

31 Internet tools

These days the hottest topic in the land of BBSs is connecting a BBS to the Internet. The previous chapters on UUCP (see Chapter 30 [UUCP], page 113) and FidoNet (see Section 29.1 [FidoNet], page 105) have described the support for file, mail and news transfer offered by these systems. Unfortunately the type of connection that is implemented by either UUCP or FidoNet is not real time. This is where the Internet comes in. When a BBS is connected to the Internet a number of services become accesible on a real time basis, including such things as:

- email transfer, as soon as the message is entered onto the BBS it can be sent to its destination,
- news transfer, like email news (SIGs) can be propagated to other systems much quicker,
- IRC, real time conferences spread among users or computers all over the globe 24 hours a day,
- FTP, file transfer between computers can allow your users to browse and download from other systems around the world without long distance charges,
- WWW, the World Wibe Web, perhaps the most hyped, and certainly fastest growning software in the world. A hypertext based information retrival system that spans the glob and anyone can use and contribute data to.

At the current time¹ IceBBS provides direct support for IRC and FTP through a pair of user accessible doors. I expect to be adding more direct support for Internet over time as well as improving the current doors. It may also be possible to use some of the Internet email and news transfer tools with IceBBS instead of UUCP, but I have not experimented with these yet. Let me know if you try this.

31.1 Connecting to the Internet

In order to make use of any Internet tools you must first connect your BBS machine to the Internet. This is perhaps the biggest hurdle you will face. To do this you must solve three problems:

- find someone to provide you with a connection to the Internet,
- select the software that provides the link between your machine and the Internet,
- configure AmiTCP, the so called *TCP/IP protocol stack*, on your BBS machine.

¹ April 16, 1995.

51 Thanks..... 217

Tool types and Buttons Index 219

Command Index 221

Concept Index..... 223

messages to be exported back to the mailing list. To do this you need to use the ‘outmail’ command, in particular the ‘-mp’ mode. Continuing the example:

```
outmail -i -edhb:lastscan -mp +
-mamailinglist@some.internet.site +
-mslightwave_sig
```

You need to specify the name of the SIG that is being used to hold the mailing list, and the address of the internet account that an individual subscriber would normally send new postings or replies to.

30.3.5.3 Hosting a mailing list

The difference between sharing a mailing list between several users on the same BBS and hosting an entire mailing list¹¹ is not too large. You need to setup ‘inmail’ to look for incoming messages that are directed to the name that the mailing list operates under and to have it place those messages into a SIG. Then you need to use the ‘-mh’ mode of ‘outmail’ along with the ‘-ma’, ‘-ml’ and ‘-ms’ options to get ‘outmail’ to send all the new postings and replies that appear in the SIG to be sent to the list of addresses in the file specified by ‘-ml’. The address list file’s format is quite simple, one email address per line, like this:

```
svermeulen@ragnarok.mtroyal.ab.ca
john_lees@amuc.mtroyal.ab.ca
vermeuls@cuug.ab.ca
stephen_vermeulen@1:134/92
```

and yes, it should handle sending to FidoNet addresses to¹².

¹¹ This mode has not undergone much testing at this point in time, so be careful.

¹² An almost useless feature at the present as ‘inmail’ does not handle receiving mail from a FidoNet address yet, but it’s planned for.

34.2.8	Support Utilities.....	160
34.2.8.1	Trim.....	160
34.2.8.2	Updatesigs.....	160

35	Mail.....	161
-----------	------------------	------------

36	Electronic Mail.....	163
-----------	-----------------------------	------------

36.1	Email directories.....	163
36.2	Between users on the BBS.....	164
36.3	Between BBSs via FidoNet.....	165
36.4	Internet mail.....	166
36.5	Support Commands.....	166
36.5.1	CheckMail.....	166
36.5.2	Email.....	167
36.5.3	Forward.....	169
36.5.4	Send.....	169

37	Monitor.....	171
-----------	---------------------	------------

38	Conferencing.....	173
-----------	--------------------------	------------

38.1	Configuring.....	173
38.2	The Confer Command.....	173
38.3	Potential Problems.....	174

39	News Bulletins.....	175
-----------	----------------------------	------------

39.1	The S:IceBBS.NEWS File.....	175
39.2	News files.....	175

40	The Remote DOS Shell.....	177
-----------	----------------------------------	------------

41	The Doors System.....	179
-----------	------------------------------	------------

41.1	Introduction.....	179
41.2	Icon Tooltypes.....	179
41.3	Theory.....	180
41.4	The RemFile Door.....	181
41.5	The Preview Door.....	182
41.6	Edituser Door.....	183
41.7	CircleMUD Door.....	183
41.8	ARexx Doors.....	183
41.9	Internet FTP.....	183
41.9.1	What is FTP?.....	183
41.9.2	Installation for sysops.....	184
41.10	Internet relay chat.....	184

30.3.2 UUExport

This command is used to extract copies of new postings that have been entered by users of the BBS (either through the Sigs function or via the IceOLR and BatchSIGs function) and place them into an outbound directory within 'UUNews:' for transmission to other USENET sites.

30.3.3 InMail

This command scans the 'UUMail:' directory for any mail that has arrived for users on the BBS.

'Inmail' makes several attempts at finding the account to send the email to, if the mail file's name does not match that of a user's account file on the BBS it then attempts to find the user's account by first hashing the unmodified name, if that fails it replaces any underscores's with spaces and tries again, if that fails it replaces any period's with spaces and tries again.

Some recent additions give 'inmail' the ability to act as a simple email information server and as a simple mailing list.

You tell inmail to look for mail directed to a certain account, and if any is received it sends the contents of a named file to the sender of the mail. With this you can set up a number of basic information files and users of Internet email can get copies by just sending messages to the special account names. On the Internet it is quite common to find the 'info' account name used to get an overview of what other information or services might be available through automatic email.

The *mailing list* mode is an extension of the info server mode. It logs the received email into a particular IceBBS SIG. There the message can be read by all people who have access to that SIG. Note that this mode does not send any automatic reply back to the author, so if one is desired you need to run the 'inmail' command twice, once with the '-i' switch and once with the '-s' switch.

The mailing list mode provided by 'inmail' is useful for cases where a number of users of your BBS subscribe to the same mailing list. This results in a lot of duplicated usage of the email SIG. By using email to place the messages from one subscribed account into a regular SIG you can save disk space and reduce the use of email. Note that you will have to add some additional configuration to the 'outmail' commands (see Section 30.3.5.3 [Outmail], page 117) to allow replies to messages in this SIG to be sent back to the mailing list.

29	FidoNet and the BBS	105
29.1	Introduction	105
29.2	Script Files	105
29.3	Security	105
29.4	Multiple Nets	106
29.5	FidoNet Commands	106
29.5.1	Export	107
29.5.2	Import	108
29.5.3	Fido	109
29.5.4	FidoPost	109
29.5.5	FidoTime	110
29.5.6	SafeWait	110
29.5.7	EMail2Cut	111
29.5.8	Pkt2EMail	111
29.5.9	SEcho	111
30	UUCP and the BBS	113
30.1	Finding a USENET feed	113
30.2	Configuring for UUCP	114
30.3	UUCP Support Commands	114
30.3.1	NewsIn	114
30.3.2	UUExport	115
30.3.3	InMail	115
30.3.4	FwdEmail	116
30.3.5	Outmail	116
30.3.5.1	Info server	116
30.3.5.2	Sharing a mailing list	116
30.3.5.3	Hosting a mailing list	117
31	Internet tools	119
31.1	Connecting to the Internet	119
31.2	Internet real time conferencing	121
31.3	File transfer	121
31.4	Is the net up?	122
32	Finding Files	123
32.1	Find, the sysop interface	123
32.2	MUIFind	124
32.2.1	Installation	124
32.2.2	Usage of MUIFind	125
32.2.2.1	Search by file contents	125
32.2.2.2	Search by file name	126
32.3	FindScan, the first pass	127
32.4	MySort	127

30 UUCP and the BBS

UUCP¹ is an electronic mail-based networking package that you can run on your Amiga to connect into what is loosely termed *USENET*. USENET is roughly akin to FidoNet except it runs across a much larger network of machines and the machines it connects are generally mainframes. The drawbacks to running UUCP are that it requires a fair bit of disk space and that the volume of news traffic can be enormous. Some estimates place the load for a complete feed at about 100Mega bytes per day! The terms USENET, UUCP and Internet are often used somewhat interchangeably.

