response in five minutes, then Fido resumes normal operation automatically.

Space bar

When a user has started a Yell, hitting the space bar immediately causes the "sysop not available" message.

The scheduler can be used to limit the hours in which the Yell command can be used.

Questionaires

A questionaire is a pseudo-language file that is interpreted by Fido. If someone invokes the A (Answer Questionaire) command in the Main Section, and there is no questionaire, it says:

"No questionaire today"

Questionaires are simple text files that are interpreted by Fido. There is a very simple line oriented pseudo language that interprets these scripts. There are only a couple of "commands", and only two crude conditionals.

The answers to the questionaire are put in a seperate file, without the questions. They are also normal ASCII text, but in a fixed field format with field characters, to make post- processing of the questionaire easier. (Some day I'll actually write the processor!)

The questionaire consists of two files:

QUESTION.BBS	The questionaire script
ANSWERS.BBS	The answers

Answers are appended to the end of the answer file; it is create first if it does not exist.

There is also a second questionaire: QNOPWD.BBS. If this exists, users who forgot their password, or if the /M (private system) is specified, they get to fill it in. You can put in requests for passwords, address and phone numbers, etc here.

The first character of each line tells Fido what to do with the line. This first character is a normal printable character. If the first character is not one of the known commands, the line is merely displayed.

NOTE: Do not have blank lines in the questionaire. Interrogation will stop at end of file or a blank line.

The questionaire commands are:

/<prompt string>

Display <prompt string>, then input a single line of text. Maximum line length is 78 characters. The usual editing is available. The inputted text is written to the answer file.

*

Put users info to the answer file: name and last (current) signon time and date. Useful for the "what kind of system ..." type questionaires. Leaving this out allows confidential questionaires, as there is no other record of which caller filled on in.

+N <prompt string>

Multiple choice. The single digit number N is the number of possible choices. (1 - 9) If an illegal number or the number is missing, the number of choices defaults to one, so at least it wont hang up. The choice entered by the user is checked for in range 1 to N, and saved in the answer file.

There is a special feature of multiple choice: an internal flag is set if the last choice (choice N) is selected, that can be tested by the ? and ! commands, below.

? <prompt string>

To be used ONLY immediately after a multiple choice question. This command will be executed only if choice N is made. This allows implementing "A, B, C, Other" type questions. Put "other" as the last choice; if selected, then the ? command following it will be executed. See the example below. If any other selection was made, this line will be skipped.

The conditional flag is cleared by: + command, ? command, _ command. No other commands affect it.

!

To be used ONLY after a multiple choice question, similar to ? above. If the last choice of a multiple choice was selected, this terminates the questionaire. Allows asking "Do you want to fill this out (1) yes (2) no".

_

(Underscore) Unconditionally clear the conditional flag.

Questionaire Example

The following is an example of an executable questionaire, though the questions are useless.

```
This is not a command, and will merely be displayed. This
can be used as a header or prompt for long questions.
* Insert user info (text after * is ignored)
Want to fill this out:
+2 (1) Yes (2) No
!
/Prompt, input a line of text:
+3Multiple choice: (1) green (2) blue (3) other:
?What other color?
?will never be executed; ? above cleared flag.
+2Multiple choice: (1) this (2) that
This is just plain text. Does not affect the flag.
?What is that?
Text can be used as a prompt for a long question, like this:
What time did you last jump out the window?
/
```

Example Answer File

The answer file is a specially formatted text file. It can (has to be for now) interpreted manually. (The answer file for QNOPWD.BBS is ANOPWD.BBS) For the questionaire above, here's what the answer might look like: (manual comments in parens)

```
* Some User 32 Jan 44 23:59:59
  1: A line of text   (text inputted by / command)
  2: 2                (choice 2; ? after it not executed)
  3: 2                (choice 2. == N, so ...)
? 3: the other thing  (conditional question, "other")
  4: at midnight      (ans to "when ... out the window?")
```

Each question is numbered by BBS. Characters are 3 digits max, right justified.

Special Characters in Column 1:

* User record info.