There is a large and active Amiga newsgroup on USENET, along with a sources and binaries group for the Amiga. If you were to receive a complete feed of these, the daily load would be on the order of several hundred thousand bytes.

30.1 Finding a USENET feed

The only crucial difficulty that you will encounter when you try to set up UUCP on your system will be in obtaining a *feed*². If you cannot find someone to feed you the news groups you are interested in then there is nothing³ you can do about it. Though you could always pay for a feed from one of the so-called *public access sites* like UUNET.

If you are searching for a news feed nothing will beat a few visits to your local Amiga User's Group meetings. Look for someone who has contacts with a local computer (software) company or a nearby college or university. There are also UNIX Users Groups appearing, some of these obtain a feed of their own, and often will pass it on in return for a membership in their group. The HAM Radio community can also be a good place to search for a feed. Additionally, you might check the back issues of the C-Users Journal, or some of the BBS specific publications⁴ or get a copy of the book *Managing UUCP and USENET* published by O'Reilly and Associates. There are now many books on the Internet, and a number of these have lists of *internet service providers* (ISPs) which

15.6	Lines 6 and 7	57
15.7	Line 8	57
15.8	Line 9	57
15.9	Line 10	57
15.10	Line 11	57
15.11	Line 12	58
15.12	Line 13	58
15.13	Line 14	58
15.14	Line 15	58
15.15	Line 16	59
15.16	Line 17 and beyond	59
15.16.1	The first eleven fields	59
15.16.2	Additional fields	60
15.16.3	Example line 17 (Use with TrapDoor)	60
15.16.4	Example line 18 (polling script)	61
15.16.5	Example line 19 (a normal line)	62
16	The S:IceBBS.DIRS File	63
17	The S:IceBBS.HAIL File	65
18	The S:IceBBS.ARBITRATE file	67
19	The S:IceBBS.NOASCII file	69
20	The IceBBS.FIDOCFG file	71
21	The ASCII Menus	73
21.1	Account	73
21.2	Alias	73
21.3	bye	74
21.4	cd	74
21.5	chat	74
21.6	confer	75
21.7	ctrl	75
21.8	describe	75
21.9	dir	76
21.10	display	76
21.11	download	76
21.12	email	77
21.13	feedback	77
21.14	goto	78
21.15	lock	79

¹ The UUCP package I am referring to here is the one most recently worked on by Matt Dillon, the first version I tried with IceBBS was v1.13.

² A feed is a site who is willing to supply you with access to USENET.

³ This is not strictly true, you can use the UUCP software to set up a local mini-usenet between a number of Amiga systems.

⁴ Such as: Boardwatch Magazine or the BBS Callers Digest

29.5.7 EMail2Cut

This program has been replaced by ‘outmail’ which handles the export of both FidoNet and USENET email. Because ‘outmail’ does both tasks it can also do some automatic cross routing between FidoNet and USENET. See Section 30.3.5.3 [Outmail], page 117 for more information.

This program is designed to convert IceBBS email outbound files into FidoNet ‘*.CUT’ files. The ‘*.CUT’ files will be named for the BBS they are going to with a full four dimensional address in the form:

```
Zone.Net.Node.Point.CUT
```

If such a file already exists the software will append the message to that packet file.

By default ‘email2cut’ will not host route any mail, any out of net (or zone) mail (when host routing is not enabled) will be processed into HUT packets. These packets are then held until the system they are destined for calls, or some other process handles them (such as a special script or manual intervention from the sysop).

If host routing is enabled by one of the ‘-z’ switches then some or all of the out of net mail will be sent via other hosts as low priority net mail.

29.5.8 Pkt2EMail

This program is obsolete and should no longer be used, its function is now fully provided by the ‘import’ program.

29.5.9 SEcho

This command can be used as an aide to debugging programs and script files. It acts like the AmigaDOS ‘echo’ command except whatever it echoes goes out the Amiga’s internal serial port. The idea is to use this with other debugging tools which write out the serial port like ‘mungwall’ and ‘enforcer’. The syntax is:

```
secho [-d] "string to print" ["more strings" ...]
```

if you want to know the time at which a string is printed just include the ‘-d’ flag.

Note that if you are using the internal serial port for your BBS you should not use this command as it will interfere with any communications that are occurring on that port.

Table of Contents

1	Conventions	1
2	Introduction	3
2.1	The Demo Version	3
2.2	Features	3
2.3	Custom terminal	4
2.4	Performance	5
2.5	Requirements	5
3	Quick Start	7
3.1	The Distribution Archive	7
3.2	The Install Script	8
3.3	Configuring Your Modem	8
3.4	Commands	9
3.5	Using IceConfig to configure the BBS	9
3.6	The S:IceBBS.HAIL File	10
3.7	The S:IceBBS.BINIT File	10
3.8	The S:IceBBS.DIRS File	10
3.9	The S:IceBBS.NEWSIGS File	11
3.10	Setting up the Feedback Section	11
3.11	Enabling the Remote Shell	12
3.12	Running the BBS	12
4	Serial Ports	15
4.1	Internal Serial Port	15
4.2	A2232	15
4.3	ASDG - Dual Serial Board	18
4.4	Multiface II and III dual serial cards	18
4.5	ComPorts 8 port card	19
4.6	Applied Engineering’s Internal Modem	19
4.7	Serial Cable Requirements	19
4.8	Direct Line Hookup	20
4.9	Simulating a null modem in software	20
5	Required Libraries	23
6	Modem Register Settings	25
6.1	Old style modems	25
6.2	Advanced modems	26
6.3	MNP Modems	27

to see if it is addressed to a person on your BBS. If it finds this is so then it will place a *copy* of the message in the person's email SIG. This was done so that someone can make a posting to a FidoNet SIG area that he does not normally read and then wait for replies to appear in his email rather than having to read a lot of messages in that SIG in the hope of finding a reply. Some users (particularly if they are active on several BBSs) find this annoying, for this reason there is a button marked 'CC: Fido' in IceTERM they can use to disable this feature.

29.5.3 Fido

This is a debugging utility that can be used to inspect the contents of FidoNet packed message files⁹ (*.PKT' and ' *.CUT' files). A packed message file contains one or more individual messages. All the messages in a packed message file are being sent to the BBS for which the file is named (in the case of outbound messages), the individual messages may then be passed on to other BBSs depending on their software.

The 'fido' utility prints the contents of the packet header, the contents of each message header, and the contents of each message found. It will expand CTRL-A characters to a '<CTRL-A>' string, as these are fairly important in FidoNet. If you are trying to diagnose a problem in echo mail this command will allow you to inspect the 'AREA:', 'SEEN-BY', and 'PATH' fields.

29.5.4 FidoPost

This program is designed to make a FidoNet netmail posting out of a plain ASCII text file. It takes the ASCII file, along with a user name, an address and a subject string and converts it into a '*.CUT' file in the FidoNet outbound directory (usually 'MAIL:OutBound' in a TrapDoor installation). The '*.CUT' files will be named for the BBS they are going to with with a four dimensional address (eg: '1.134.92.0.CUT'). If such a file exists already the software will append the message to the packet.

This program will compare the destination and source addresses, if they are both in the same zone and net then the message will be written for direct transmission to the destination. If they are within the same zone but in different nets the message will be externally addressed to your net host (which is usually node one in the same net as you, but if different you can use the '-h' switch) but will be marked internally for forwarding to the true destination. If the message is addressed out of your zone then it will be routed through your zone's zonegate¹⁰.

⁹ To inspect compressed mail files you will need to unpack them first.

¹⁰ This has not been extensively tested at this point.

Short Contents

1	Conventions	1
2	Introduction	3
3	Quick Start	7
4	Serial Ports	15
5	Required Libraries	23
6	Modem Register Settings	25
7	Access Control	33
8	The IceBBS.ACCESS Files	39
9	The S:IceBBS.SHELLACCESS File	41
10	The S:IceBBS.MAXACCESS File	43
11	The S:IceBBS.MSGID File	45
12	The S:TRIM.LASTTRIM File	47
13	The S:IceBBS.SIGS File	49
14	The S:IceBBS.NEWSIGS File	51
15	The S:IceBBS.BINIT File	55
16	The S:IceBBS.DIRS File	63
17	The S:IceBBS.HAIL File	65
18	The S:IceBBS.ARBITRATE file	67
19	The S:IceBBS.NOASCII file	69
20	The IceBBS.FIDOCFG file	71
21	The ASCII Menus	73
22	Error Message Files	85
23	Message of the Day	89
24	Account	93
25	Edituser	95
26	EU, the MUI Edituser	97
27	The Coupon System	99
28	Sharing Lines	101
29	FidoNet and the BBS	105
30	UUCP and the BBS	113
31	Internet tools	119
32	Finding Files	123
33	File System	131
34	Message Bases	143
35	Mail	161
36	Electronic Mail	163
37	Monitor	171
38	Conferencing	173
39	News Bulletins	175

29.5.1 Export

This program will scan the various areas of the SIGs message base and extract any new postings or replies and export them to the FidoNet outbound mail directory. TrapDoor will occasionally scan the outbound mail directory and any packets it finds will be sent out to their destinations.