? Answer to conditional question. Note that the number is the same as the question that triggered the conditional question, even if there was text inbetween.

(blank)   Normal answer.

The SYSOP Utility

The SYSOP utility is used to maintain the user list file, USER.BBS. It is a simple data base type program, has online help, and is fairly easy to use.

First, a general description of the user list. The user list is a variable number of records, with one record per user end to end. New users get added to the end. SYSOP allows going back and forth through the user list, changing things such as names, passwords, privelege levels, and more global things like purging old users from the list.

It will be much easier to use this section while actually running the SYSOP program. When first run, SYSOP displays the first record in the list, usually the sysop. You can get a brief description of each of the commands by entering a ?, then the command key listed in the menu.

SYSOP automatically saves any changes you make. To change a password, for instance, use the P command (below) and thats it. SYSOP will save any changes before exiting to DOS or selecting another user record.

Selecting a User Record

There are a number of ways to look at (and maybe change) user records. You can use the following key to move to the next and previous record:

Next Highest	Previous
>	<
+	-
.	,
_	=

The last four don't appear to make any sense, but they happen to share the same keytop, either shifted or unshifted.

B -- Beginning of the User List

B takes you directly to the first record.

Z -- End of the Users List

Z takes you to the end of the list.

A -- Select Record Number

You can also enter the record number directly with the A command. This is handy only if you know the record number.

F -- Find a Record

This works similar to the I command in Fido. You are prompted for a string; SYSOP then searches for that string

in the name portion of the user records, and stops at the first one it finds.

It starts searching from the *next* record; i.e. if you are currently at #100, it starts looking at #101. When it reaches the end, it wraps around, and continues from record #1. It stops if a match is found, or until it reaches the point where it started. Therefore, if no match is found, it leaves you where you were.

If you are searching for "John", you will probably get more than one match. If it matches one you did not want, then just repeat it; it will start searching again with the *next* record. For speed, you can use the Control-R key as in Fido, to recover the text you had entered ("John" in this example, by typing:

F	Find command
^R	Restore old line
CR	Execute command

E -- Enter New User

E adds a new user to the list. First, it goes to the end of the list (may take a while) then makes a clear, empty user record. You use the usual SYSOP commands (below) to fill out the record.

After doing an E, SYSOP will save the new user record only if you change at least one field in it. If you just do an E, then a command such as B or - it will not save the new record. Once you have changed anything (name, password, etc) it will save it automatically.

N -- Name

N prompts you for a new name. You can enter one or two words. Entering a blank line leaves the present name.

P -- Password

Enter a new password.

V -- Privelege Level

Change users privelege level. You can enter only the first letter of the privelege level name, or enter the whole thing. The levels are listed by ?V. They are:

TWIT
DISGRACE
NORMAL
PRIVEL
EXTRA
SYSOP

Q -- Quit to DOS

Saves any changes, and returns to DOS.

O -- Mark Old Users

This is used to mark old users that haven't called in for a long time, to be later purged from the list. (One time callers, etc) The user records are not deleted by this command; you must use the ! command to do it after you have them all marked.

It starts at the current record, and if a caller hasnt called within the specified number of days, that record is marked as "deleted". Users within the number of days are marked as "Un-deleted". You can use this if you change your mind, for instance, to un-delete everyone use the O command to mark all users that havent called in, say, 1000 days. This will undelete anyone that has called in within 1000 days.

A good number to use is 32; you will find that almost all users that get deleted have called only once. SYSOP displays each user record that it is marking.

D -- Mark Current User Record as Deleted

This marks the current record as deleted, as if it were marked by O.

! -- Purge deleted Users

This does the actual deletion from the user list of any marked records. First, it makes a backup copy of USER.BBS in USER.BAK. As it deletes records, it puts the deleted ones into USER.OLD, where they can be recovered from if a mistake is made.

You can recover deleted records by either renaming USER.BAK to USER.BBS, or by appending USER.OLD to USER.BBS by the DOS command:

```
REN USER.BBS USER.WRK    rename for now,
COPY/B USER.WRK+USER.OLD USER.BBS
```

Note that SYSOP will not delete users that have a privelege of EXTRA or above or any credit left.