The **'export'** program will only scan one SIG per time it is run. This is because when connected to FidoNet you might receive one SIG from one site, several from another, and have a few more that are local to your BBS only.

'Export' works by doing a recursive scan of the specified IceBBS SIG area and extracting any messages that are newer than the last scan date and then appending those messages to the appropriate **'*.CUT'** file (in the FidoNet outbound mail directory) for transmission to the destination system. If a **'*.CUT'** file for the destination site does not already exist it will create one.

Usually **'export'** is run once for each SIG to be exported somewhere near the begining of the FidoNet script file (before any TrapDoor calling takes place).

It will also verify that they any new messages have not already been seen by the node we are exporting to (by looking for FidoNet's **'SEEN-BY'** lines) and append them to the appropriate CUT file only if they have not been seen yet.

'Export' has the ability to act as a hub (to some extent) allowing messages that were imported to a IceBBS SIG to be exported to other nodes. It checks the **'SEEN-BY'** lines to see if the node that the message is being exported to has been seen, if it has not then the export proceeds, if the message has already been seen then **'export'** skips it. The **'PATH'** line is also updated. An additional flag **'-g'** can be used if you wish to export a SIG to another zone (g is for zoneGate). When the **'-g'** is present the **'SEEN-BY'** lines for the source zone will be removed and fresh **'SEEN-BY's** for the new zone will be added.

If you are acting as a zone gate then your address (the **'-n'** flag) should specify your address in the zone you are exporting to. So if I was exporting a SIG from **'1:134/*'** to a BBS in **'222:1/*'** both the **'-n'** and **'-d'** flags would specify addresses in **'222:1'**. The machine in the other zone you export to does the reverse procedure when it exports messages from **'222:1'** to **'1:134'**.

'Export' generates message identification kludge lines (MSGIDs) as it exports the messages, the actual unique identification number is assigned to the message when it first enters the system, so if a message base must

Serial ports, A2232.....	15	SIGs: traversing links.....	153
Serial ports, ASDG card.....	18	Slow CDROM directories.....	63
Serial ports, cables for.....	19	Sorting, files.....	127
Serial ports, comports 8 port card...	19	Spaces.....	1
Serial ports, configuring.....	59	Special interest groups, see SIGs...	143
Serial ports, multiface card.....	18	Specifications.....	3
Serial ports, null modems.....	20	Splitting files and descriptions.....	134
Serial ports, sharing with other programs.....	101	Square brackets.....	1
Serial ports, supported.....	5	stack, door tool type.....	179
Server.....	179	Statistics, for find databases.....	128
Server shutdown.....	85	Status of internet.....	90
server, door tool type.....	179	Styles, in text.....	199
Server, problem starting.....	85	Support.....	213
Server, see doors.....	58	SupraFAX modem.....	29
Session limits.....	37	Swapping, of CDROMs.....	67
Session passwords.....	105	SyQuest drives.....	195
Setting up.....	7	Sysop, message reader.....	159
Sharing serial lines.....	101	System name.....	57
Shell access.....	41	System privileges.....	36
Shell, no access.....	85		
shutdown.....	85	T	
Shutdown line.....	85	Talk.....	23
Shutdown server.....	85	Task priorities.....	197
SIGs.....	145	Task quota limit.....	85
SIGs, access control.....	52	tempdirinuse.....	85
SIGs, adding new ones.....	160	Templates, for account types.....	97
SIGs, data items.....	149	Temporary directory in use, message	85
SIGs, exporting messages from.....	107	Temporary files, saving.....	138
SIGs, exporting messages to USENET	115	Terms, explained.....	207
SIGs, feedback.....	143	Testing files, CRC.....	134
SIGs, file format.....	145	Testing, integrity of files.....	140
SIGs, finding a message by number	153	Texinfo.....	1
SIGs, finding free space.....	159	Text on first connect.....	65
SIGs, finding the next message.....	157	Text only BBS.....	73
SIGs, finding the previous message	158	Text, ANSI codes.....	199
SIGs, first link.....	146	Text, bold, italics etc.....	199
SIGs, importing from FidoNet.....	108	Text, erasing.....	200
SIGs, importing from USENET.....	114	Thanks.....	217
SIGs, link item.....	147	Time bank.....	37
SIGs, moderator access.....	51	Time limits.....	37
SIGs, names of.....	51	Time of day, testing.....	110
SIGs, reader for.....	159	Time of last trim.....	47
SIGs, trimming.....	47, 160	Timed script files.....	61
SIGs, VMSG format.....	149	Too many calls, message.....	85
		Tool types, for doors.....	179
		Tool types, problems with.....	185
		Tools, control of access.....	36

29 FidoNet and the BBS

29.1 Introduction

FidoNet is supported with a number of external utility programs that run along side IceBBS. Additionally, you will need a front end mailer for FidoNet, TrapDoor¹ is a good choice. WPL or JamMail² (formerly Welmat) is another front end package that IceBBS has been used successfully with, most BBSes will have a copy. The current implementation allows incoming FidoNet calls to be received at any time, and your system can poll other systems during the day for news and to send new mail.

29.2 Script Files

There are two approaches to implementing the FidoNet support script files for use with IceBBS. One is to make use of the timed script file capability of the BBS, this is fine when the BBS is controlling access to the line. The other is to run a script file in a loop or via a ‘cron’ program, this is choice when the BBS is using a front end program like TrapDoor to control the modem (and hence the BBS cannot run script files on that line).

29.3 Security

There are several ways in which security can be applied in FidoNet communications. The primary line of defense is the session level password. This is a password that your front end software³ and the corresponding software on the other machine that is calling you or you are calling share. The two mailers will exchange passwords during the session handshake (before any mail is exchanged) and, in the case of TrapDoor, if the passwords match any mail packets that get transferred will be marked as secure.

¹ TrapDoor is a product of Maximilian Hantsch and Martin Laubach, it is a shareware product and should be available from any good BBS. Failing that you can write: TrapDoor Development, c/o Maximilian Hantsch, Matzleinsdorfer Platz 3-4/3/10, A-1050 Wien, Austria, Europe.

² Written by: James McOrmond, 264-2nd Avenue, Ottawa, Ontario, K1S 2H9, Canada.

³ Such as TrapDoor or WPL.

L

Levels, access	33
LHA	140
Libraries, required	23
Limitations, of file finder	129
Limits, default	37
Line shutdown	85
linenotopen	79, 85
Lines, see also serial ports	59
Lines, see serial ports	15
Lines, sharing with other tasks	101
Linkage files, for SIGs	145
Links, in SIGs	153
Listing new files	138
Listings, of files	138
Log file	55
Login message	85
Login message, see also MOTD	89
Login text	65
Logout	73
Lost or bad serial data	15
Low priority mail, LPM	109
LPM	109
LZH	140

M

Mail, see also email	116, 163
Makemotd	89
Making IceBBS files	133
Man	51
Marrieta	217
Mastering, CDROMs	139
Max access	43
Memory needed	5
Menu files, first for ASCII	57
menuname, door tool type	179
Menus, ASCII mode	80
Menus, in ASCII mode	73
Message bases, see also SIGs	143
Message of the day, also MOTD	89
Message, sent when modems connect	65
Messages, IDs for	45
Messages, see also error messages	85
Messages, trimming	47
MNP errors	32
MNP modems	27
Modems, advanced	26

Modems, Hayes	25
Modems, high speed	26
Modems, internal	19
Modems, MNP	27
Modems, null	20
Modems, old style	25
Modems, radio	119
Modems, see also serial ports	59
Modems, supported	5
Modems, SupraFAX	29
Modems, USR Dual Standard	29
Modems, USR HST	28
Modems, V.32 and V.32bis	30
Modems, V.34	31
Modems, V.42 and V.42bis	31
Moderator access, to SIGs	51
Monitor, history depth	57
Monitor, of user activity	171
More, ASCII command	76
motd	79, 85
MOTD, adding a file to it	91
MOTD, adding account info. to	90
MOTD, adding current network status	90
MOTD, adding new files listing	90
MOTD, does not appear	89
Mountlist, for VAUX handler	177
Movement, of text cursor	200
MSGID	45
MSGID, FidoNet	107
MUD	183
Multiface, serial card	18
Multitasking, slow	15
Multitasking, with other programs	197
MySigs file	51

N

Narrator	23
Net mail, exporting	111
NetMail	165
Netmail, sysop utility for	109
Network, testing status of	90
New file listings	138
New files listing	90
New user, message for	85
New users	37
News bulletins	175

It is advisable that any script files you start in this fashion should have a 'STACK 10000' (or so) command followed by a few 'PATH' commands at the beginning of them. Otherwise you may find commands used within the script crashing because of insufficient stack, or perhaps not even being run because AmigaDOS cannot find them in the default search path that the script file is started with.

I am successfully running UUCP v1.15 and TrapDoor along with IceBBS in this fashion, see the UUCP and FidoNet chapters for more information. An example script file is included in the distribution archive, look for the 'complex_script' file in the 'BBS_S' directory.

28.3 Front door programs

IceBBS can be run in a shared serial port mode, which allows several programs to have the serial port open at the same time and share its use between them (see Section 15.16.3 [Use with TrapDoor], page 60). This is implemented via the 'S:IceBBS.BINIT' file and the 'bbsnow' command.

The 'bbsnow' command takes several additional parameters

- l reports the speed of the modem-modem link rate (in baud),
- b reports the speed of the serial port to modem link (in baud),
- v forces the BBS to start the IceTERM protocol right away without giving the user a chance to select ASCII mode,
- a forces the BBS to start the ASCII mode right away without giving the user a chance to select the IceTERM protocol.

for example the BBSNOW command string for trapdoor is:

```
BBSCOMMAND "dh2:bbs/bbs/bbsnow VBBS2 -l%B -b%b"
```

Doors, theory of operation	180	Email, via FidoNet	165
Doors, tool types for	179	Email, via the Internet	166
DOS Shell	177	emailfileswaiting	85
DOS shell, access levels for	35	emailwaiting	85
DOS Shell, access to	41	Encryption, see also PGP	163
DOS shell, no access	85	Enhancements	205
Downgrading accounts	99	Erasing, text	200
Download ratio exceeded message ...	85	Error messages, files for	85
Downloading, file reads	58	Error messages, specifying directory ..	56
Downloads not allowed, message	85	Errors, MNP	32
Dreams	205	Escape sequences	199
Drive not online message	85	EU	97
drivenonline	85	Exceeded ratio	85
DRM-600, Pioneer drive	67	exceededbratio	85
Dropped connections	32	exceededdaily	85
DSB, ASDG dual serial board	18	exceedednratio	85
DSR, data set ready	19	exceededquota	85
Dual standard modem	29	Expansion, see doors	179
		Export, of messages from SIGs	107
		Exporting, email	111
		External protocol libraries	80

E

Echo, replacement for	111
Edituser, GUI version	97
Electronic mail, see email	163
Ellipses	1
Email	163
Email files waiting message	85
Email SIG, name of	52
Email waiting message	85
Email, access to	52
Email, between users on the BBS ..	164
Email, cannot find user	85
Email, checking for waiting mail ...	166
Email, configuring for Internet	59
Email, directories for	163
Email, downloads	163
Email, FidoNet and Internet	116
Email, files	163
Email, forwarding	169
Email, forwarding to the Internet ..	166
Email, forwarding to users	116
Email, importing from USENET ...	115
Email, internet not available	85
Email, on FidoNet	109, 111
Email, sending files	167
Email, sending junk mail	164, 167
Email, specifying directory	56
Email, sysop command for	167
Email, user profile	163

F

Features	3
Feed, finding a	113
Feedback, from ASCII mode	77
Feedback, old SIG system	143
Feedback, specifying files for	57
Feedback, sysop's reader	161
FidoNet node number, default	71
FidoNet packets, viewing contents ..	109
FidoNet, arc mail	108
FidoNet, email utility	109
FidoNet, exporting messages to	107
FidoNet, importing to SIGs	108
FidoNet, net mail	111
FidoNet, PKT files	108
FidoNet, posting to	109
FidoNet, security on	105
File areas	63
File descriptions, directory for	58
File descriptions, in ASCII mode	75
File does not have description	85
File download, read size	58
File format	131
File format, indirected	133
File not deleted message	85
File security	39

28 Sharing Lines

Some times it is necessary to share a line that the BBS is using with another program. IceBBS offers three ways of doing this¹.

28.1 Pause Line

IceBBS has a *pause line* feature that is useful for sysops who wish to dial out to do other things when the BBS is not busy. Each modem line monitor window has a menu with a pause line entry in it. By selecting a line and selecting the pause line menu item for that line you make IceBBS release that line for your use. You can then proceed to use a regular terminal program to dial out on that line and visit other systems. Once you are done using the line you should just select the ‘Restart’ menu item and the line will be returned to IceBBS for it to use.

The pause line mode also works for lines that are shared with another program (like TrapDoor) but you will also have to get the other program to release its hold on the line before you can use it.

28.2 Timed Script Files

IceBBS also has another way of sharing its lines with other programs. For each line you can specify the name of an AmigaDOS script file (which can contain whatever commands you want to be run) that is to be executed at a certain time of day. A different script file can be specified for each modem line, and the script files may be run at different times of the day and even at different frequencies. In fact script files may be run as often as once every 15 minutes, or and infrequently as once a week or more.

Timed script files are configured as part of the serial line configuration lines in the ‘S:IceBBS.BINIT’ file (see Chapter 15 [IceBBS.BINIT file], page 55). The 11th entry of one of these lines determines whether or not a script file will be run for this line. If this entry is a ‘-1’ or ‘-2’ then no script file is to be run. If the 11th entry is a positive number then a script file will be run on this line occasionally and two additional numbers and the name of the script file must also appear on the line.

Together this leaves us with four entries that control script execution: three numbers and a file name. The first number is the time of day at which the script file will be first run after the BBS starts. This time is in seconds from midnight, so to run a script at 7 AM you would enter

¹ OwnDevUnit support will be probably be added at some time in the future too.

Concept Index

0

00_FILE.LST, making and using ...	139
00_INDEX.TXT, making and using	139

A

A2232 serial card	15
A590	190
Access control	33, 137
Access control, CDROM	138
Access control, directory for	58
Access control, on CDROMs	189
Access control, to DOS shell	177
Access levels	33
Access levels, number of	35
Access levels, setting	36
access, door tool type	179
Access, to DOS shell	41
Accounts, automatic upgrades and downgrades	99
Accounts, creating new ones	161
Accounts, editing remotely	183
Accounts, editing with EU	97
Accounts, file format	187
Acknowledgements	217
Adding SIGs	160
AmiCDROM	192
Amiga Library Services	193
Amiga serial port	15
Amiga Users of Calgary	193
AMUC	193, 217
anonymous	79, 85
ANSI codes	199
Appending files	91
Applied Engineering, internal modem	19
Arbitration, of CDROMs	67
ARC	140
Arc mail	108
Areas, see SIGs	143
arexx, door tool type	179
ARexx, doors	183
ARP library	23

ASCII commands, hiding from users	79
ASCII menus	73
ASCII menus, chaining	78
ASCII mode BBS, turning off	69
ASCII mode, first menu file	57
ASCII mode, hot keys	75
ASCII mode, menu directory	57
ASCII, defining new commands	73
ASDG, serial card	18
AsimCDFS	192
AsimWare Innovations	192
Assign, defer option	195
Assign, use of defer option	189
Auto	51

B

Bad password	85
Baldrick, a dog's body	97
bankbroke	79, 85
batch directory	115
BBS log file	55
BBS name	57
BBS, ASCII mode, turning off	69
BBS, initialization file	55
BBS, text mode	73
Bits, privilege control	36
Black Adder, the	97
Bold face	199
Brackets, square	1
Buffers, for CDROMs	67
Building, find database	128
Bulk file operations	139
BULKTEST.CFG	140
Bulletins, see news	175

C

Cables, for serial ports	19
Calls, too many	85
Canadian Prototype Replicas	191
cantfindnewsitem	85
Carrier detect	19
CD, carrier detect	19

27 The Coupon System

Maintaining user accounts on a large BBS system can be a time consuming chore. A lot of the work you do is quite repetitive, and thus can be simplified by making AmigaDOS scripts that encapsulate sets of ‘edituser’ commands or by making use of ‘eu’ templates. However, there are still things you cannot accomplish with such a semi-automatic approach:

- automatically downgrading the account on a certain date or after a certain time,
- automatically upgrading the account when the user wants more access,
- being able to sell (or give away in a controlled fashion) account upgrades which users can apply as they need, when they need.