$ -- Set Credit and Debit

This is used for supplying users with credit for sending FidoNet mail. It prompts you for a dollar value, which is the added to the current credit, after subtracting any debit from it. If there were already $5.16 credit left, entering "10" would leave them with $15.16 credit.

To set the credit to a specific number, enter:

```
=10
```

This clears any remaining credit before adding the $10.

M -- Set Default Message Area

Fido saves the message area that a user was in last.

Normally, new users end up in message area #1. For certain types of systems, you might want to eliminate the A (Area-Change) command, and lock users into particular areas. This determines what area the user will end up in when entering the Message Section.

Note that if the specified area does not exist, or it's privelege level is too high, Fido will automatically bump them into area #1. Make sure you have these set up right.

K -- Set Default File Area

Same as M, except it sets the default File Area.

Aborting a SYSOP Command

Some of SYSOPs commands may take a while, such as searching for a name or marking users. You can abort most commands by hitting the ESCape key. It will stop at the current record.

TIMELOG -- The Time Log Utility

Fido keeps a record of all callers in a simple log that records the number of callers per hour, for every day, for seven days. When it is seven days old, it gets renamed to the current date, and a new one is started.

The TIMELOG.BBS file contains 168 slots: one for each hour of the day, for seven days, plus the date when started and ended, etc. Every time someone calls in and logs in succesfully, (enter their name, and guess their password) Fido increments the appropriate slot. When all seven days are used, it renames it to the current date (TIMELOG.BBS becomes for instance, 01JUL84.TLG, or 21MAY84.TLG, etc) and starts a new TIMELOG.BBS.

These .TLG files are fairly small; 378 bytes each, so you dont have to worry about them filling your disk for quite a few months.

Note that TIMELOG.BBS is not started on Sunday; it starts whenever you first run Fido. If you want to synchronize it to start on a particular day, delete TIMELOG.BBS on that day of the week, and when Fido is run, it will create a new one.

Running TIMELOG.EXE

TIMELOG.EXE will run on any MSDOS machine, and is not hardware sensitive. To run it, enter:

```
        TIMELOG <name>
```

Where <name> is the name of a .TLG file. It will display the contents as crude bar graphs. There are two options:

```
        TIMELOG <name> /P       Pause
        TIMELOG <name> /F       Format
```

/P tells TIMELOG to pause between each graph (so they dont run off the screen). /F formats the output for a printer, putting formfeeds in between each graph. (To send to the printer, use redirection: TIMELOG <name> /F >PRN)

What FidoNet Does

FidoNet is a true electronic mail network, not to be confused with other systems that use the word "net" because it sounds trendy.

FidoNet links all of the 100 or so Fidos across the country, all together once a night (or more), to accomodate automatic message and file transfers.

Simply put, you enter a message on one system, say in San Francisco, and Fidonet later sends that message to the right Fido, which may be in Boston. All of this is totally automatic.

From the users point of view, FidoNet is just another message area. All of the magic is hidden away. As a sysop, you should make an effort to understand it, as it will make it a lot easier to install and use.

The most important point in the original FidoNet design was low cost. Since most Fidos are run privately, having it send *.EXE to every Fido in the country is not too desireable. Therefore, Fido has complete cost control at every step; FidoNet need not cost you anything to operate. You get to control absolutely everything that FidoNet does.

In some areas of the country, there are enough Fidos that they can benefit by having a host system. A host takes care of receiving mail from across the country, and passing it along later to the local systems. Some also have outgoing hosts; mail destined for far away places gets sent to your host (usually a local call) who then passes it on to where ever its going.

Fido keeps logs and "accounts" with each user, and each host and node system. These are used to control costs (a user with no credit cannot send mail) and to make it easier to pay a host for passing you mail across the country.