The automatic coupon system addresses all of these points. The users *cash in* their coupons¹ at any time they want. IceTERM v115 and higher provides the necessary user interface (which is just a few string prompts to get four 8-digit numbers from the user). Then the BBS checks the coupon numbers against the set it holds, and if there is a match it runs the appropriate AmigaDOS script file. The script file can contain anything, but it usually contains a number of ‘edituser’ commands to effect the upgrade. Once this is done the user’s account is updated and the next time he logs into the system he will have the new access permissions or levels.

¹ The BBS ensures they can use a coupon only once, and the coupons are secure against cracking.

Command Index

A

Access	137
account	73, 93
age	145
alias	73
amuc.coupon.rx	99
append	91
assign	189

B

batchnews	115
BBSJoin	133
bbsnow	103
BBSSplit	134
BulkJoin	140
BulkSplit	139
BulkTest	140
bye	74

C

cd	74
ChangeTaskPri	197
chat	74
CheckMail	166
CircleMUD	183
confer	75, 173
Coupons	99
ctrl	75

D

defer	189
DelOld	137
describe	75
dir	76
display	76
download	76

E

edituser	95
email	77
Email	167
email2cut	111

eu	97
EUMUI	183
export	107

F

feedback	77
fido	109
fidopost	109
fidotime	110
FileList	138
find	123
findbuild	128
findscan	127
findstat	128
fmt	135
forward	169
fwdemail	116
FwdEmail	166

G

getty	114
goto	78
GPRReader	145

H

HashAccess	138
------------------	-----

I

IceBBSNEWUSER	37
IceOLR	159
import	108
inmail	115
InMail	166
isnetup	90

J

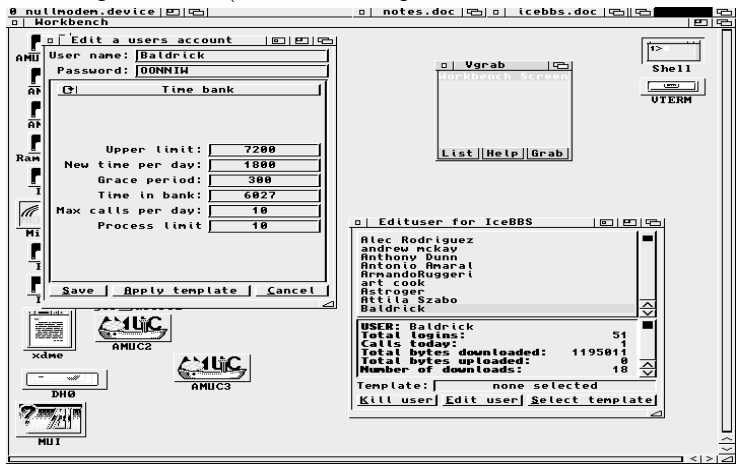
JamMail	105
---------------	-----

L

lock	79
login	79

26 EU, the MUI Edituser

‘Eu’ is a MUI based version of the ‘edituser’ command. It is great for browsing through accounts looking for dead users, duplicates and messed up user names. ‘TAB’ key cycling through the gadgets and between the windows with ‘ALT-TAB’ is implemented, so you can literally use it without the mouse. I there is also a remote sysop version of this program (see Section 41.6 [Edituser door], page 183), so a co-sysop can more easily able maintain your users and won’t need shell access to do so. The following figure shows the main window where you select the account you want to edit (by double clicking on a user name) and the account editing window (shown editing a user named “Baldrick”¹).



The EU Tool

The ‘eu’ tool has a powerful feature called account templates. These allow you to build a set of template files that describe the various types of standard accounts on your BBS. You can then select a particular template from a list and apply it to various users. This allows you to quickly upgrade and downgrade users with less chance of forgetting something.

You create templates to describe what characteristics each sort of user gets, then to add those settings to a user’s account you just use the apply template function. Note that all the template entries have two parts: the attribute (like a session time limit) and a checkmark to the right of it. If the checkmark is checked then that attribute will be changed to the value you have set when the apply template button is pressed. If the checkmark is clear then the apply template function will not change that attribute in the user’s account.

Tool types and Buttons Index

.		M	
.dat	145	menuname	179
.lnk	145	MessageHeader	149
		MsgHeader	143
A		N	
access	179	na	115
arexx	179		
		P	
D		priority	179
FormItem	149		
		S	
F		SeekNextFree	159
FindMessageAfter	157	server	179
FindMessageBefore	158	stack	179
FindMessageNo	153		
FirstLinkItem	146	V	
		VMSG	149
L		W	
LinkItem	147	world	115

¹ Obviously a Black Adder fan.

25 Edituser

25.1 Introduction

This is the utility the IceBBS sysop will use to edit account files (the format of which is given in the next section). This is a CLI utility that has a number of command line switches to allow the sysop to do most anything he wants.

Related topics include the MUI-based account editor (see Chapter 26 [EU], page 97), the coupon system (see Chapter 27 [Coupon system], page 99) and the chapter on access controls (see Chapter 7 [Access control], page 33).

Typing `'edituser'` by itself will bring up a help screen about the utility.

`'Edituser'` can be used to edit a single user's account or to perform operations across all selected accounts. All IceBBS accounts are stored as individual files in a single directory. `'Edituser'` reads the `'S:IceBBS.BINIT'`¹ file to determine what directory to look for account files in.

The edituser command replaces the `'setaccount'`, `'setuser'`, `'pass'`, and `'account'` commands present in previous² IceBBS releases. These older utilities should not be kept in use as the structure of the account files has changed and they will cause problems.

51 Thanks

I would like to thank all those who helped in the testing of IceBBS and its terminal program. In particular I would like to thank the members of the Amiga Users of Calgary Society (AMUC) who called a lot and often loaned me the use of some of their phone lines for the multi-line BBS testing.

Last but not least, thanks to Marrieta for putting up with all the time I spend on IceBBS developement and for baking all those cookies.

¹ See the `'S:IceBBS.BINIT'` chapter for information on the format of this file and how to specify the name and location of the user account directory. Note that you should not store any other files in the user account directory as the `'edituser'` program may modify or delete any file it finds there.

² Versions before v45.

24 Account

This utility program is now obsolete, along with ‘`pass`’, ‘`setuser`’ and ‘`setaccount`’. In fact all of these have been replaced by the ‘`edituser`’ program (see Section 25.1 [Edituser], page 95). Do not use any of these obsolete programs with the newer IceBBS versions, as doing so will damage the account files as the format of these files has changed.

50.5 Disclaimer

No warranty, either express or implied, is made with respect to the fitness or merchantability of IceBBS or its support utilities.

Stephen Vermeulen (referred to as *the Author*), reserves the right to not develop any future versions of IceBBS or its support utilities.

The Author will try to make a good faith attempt at correcting any problems if any are discovered, but is in no way required, nor bound to correct them.

The Author neither assumes or accepts any responsibility for the use or misuse of these programs. The Author also will not be held liable for damages or any compensation due to loss of profit or any other damages arising out of the use, or inability to use these programs.

The Author will not be liable for any damage arising from the failure of these programs to perform as described, or any destruction of other programs or data residing on a system attempting to run the programs. While the author knows of no damaging errors, the user of these programs uses it at his or her own risk.

23.5 Append

The ‘**append**’ command is used to append two files, much like the AmigaDOS ‘**join**’ command, but it does not have to create a third output file. This command is primarily intended for use in building up the message of the day.

50 Matters Legal

This chapter contains information about various legal matters relating to the software.

50.1 Direct Updates

The current version of the complete IceBBS and utilities set, including the printed documentation for IceTERM and IceBBS is available for US\$60.00 (plus \$10.00 for shipping) from the Author at the address in the following *Support* section. This price may go up in the future as the project gets larger.

For registered users software upgrades are available for free from the support BBS (once an account has been created for you). Upgrades by mail (on floppy disk) may also be obtained for US\$10.00 per set. You can pay in advance for more than one upgrade at a time, in which case upgrades will be sent out automatically as they are available (typically there is about 6 weeks between upgrades), or when you request them by mail or email.

The manual set is also being upgraded from time to time. Between releases of the manual notes on the changes are provided with the software upgrades, and by issuing new versions of the AmigaGuide version of the manual. A complete upgrade (fresh printed copy of the manual and the current IceBBS software on disk) is US\$20.00.

50.2 Distribution Restrictions

IceBBS and its utility programs and documentation may not be distributed by anyone without prior written consent of the Author, Stephen Vermeulen.

Only the demonstration version of IceBBS along with the demonstration version of the documentation (in AmigaGuide or PostScript format) may be freely distributed. The full version of IceBBS along with the tools for FidoNet and USENET message processing and a printed copy of this documentation are available from the Author for US\$60.00, plus \$10.00 for shipping.

50.3 Support

If you have any suggestions, bug reports etc., feel free to contact the Author at:

23 Message of the Day

Most BBS systems allow the sysop to create and maintain a file which is shown to each user when he logs into the system. IceBBS allows this through the `'motd'` (message of the day) file that is stored in the error messages directory. This file is shown to all users when they log into the system. As a sysop you can change this to anything that you wish to. The recommended use for this file is to provide recent BBS statistics, names of new uploads, and other important information.

There are actually two `'motd'` files, one for users of IceTERM and the other for ASCII mode users. The ASCII mode version of the `'motd'` file is stored in the ASCII menus directory.

Like other files stored in the error messages directory the first line of the `'motd'` *must* only contain four numbers to specify the size and position of the window to display the message in. If you fail to do this you will find IceTERM users complaining that they don't see the message of the day.

23.1 Customizing

The login message that is sent to IceTERM users can be customized¹ to include information specific to the user. This is accomplished with the `'S:MAKEMOTD'` script file.

When a IceTERM user logs into the system the BBS will check the `'S:'` directory to see if the file: `'MAKEMOTD'` exists. If this does not exist then the BBS will only send the user the regular `'motd'` message. If the file does exist then the BBS will attempt to execute it. `'S:MAKEMOTD'` can either be a script file or a regular program. When the BBS executes the command it will pass it two parameters, the first is the complete path and filename of a temporary file which the BBS will send to the user when the `'makemotd'` command is done, the second is the complete path and name of the user's account file.

With this information a custom login script can extract user-specific information and write it into the temporary file, the BBS will then send this to the user. This can be useful for displaying specific account information (ie. account limits) or other interesting information (ie. new files since the user was last on the BBS).

To support this facility two utilities are provided which are described in the next sections. A `'makemotd'` script file that uses these is shown below:

V.FAST	Because the international standardization committee of the CCITT was taking so long defining the V.34 standard a number of modem manufacturers started making fast modems based on the proposed standards. These modems are usually advertized as V.FAST modems and were available at various speeds for about a year before the true V.34 modems started shipping.
WPL	A FidoNet frontend mailer package for the Amiga, this is the software that handles the transmission of files between FidoNet systems.

¹ With version 47 and higher of IceBBS.

nodoorserver

this is sent to a user if he tries to use a door which does not have a server installed on this BBS. This may also be sent if the server cannot be loaded for some reason (version mismatch or insufficient memory for example), see Section 41.1 [Doors], page 179,

nofiledesc

this message is sent whenever a user requests a long file description about a file in the library which does not have a description attached, see Section 33.1 [File format], page 131,

nointernetmail

this is sent when a user attempts to send some internet mail (using the Send Internet button in email) on a BBS that is not running UUCP (ie. one that does not have a valid 'sendmail' entry in the 'S:IceBBS.BINIT' file), see Section 15.15 [Line 16], page 59,

noserver this will be sent if there is a problem trying to start one of the built in BBS servers, usually this indicates a fairly serious problem, like low memory,

noshellaccess

this will be sent if the user is trying to use the remote AmigaDOS shell function without having been given permission to do so, see Chapter 40 [DOS shell], page 177

notaniffpic

this will be sent if the user is trying to view a file that is not an IFF picture with the 'preview' door. Note that this error message is included with the 'preview' door archive, see Section 41.5 [Preview door], page 182,

nouserprofile

this is sent when a user requests the profile file belonging to another user who has not created one yet,

shutdown this message is sent when a debugging version of IceBBS is being run and a user selects a function that is not available, or is under construction,

tempdirinuse

this can be sent by the remote sysop's file maintenance door when there is some sort of conflict between two co-sysops working at the same time, see Section 41.4 [RemFile door], page 181,

toomanycalls

this message is sent to the user if he has called more times in one day than his account limit allows, see Section 7.8 [Time limits], page 37

the Amiga it is usually used to indicate the end of a line of text.

Modem

A modulator-demodulator, in other words an expensive black box that will allow you to connect a computer's serial port to the phone lines and through it send and receive data from other distant computers.

MNP

Microcom Networking Protocol, the first widely available and successful form of ARQ (automatic error correction) on modems. This is available in a variety of levels of sophistication, usually with a numeric designation. The normal basic levels are 4 and 5 with levels up to 10 being available on some modems today.

MSGID

Message identifier, this is some unique number or string that is embedded within a message, its primary purpose is to help in the detection and deletion of duplicate messages in the FidoNet echomail and USENET conferences (SIGs). It can also be used to help in the detection of loops in the network topology. USENET messages also use the message IDs to link postings and replies. See Chapter 11 [IceBBS.MSGID file], page 45 and Section 34.2.5 [VMSG struct], page 149 for more information.

PGP

Pretty Good Privacy, a freely distributable encryption package that runs on just about every computer platform. This package features the RSA public key algorithm to make exchange of keys easy and includes a number of interesting features for key management (trust networks and key revocation) and public message authentication (clear signing).

PKT

A message packet file in FidoNet. This file may contain several messages, usually these will be compressed into a bundle for transmission, see Section 29.5.2 [Import FidoNet], page 108 and Section 29.5.3 [Viewing FidoNet packets], page 109.

PPP

Point to Point Protocol, the replacement for SLIP and CSLIP.

SIG

Special Interest Group, a group of computer users who share a common interest in some topic and meet to discuss it, either in person or electronically.

SLIP

Serial Line Internet Protocol, the error detecting and correcting link software that usually runs on a serial line connecting two computers. It is used to transmit Internet data packets between the two computers.

22 Error Message Files

There are about 30 text files which must reside in the error messages directory¹. Each of these files is essentially a plain ASCII text file that will be sent to the user of IceBBS whenever certain events occur. You are free to modify these files as you please, but remember to keep them relatively short and informative, and to take care in selecting your display box dimensions². The names of the files and their intended use is listed below:

anonymous

this message is sent to any user who successfully logs into IceBBS as an anonymous user,

bankbroke

sent to a user who has run out of connection time credits when he logs in, see Section 7.8 [Time limits], page 37,

cantfindnewsitem

this message is sent by the news system if one of the news bulletin files cannot be found, see Chapter 39 [News bulletins], page 175,

drivenotonline

sent when a user attempts to download a file from a disk that is not currently in a drive. This would typically be encountered when you have a single CDROM drive and several CDROMs which you have placed the description files for on your hard drive, see Section 33.2 [Indirected storage], page 133,

emailfileswaiting

if a user has files waiting in his email files directory when he logs in this will be sent, see Section 36.1 [EMail directories], page 163,

emailwaiting

this message is sent to any user who has email waiting for him in his email directory when he logs in, see Section 36.2 [Mail between users], page 164,

¹ See the 'S:IceBBS.BINIT' file chapter for information on how to define the location of this directory. The default is 'ErrorMessages' in the 'IceBBS:' directory.

² You must make certain that the first line always contains four numbers, these are the X and Y location and width and height of the window that is to be opened on the user's terminal to display the message.

49 Glossary

This is a list of various mysterious terms that appear from time to time in this manual.

Arc mail A FidoNet term for a compressed bundle of mail. So named because the first archiver used to do this was the venerable 'arc' program. See Section 29.5.2 [Import FidoNet], page 108 for more details.

ARQ Automatic Repeat reQuest, a term for the error detection and retransmission error correction schemes to be found on various more advanced modems. Examples of this are the MNP and V.42 protocols. When a pair of modems are communicating through an ARQ controlled link almost all errors introduced by line noise will be detected and automatically corrected without the intervention of the computer at either end.