Costs

Each node maintains a node list, that contains the accessible remote systems. Systems not in the list cannot be mailed to. The sysop controls this file, updating or changing it with any text editor. It can only be read from within Fido; users cannot otherwise access it, and cannot see the actual phone numbers used to dial.

If your Fido is a public "free" one, then probably no toll call systems will be allowed for general use. All users credits default to zero, so that users cannot send mail willy nilly. In order for a user to enter a mail message (to a toll call system) the sysop has to assign them credit with the SYSOP.EXE utility program.

Some local areas have a host that will accept toll call mail for free; contact the Fidos in your area to find out about this.

If a system in the node list is marked as free, then anyone may send messages to it.

Each user record contains a credit and a debit. Messages entered are charged to debit, and cannot exceed the credit. Deleteing a message (that has not been sent yet) entered lowers the debit by the cost of the message; the actual cost of the message is stored within the message. Onec a message is sent, the cost cannot be recovered by deleteing it.

Installation of FidoNet

Assuming you already have Fido running, setting up FidoNet is simple. If you dont have Fido running, then skip this part until you do.

The basic steps to do are:

(1) Set the basic mail parameters shared by Fido and FidoNet.

(2) Set up the MAIL message and file area.

(3) Make up a working node list (list of other systems.)

(4) Set the scheduler to run FidoNet.

(5) If you have a host system, make a route list.

(6) Obtain a node number.

That is all there is to it. Each step is described below. Also, I can't emphasize how important it is to ...

HAVE THE TIME AND DATE SET ACCURATELY!!!!!!!!

Fidonet depends heavily on the correct time and date. Within 5 minutes or so is fine, its not that critical, but since this is a time driven system, it is very important.

First, create the message area that you want to use for mail, or use an existing one. Pick any name or place you want. This is identical to making a regular message area; see the step by step installation section for details. Set the SYSTEMn.BBS system file, the DIR.BBS descriptions, etc as described there.

Also, create a file area for files received from the net. This can be your upload area, or a unique mail file only area. It does not need to be a public area, accessible by the File Seciotn commands, though it can be.

Now, you have to tell Fido that this is a mail area, instead of an ordinary message area. This is done with the 1 command.

```
     1 <number>     Select your mail area,
     1 1\A          Set MAIL attribute
     1 S            Save the changes
```

Now Fido knows that this area is a mail area; the extended features that support Fidonet operation are now enabled when in that area.

Setting MAIL.SYS

Next, you have to set MAIL.SYS, the file used by both Fido and FidoNet. It tells Fido what your node number is, and tells FidoNet where the mail message area is. (This is a bit redundant, but this file is used to connect Fido and FidoNet together.)

Using the 4 command (See *Fido's Installation Manual* for details) set the mail path to be the same as the message path you set for mail use above:

```
     4 <pathname>\P      Set MAIL Path
```

This is the path that Fidonet uses to get and put messages.

```
     4 <pathname>\F      Set Mail Files Path
```

this is where any files received are to go. It should NOT be your default Fido area, as you might get files of the same name, wrecking ones you have. Best to make it a seperate, new area, or use an upload area.

If you have a node number, set it now. If you do not, PLEASE DONT set a fake number. You do not need a number to send mail.

```
     4 <number>\N        Set Node Number
```

TO OBTAIN A NODE NUMBER:

Enter a Mail message, and send it to Fido #51 via FidoNet. This way, they will know that you are really up and running. Remember to include:

Systems name,
Sysops name,
Phone number,
Hours of operation,
City and state,

Any other important details.

FidoNet Glossary

First, a small glossary of terms used in this manual to describe FidoNet and all it's permutations.

Network: A group of two or more Fidos, that send mail to/from each other at a specified time.

Node: One Fido, running FidoNet. It may be a host as well.

Host: A node that handles mail for itself as well as other nodes. It may accept mail, and send it on to another node.

Local Net: A group of nodes, presumably within a small geographical area, (ie. non-toll call or cheap calls) that consists one or more nodes and a host. Basically, a small, self contained Network.

Local Node: Really the same as just "Node", but implies that it is a node that has a host serving it.