ASCII American Standard Code for Information Interchange, this is one of the earlier (and by far the dominant) standard for describing which binary values mean which letters in a computer's memory or in a disk file.

BAUD A rather abused term, this originally meant the number of distinct symbols per second that a communications line carried. The problem with this definition is that the number of bits of information per symbol is left variable so BAUD rates are somewhat ambiguous. In fact the common 2400 BAUD modem is really a 600 BAUD modem which sends one of 4 possible symbols (hence 2 bits of information) at a time; and hence, transmits 2400 bits per second (bps). Over time BAUD has become synonymous with bits per second.

BBS A Bulletin Board System, a computer system that is usually connected to the phone lines via modems and is used to facilitate the interchange of information (files, messages...) between various other computers.

CR Carriage Return, this is ASCII character 13, it is a character that a computer sends a printer to tell it to return the print head to the beginning of the line. It is also used to indicate the end of a line and in the case of a modem the end of a command. In the MSDOS world the pair of characters CR and LF together is usually used to indicate the end of a line of text.

CLI The Command Line Interface of AmigaDOS.

21.23 top

This command is a special form of the ‘cd’ command. It always takes the user to the root directory in which only those directories in the ‘S:IceBBS.DIRS’ file are visible.

21.24 upgrade

This command (which takes no arguments) will prompt the user for the serial numbers of an upgrade coupon (see Chapter 27 [Coupon system], page 99). It will then check the numbers and if they are correct it will implement the upgrade. If the numbers are incorrect it will inform the user of the problem and then the command is finished.

21.25 upload

This command starts the xmodem upload process. This can be aborted by sending a stream of ‘CTRL-X’ characters or by breaking the connection.

21.26 users

Not yet implemented.

21.27 who

21.28 xdown

This is the command to download a file from the BBS to the user using an XPR protocol which has been selected by the ‘protocol’ command (see Section 21.20 [ASCII protocol], page 80). Note that if the XPR supports wildcards then they can be used to transfer a batch of files. The syntax is:

```
xdown filename
```

Wildcards supported include *full* AmigaDOS pattern matching, which allows a command like:

```
xdown ~(#?.lzh)
```

to download all the files that *do not* end in ‘.lzh’, as well as the traditional ‘*’ wildcard.

48 Future Enhancements

More ASCII terminal access features, improvements to both message bases, more doors, remote file retrieval, additional FidoNet and USENET support, better sysop utilities, and protocol enhancements. Stand-alone file and message transfer utilities for access of IceBBS from AmigaDOS scripts and more direct Internet support.

There are other XPR libraries to support other protocols (such as Kermit). Let me know if you find any others that work.

The ‘protocol’ command is used to configure the protocol that the user wants to use. Its full syntax is:

```
protocol xprprotocol.library xpr_init_string
```

Where the ‘xprprotocol.library’ is the name of the XPR protocol library the user wants to use, and ‘xpr_init_string’ is some (optional) protocol-specific configuration string.

NOTE: this command *CANNOT* be entered by the user, this is because he might specify something like ‘graphics.library’ which would result in a crash. So you will have to place this command in a menu file. The following is a simple menu file that prompts the user for the protocol he wants to use:

```
alias temp y goto yprotocol
alias temp z goto zprotocol
prompt setproto.pmt
```

The prompting text (in file ‘setproto.pmt’) might look like this:

```
-----
y    to pick ymodem
z    to pick zmodem
(the protocol is automatically
saved when you logout)
-----
```

The ‘zprotocol’ menu file would look like

```
display zprotocol.txt
protocol xprzmodem.library
goto main
```

Where the file: ‘zprotocol.txt’ might look something like this:

```
-----
ZModem file transfer protocol
now enabled.
-----
```

When the user logs out his current protocol selection will automatically be saved in the user account files area.

21.21 sig

The SIG system is implemented with the ‘sig’ command, in a manner very similar to the ‘feedback’ command. The ‘sig’ command can be followed by a single parameter which must be one of:

```
next          non-threaded read next message,
```

‘6’	unknown <i>unknown</i>
‘7’	reverse video <i>reverse video</i>
‘8’	invisible <i>invisible</i>

21.15 lock

The ‘lock’ command is used to prevent the user from executing any of the built in IceBBS menu commands directly in a particular menu. It takes no additional parameters and it also prevents any of the permanent aliases from being used too.

21.16 login

This command takes no parameters. It prompts the user for his name and password, checks them, reports any problems to the user, and then sets the session timer appropriately.

The messages this command can send to the user are stored in the ASCII menus directory in files called:

- anonymous This is the message sent to anyone who just hits return twice to the login and password prompt. The session timer will be set to 5 minutes,
- bankbroke This is sent if the user is out of time credits,
- imposter This is the message sent to anyone who enters the wrong password. The session timer will be set to 5 minutes,
- linenotopen This is sent if the line the user is trying to use is in one of its shut down time periods,
- motd This is the normal login message to anyone who logs in successfully. The session timer will be set to the time limit in the user’s account,
- newuser This is the message that will be sent to all new users once they supply a name and password. The session timer will be set to either half an hour or the value in the ‘IceBBSNEWUSER’ account,
- toomanycalls This is sent if an ASCII user calls too many times in one day.

21.17 news

Not yet implemented.

background colour. Select colours from the list below: (Note: the following Amiga colours were with preferences settings of blue, black, orange and white—the default AmigaDOS 1.3 colours).

In the following tables the Amiga ANSI result is in italics (it is the second result).

- ‘30’ black foreground
blue foreground (text)
- ‘31’ red foreground
white foreground (text)
- ‘32’ green foreground
black foreground (text)
- ‘33’ yellow foreground
orange foreground (text)
- ‘34’ blue foreground
blue foreground (text)
- ‘35’ magenta foreground
white foreground (text)
- ‘36’ cyan foreground
black foreground (text)
- ‘37’ white foreground
orange foreground (text)
- ‘38’ unknown
unknown
- ‘39’ unknown
unknown
- ‘40’ black background
blue background
- ‘41’ red background
white Background
- ‘42’ green background
black background
- ‘43’ yellow background
orange background

abort an Xmodem transfer by typing a string of ‘CTRL-X’ characters into his terminal, or by breaking the connection.

21.12 email

Support for the old-style email has not been added to the ASCII side. However, support for the new email SIG is now complete, see Section 21.21 [ASCII SIGs], page 81 for more information.

21.13 feedback

This command allows the user to read, reply and post into the Feedback message section of the BBS. This command should normally be aliased to several user commands, as using the Feedback system with just this one command will require a lot of typing on the user’s part. The command takes a single parameter should be one of:

next	Moves the user to the next message and types that message to the user’s terminal, this is a non-threaded foward read,
prev	Moves the user to the previous message and types that message to the user’s terminal, this is a direct (not threaded) reverse read,
goto	This takes an additional numeric parameter which is the number of a message that the user wishes to move to and see. Note that if the user specifies a number that is too large the BBS will show him the last message in the Feedback data base,
last	This moves the user to the last message in the feedback database and displays it,
first	This moves the user to the first message in the feedback database and displays it,
again	This redisplayes the current message without modifying the current message counter,
enter	This puts the user into the on-line message editor, so that the user can post a message,
reply	This puts the user into the on-line message editor, so that the user can post a reply to the current message. Note that it preloads the current message into the edit buffer and quotes it for the user.

zmodem) is available through the external XPR protocol libraries (see Section 21.20 [ASCII protocol], page 80).

47 ANSI Codes

The chat, conference and feedback message system of IceBBS all use the Amiga’s console device for their text rendering. As a result of this a user may include any of the ANSI codes that the console device recognizes to achieve various style effects within his text. The most useful are the colour and style codes, but others (like cursor movement) may also be used.

47.1 Escape sequences

The escape sequence starts with either the two characters ‘ESC [’¹ or with the single ‘CSI’² character.

The ‘CSI’ character is followed by up to 3 rendition codes separated by semicolons and terminated by a lower case ‘m’. The rendition codes can appear in any order and none of them actually have to be present.

The first of these rendition fields is the *style*. This is a single character:

0	specifies plain text,
1	specifies bold-face text,
3	specifies italic text,
4	specifies underlined text,
7	specifies inverse-video text.

Note that to select several of these parameters you send a command like: ‘CSI 1;3m’ which would select both the bold-faced and italicized text modes.