Topography: Basically, if you were to draw a map showing all the nodes and their interconections. Some of the "shapes" made by the map are called "topographies" if they show some recognizable pattern, such as star, ring, etc.

Traffic: The actual messages being sent. If you picture messages as things that travel along the phone connection, it makes more sense.

FidoNet Operation

Fido and FidoNet are both contained in one program, run as specified by the scheduler. Fidonet is a time driven system, and the national time slot is at 1:00 AM Pacific Time. (4:00 AM eastern Time) During normal Fido operation, users can enter messages, and during the FidoNet time, these messages are made into packets and sent to the right destination. The messages may be destined to any one or more of the available remote nodes in the nodelist.

At the predetermined time, Fidonet takes control. It then (if enabled) creates mail packets, one per node, containing the messages for each node. If there is no mail to a node, no packet is created, and no call is made to that system.

After the outgoing packets are made, FidoNet alternately waits for calls and attempts to place calls. Mail packet transfers are done on a collision detection basis. After the first few collisions, the network synchronizes. If there are a number of nodes to send mail to, each one is called in turn, until all are sent, or mail time is over. If it fails with one node, it goes on to the next, and repeats the failed one only after trying all of the others first.

In between outgoing calls (if any) FidoNet delays a random interval, during which it waits for incoming calls. This interval, along with the redial algorithm, synchronizes the net after the initial collisions.

If an incoming call is detected, it attempts connection with it. The baud rate is determined (same as a normal caller would in Fido) and a message to human callers is displayed (warning them that it is accepting only other FidoNodes) and a synchronization process is started. This process must complete within 60 seconds, or the call is terminated. Once synchronized, the packet transfer is made. The receiver just stores that packet for later use, and then disconnects.

Whenever an incoming call is received, FidoNet calls out immediately afterwards (assuming there are calls to be made) since there is a high probability that the line is now clear. This helps synchronize the network.

To place an outgoing call, the sender dials the number, performs the sync process mentioned above, and transfers it's outgoing packet. (Messages to a given node are again checked against the node list at mail time; if they do not match, the packet is not sent, and an error is logged.) If there are any files to be sent, they are sent after hte packet. If the transfer was successful, the destination node number is deleted from the senders list of nodes to call.

The collision detect algorithm is optimized such that

during the first few minutes of mail time, there are many collisions, after which the net synchronizes, and none or few collisions occur.

When mail time is over, FidoNet deletes all it's outgoing packets that were assembled, and for each one that was sent sucessfully, marks those messages (in the mail area) as SENT, so the originator can tell if they went out or not. Then, the incoming packets are unassembled, and the messages placed sequentially in the mail area. These packets are then deleted. Any files received are marked in the FILES.BBS file list in the mail file area.

If any mail at all was sent, the user credits are balanced. This is somewhat unsatisfactory, as it balances the accounts even if the mail was not sent. This is to prevent extremely long processing time necessary to account for each message and user. (Users lists run upwards of 600 entries typically; on a floppy based system this would become unworkable.)

Fidonet then terminates, and if there were no fatal errors invokes Fido for another day. Messages received are then accessible like any other message.

Using the Scheduler

The scheduler is the way that Fido controls when, as opposed to how, FidoNet (and other things) are run. The details of operating the scheduler commands is described in the Sysop Only Commands section. This section tells how to use the events.

The scheduler contains a list of "events", each of which consists of a day and time and other information on what to do at that time. While Fido is waiting for a caller, it is also watching the scheduler.

When Fido accepts a caller, it watches the time until the next scheduled event; if it is less than that callers time limit, it sets their limit to be the time until the event, and warns the user that they have less time than normal. Fido will enforce this limit to ensure that the caller logs off so that the event can be run.

The scheduler keeps information on what is to be done to execute the event. The schedule Tag is what determines what kind of event it might be.

Tags A to W are reserved as FidoNet schedules; when an event tag A to W is found, FidoNet is run. This tag controls which nodes in the list mail is sent to; this allows sending mail to certain other systems during certain times, by setting a schedule with each desired tag.

This example covers making Fido run FidoNet in the national time slot. The time of day depends on your location, that is your time zone. The example below assumes Pacific Time; adjust accordingly.

Pacific Time	Central Time	Eastern Time
01:00 AM	03:00 AM	04:00 AM
60 minutes	60 minutes	60 minutes

Do two things: Set Fido to run FidoNet every day at 1:00AM for one hour (normal FidoNet operation) and allow the Yell command to be used only between the hours of 9:00AM and 5:00PM.

```
(Enter command '6')
Event # [0 - 34]: 0            (over write the default one)
Day of week [All, 0=Sun ...]: A (every day)
Start Time [hh:mm]: 9          (9AM)
Window Width: 480              (8 hrs)
Schedule Tag: [A - W, X, Y, Z] Y (Yell command)
(Yell command now enabled from 9:00 AM to 5:00 PM)

(Enter command '6')
Event # [0 - 34]: 1            (New event)
```

```
Day of Week [All, ...] A       (every day)
Start time [hh:mm]: 1:00       (at 1 AM)
Window Width: 60               (1 hour)
Schedule Tag [A - W, X, Y, Z]: A (Normal FidoNet)
```

That is it. It was not necessary to choose events 0 and 1; any two could have been used. Using 0, however, replaced the default yell command enable (10:00 AM til 10:00 PM) with the desired one.

To add other events, just choose another event number and enter them as above. There is room for up to 35 events.

External Events

The scheduler can be used to cause Fido to terminate to DOS (to the RUNBBS.BAT batch file, actually) with an ERRORLEVEL value that can be used to run programs at certain times of the day.

This isnt a general purpose way to run programs like data based programs, etc. It is meant for use with programs that can run unattended, such as Fido's RENUM.EXE utility program.

You set a schedule normally, except select X for the tag. When you do this, it asks an additional question:

```
ERRORLEVEL [4 - 255]:
```

This is the ERRORLEVEL passed back to DOS. This value can be tested for in the batch file by:

```
        IF ERRORLEVEL (xx) GOTO LABEL
```

Where (xx) is the errorlevel you specify, and LABEL is a label in the batch file.

You need to understand how MSDOS batch files work; it is too complex a subject to cover here. One reminder though: the IF ERRORLEVEL (xx) GOTO LABEL does not mean "if errorlevel EQUALS (xx) goto label", it means "if errorlevel IS EQUAL TO OR GREATER THAN (xx) goto label". This means that you must put the tests in decreasing order:

```
        FIDO_XXX switches ...
        if errorlevel 10 goto x1
        if errorlevel 4 goto x2
        if errorlevel 1 goto end
```

Routing: Message Forwarding

Routing is used when it is necessary to send mail for one system to another one. The system which mail is diverted to is called a host, and usually mail for a number of systems is sent to a single host.

Routing is very useful when there are a number of Fidos in a given area; if you had a message to each of four systems in a single city, say, St. Louis, without routing it would mean four seperate phone calls. With routing, FidoNet sends all the messages in a single packet, in one call. The host worries about unpacking the messages and sending them along.

The phone company bills by the minute. At 1200 baud, Fidonet sends about 6000 to 7000 characters a minute; most messages are about 500 bytes. So, to send a 500 byte message, you are paying the full minute; the rest of the minute is wasted. Since hosts send many messages at once, the call takes longer, but is doubly cheaper, as usually all the messages fit into one, or possibly two minutes, whereas if each system called directly, each call would be one minute.

Of course, you either have to find a rich, generous host, or use the host account feature and pool your resources. (Later on that.)

All this is accomplished by the router and the scheduler. The scheduler controls when to run FidoNet (ie. when to send and receive mail) and the router controls where to send the mail.

As an example, here is a very simple system, consisting of four nodes, two of which are acting as hosts.

```
             (Host)                        (Host)
             Fido 1 ------------------- Fido 3
               |                          |
               |                          |
             Fido 2                     Fido 4
```

The idea is to make Fido 2 send all its mail for Fidos 3 and 4 to Fido 1 instead. (In real life, there are usually many other Fidos doing the same thing.) Instead of all systems calling each other all at once, it concentrates many messages into a very few calls.

The route list tells Fido how to send the mail. If there is no route list, then the mail goes out the simple way, ie. directly.

The route list is a list of alternatives. For instance:

ROUTE-TO 1

```
        3,4
```

Says to route to Fido #1, mail for Fidos 3 and 4. (This by the way is the exact syntax for the route list, ROUTE.BBS.)

Fido #1 in the example above needs to send this mail onward to Fido 3. If you are observant (or paranoid) you may wonder what prevents Fido #1 from forwarding ALL mail sent to it, as when another Fido (not shown here) accidentally (or on purpose ...) sends its mail to #1, hoping it will make the long distance calls ...

Unless explicitly stated, no Fido will forward mail NOT ENTERED DIRECTLY ON IT. This prevents the above from happening. There are two route list commands used to specify that it is OK to forward mail from some nodes, to certain others:

```
FORWARD-TO 3

ACCEPT-FROM 2
```

This tells Fido #1 (when in 1's route list) that it is OK to send mail from other systems onward to Fido 3. It also says it is OK to accept mail from Fido #2.

To sum this up, Fido #1 will send its own mail to Fido #3 (obiously) and will also accept mail from #2, and pass it on to #3. It wont allow #2 to send mail to say, #100, or any other not specified.

The last leg of the journey for a message, #3 to #4, may be different. Usually, its not a toll call from a host to a local. Its generally OK for a host to allow forwarding ALL mail to a local, since it doesnt cost anthing. (Besides, this is the function of a host ...) #3's route list must look like:

```
ACCEPT-FROM 1,2
FORWARD-TO 4
```

From this example, you can figure out what it would look like in real life. For example, the ACCEPT-FROM for a host (such as #3) would have to contain all nodes in the system, since the host takes in mail from the outside world, and passes it to its locals.

Of course, something was left out in the above example ... you may have noticed that a message has to go through three mailings to get to its destination. This requires that FidoNet be run three times, and Fidonet gets run only once?

For local host systems, FidoNet is actually run three times per night; the standard one hour national time, and a shorter one before and after it. During the before and after schedules messages travel between hosts and locals; in the national time they travel from host to host. (Or host to system, or system to host, etc, for all the host nonhost combinations.)

This is done with the scheduler, the sysop only 5, 6, and 7 commands.

Each time slot (called a schedule here) is unique. The schedule tag is a one character "name" for each time slot. The national window is schedule A, and runs from 1:00 AM (PST) til 2:00 AM. The two before and after schedules can be anything but A: for instance, B and C. Here is a non existent example:

```
Schedule B:     00:30 AM -> 00:59 AM local to host
Schedule A:     01:00 AM -> 02:00 AM national
Schedule C:     02:30 AM -> 03:00 AM local to host
```

In the example above, we want to have Fido 1 serve the locals (and vice versa) during B and C, and not any of the national ones, since its outside the national time. All of them run A, the national time, so that they will get mail from systems that dont know about routing, and because its simpler anyways.

This information goes into the route list too.

```
SCHEDULE B 2
SCHEDULE C 2
```

It is not necessary to specify schedule A; by default, all systems are A unless specified otherwise.

Route List (ROUTE.BBS) Syntax

The previous examples covered all of what the route list does; here it is again, in a reference type format:

ROUTE.BBS, the routing list for Fido, is a plain old text file, which you can create with almost any editor. It is very forgiving on it's format; it lets you use upper or lower case, blank lines, tabs, spaces, comments, etc.

There are four KEYWORDS. Keywords are followed by lists of numbers, seperated by spaces, commas, tabs, or blank lines. For instance, the following are all OK:

```
ROUTE-TO 10 22 51 16 17 4 65
Route-To 10
        22,16,17,4,65
route-TO 10
22
16
17
4
65
ROUTE-TO 10     22, 51, 16, 17, 4, 65
```

Take your pick as to which you prefer.

ROUTE-TO (host) n1, n2, n3, n4 ... nn

Send all mail for n1 ... nn to (host) instead. This is the basic routing information. This just specifies how mail may be sent; ACCEPT-FROM and FORWARD-TO, below, control IF it will be sent.

If you are not a host, then this is all you need to have in your route list.

ACCEPT-FROM n1, n2, n3, n4, ... nn

Accept mail from nodes n1 ... nn for further mailing. If you were an outgoing host, you should have all of your local nodes listed here; this means that you will accept mail from them, for further forwarding. See FORWARD-TO, below also. Mail from Fidos not in this list will not be accepted for forwarding.

This applies only to hosts; it has no effect on messages entered on your system.

FORWARD-TO n1, n2, n3, n4 ... nn

Your system is allowed to forward only to the nodes specified here. This says it is OK to send mail onward to these nodes, ie. if you were a host, that mail for your

local nodes (n1 ... nn) should be forwarded.

If you are not a host, then you do not need to have this.

SCHEDULE (tag) n1, n2, n3, n4 ... nn

Send mail to nodes n1 ... nn during the specified schedule. This is needed only if you are a host or a local to a host. If not specified, all nodes are assumed to be tag A.

The Node List

The node list is the list of remote systems, the phone numbers, etc. When you apply for a permanent node number, you will receive the latest node list. Node lists will be broadcast to all FidoNodes as it is updated.

NOTE: A slight oversight on my part: you must have your own node in the node list. Otherwise, you cannot leave a regular message in the mail section! Fidonet knows better than to try to call your own system. Messages that are To: the node it was entered on do not get mailed.

The node list, called NODELIST.BBS, describes each remote system, one per line. You do not need to create one; new ones are distributed on a regular basis, and they are easy to find if you lose one.

```
1  0  1200  Fido's_Board  1-(415)-864-1418  SF_CA

^  ^   ^    ^             ^                 ^
|  |   |    |             |                 |
|  |   |    |             |                 |
|  |   |    |             |                 +--- City
|  |   |    |             +--------------------- Phone
|  |   |    +----------------------------------- Name
|  |   +---------------------------------------- Baud Rate
|  +-------------------------------------------- Cost
+----------------------------------------------- Node
```

Each of these items are seperated by one or more spaces or tabs. Do NOT have spaces, commas or tabs in the middle of any of these.

Node Number

This is that systems node number. This identifies the system mail is for (and from); it is buried in the mail packets, and if wrong, Fidonet will not accept that mail packet. It cannot be zero.

Cost

This is the cost, in cents per message, that you want to charge a user for sending mail to this node. If the cost is 0, it is assumed to be a local call. If 0, then anyone may send messages to this node. Otherwise (if not zero) the user must have enough credit in their user record, or Fido will not allow them to enter mail. (Set by the SYSOP.EXE Utility program.)

Baud Rate

This is the speed to dial this system with when it is time to send them mail. It is any number that is a multiple of 300; i.e. 300, 600, 1200 ... 9600. For dialing purposes, 1200 and up are dialed at 1200 baud, everything else is

dialed at 300. Set 300 if your modem cannot handle 1200.

Name

This is just the descriptive name (mnemonic) for this system. Luckily, most systems have some cutesy name, like Fido's Board, Demon, etc. It must be one word. (Use hyphens or underlines.) It must also be less than 14 characters long total.

Phone

This is the phone number for the system. It is basically a standard D.C. Hayes SmartModem phone number, with one exception. If you use Sprint, etc, you must put it in each number. Do not leave spaces in the number. The legal characters in a phone number are:

1,2,3,4,5,6,7,8,9,0 # *	
-	ignored
( )	ignored
.	1 second delay

Note that comma is not the delay character; it is used to seperate items in lists. A dot is translated to a comma when dialing. In addition:

T	Touch Tone
P	Pulse

May be inserted at any point in the number as needed. Fidonet dials in Touch Tone by default, if P is not specified. The maximum length of the phone number is 39 characters.

City

This is another single word field, usually the city and state.