The next rendition field is the foreground pen colour. This is the colour that the text will appear in. On a standard WorkBench screen there are 4 colours, and on a 3 or 4 bitplane screen the full 8 colours available may be used. The colours are selected by sending the ASCII numbers ‘30’, ‘31’, ‘32’, ‘33’, ‘34’, ‘35’, ‘36’, and ‘37’. The first four correspond to the normal WorkBench colours (blue, white, black, and orange). To set the text colour to black the user would send the sequence: ‘CSI 32m’.

The last rendition field is the background pen colour. This is the colour of the background for the text. The colours are selected by sending the ASCII numbers ‘40’, ‘41’, ‘42’, ‘43’, ‘44’, ‘45’, ‘46’, and ‘47’. The

¹ in hex: 1B 5B
² in hex 9B

21.6 confer

The ‘**confer**’ command will place the user into the online multi-user conference where whatever he types will be echoed to the other users who are online. Note that there are some features missing from this one that will need to be added later, including a way for the user to set his own message head and tail prompt strings (like he can in the IceTERM program).

The user currently exits from this mode by typing a ‘CTRL-Z’ character, this may change as it can interfere with communications from UNIX machines.

21.7 ctrl

This is a modified version of the ‘**alias**’ command. This version allows the user or sysop to define single character commands that occur as soon as the user presses a key. The key that the user presses must generate a control character (ie. the user must hold down the CTRL key while typing the key) and letter keys are the only ones that are supported.

The syntax is:

```
ctrl perm|temp user substitute
```

where, ‘**ctrl**’ is the name of the command, followed by either a ‘**perm**’ or ‘**temp**’ keyword to define whether this is a permanent control alias or a temporary one. The ‘**user**’ character (a single ASCII letter character only) is a key that the user will type (while holding down the CTRL key) to trigger the control alias substitution process and the ‘**substitute**’ string is a IceBBS command that is to be substituted for the user command. Note that the ‘**substitute**’ string can contain white space, allowing you to specify any additional parameters that the command might normally take.

Note that any control aliases can only execute one IceBBS command, if you need to execute a series of commands (for example to change to a certain directory and then start an xmodem download of a particular file), you can put all those commands into one menu file and execute the file with a ‘**goto file**’ command in the control alias.

21.8 describe

The ‘**describe**’ command takes the name of a file as its single parameter and then prints any information about the file that the BBS has. This includes: who uploaded it, the date of upload, the length of

46 Using ChangeTaskPri

IceBBS¹ is a program that internally multitasks. Not only does it spawn off multiple copies of parts of itself, but it also runs those tasks at different AmigaDOS priorities. The reason for the differing priorities is to ensure that the areas which really need the CPU get the time needed to get their processing done. Generally the rule of thumb used in understanding this structure is that the interactions with the modem have the highest priority, while those with the user and disks have the lowest priority.

The IceBBS system currently has three levels of task priorities. The actual AmigaDOS priorities used are derived from the priority of the CLI or Shell window that IceBBS is run from. This *root* priority is the highest priority that IceBBS will assign to one of its tasks, IceBBS will also assign priorities that are one and two levels lower than the root priority.

Unless you execute the AmigaDOS command ‘**ChangeTaskPri**’ (or use some other equivalent utility) the root priority of IceBBS will be zero. Hence, there will be IceBBS tasks running in your system at priorities of 0, -1, and -2. If you are also intending to run compute-bound programs (such as a raytracer or an archiving utility like ‘**Lharc**’) you should do a ‘**ChangeTaskPri -3**’ in the CLI or Shell window before running those tasks. Note that the WorkBench runs its programs at priority 1; thus, if you are running IceBBS in the background and do a lot of work with WorkBench you should probably run IceBBS at a priority of 4 (so it uses priorities 4, 3, and 2). This is quite safe to do since IceBBS does no busy waiting.

¹ The same applies to IceTERM, the terminal program for IceBBS

21 The ASCII Menus

When a user elects to run IceBBS in the ASCII menu mode he will be working with a very different system. This interface can be completely configured by the sysop through a system of script-like files. The first script file that is run is the one named on line 10 (see Section 15.9 [Line 10], page 57) of the 'S:IceBBS.BINIT' file.

If you do not want to have any ASCII side to your BBS then see Chapter 19 [IceBBS.NOASCII file], page 69.

For more than the built in XModem file protocol system to work you will need to configure your system to use the XPR protocol libraries. This is explained in Section 21.20 [ASCII protocol], page 80.

The following commands are available in the ASCII mode, most of these can be executed by the user, either directly by typing the command name and certain parameters, or indirectly by typing the name of an alias that the user or sysop has set up.

21.1 Account

Not yet implemented.

21.2 Alias

This command is used to create an alias command string which the user can enter instead of a particular command. The general idea is that the sysop should define the user commands by using the alias command to make the ASCII interface more similar to conventional BBS systems,

There may be multiple aliases defined at any time, including multiple aliases for a single BBS command (like 'quit' and 'logout' for the 'bye' command).

There are two main classes of aliases, permanent and temporary. Permanent aliases last for the entire session once they have been set, while temporary aliases only last until the user changes menus, at which time they are discarded. This allows a set of global commands to be established and local commands to come and go as the user moves around the BBS.

The user may also define his own aliases during the session.

If an alias name is reused it will take precedence over an existing alias (it is added to the head of the alias list). This allows permanent aliases to be temporarily disabled when the user enters a particular menu and then automatically re-enabled when he leaves the menu.

45 Removable Hard Disks

IceBBS has been tested with the SyQuest SQ555 connected to a GVP Series II controller. In this configuration the SyQuest drives function like huge floppies, you can actually remove a cartridge that a user is downloading a file from, do some work on a different cartridge and insert it whenever the BBS needs to read a new chunk of data. Since it takes about 10 seconds to swap a cartridge you would not want to do this often, but it is possible. The same also applies to CDROM drives.

If you are going to have multiple cartridges available to the BBS users then it is recommended you use the 'defer' option of the AmigaDOS 'assign' command. This is discussed in See Chapter 44 [CDROM drives], page 189.

It is not necessary to use the arbitration feature on Syquest drives, unless someone develops an autochanger for them.

20 The IceBBS.FIDOCFG file

This file is used by some of the FidoNet support utilities. It should contain the default FidoNet node number for the BBS. The node number should be in the format:

1:134/92.0

Walnut Creek CDROM
4041 Pike Lane, Suite E
Concord, CA 94520
USA

Phone: (510) 674-0783
Email: info@cdrom.com

44.8.2 Fred Fish

Fred Fish has been publishing the most well known floppy disk based Amiga library for many years now. Recently he cut his first CDROM release called Fresh Fish. He intends to publish a new Fresh Fish CDROM about once a month, and a will also be producing a quarterly and yearly summary CDROM for BBS use.

Amiga Library Services
610 N. Alma School Road, Suite 18
Chandler, AZ 85224-3687
USA

Phone: (602)-917-0917

44.8.3 AMUC

The AMiga Users of Calgary, a non-profit Society dedicated to the Amiga computer, also publishes CDROMs. These are slightly modified images of their BBS file system. Since they run IceBBS software the CDROM is unique in that it contains IceBBS format description files for most of the files on the disk. Their intent is to cut a new CDROM each time their main hard drive fills up. The AMUC2: and AMUC3: disks were released as a two disk set in late December 1994.

AMUC
Box 34230
#19, 1200 - 37th St. S.W.
Calgary, Alta., T3C 3W2
Canada.

Email: info@amuc.mtroyal.ab.ca
FidoNet: AMUC Chairman at 1:134/27.0

19 The S:IceBBS.NOASCII file

This file is used to turn off the ASCII side of the BBS completely. This can be useful if you are running a regular ASCII BBS along side IceBBS; and hence, you do not want to use the built in ASCII BBS.

The contents of the 'IceBBS.NOASCII' file are not important, if this file exists the ASCII BBS will be turned off.

however, it is perfectly possible to use the CDTV as a CDROM drive accessed from another Amiga. The way to do this is to run a software package called ParNet that allows two Amigas to be linked over a cable between their parallel ports and for software on one machine to access the drives of the other machine. As ParNet has transfer speeds up to 20K bytes per second a CDTV could be used as a CDROM drive for a BBS machine in this way. As a CDTV sells for just about nothing these days this may not be as strange as it sounds.

44.3 CD32 and AmigaDOS 3.1

Commodore's CD32 product should be able to act as an external CDROM drive, in much the same way as the CDTV soon. At least one expansion product is planned for it that gives it the parallel port needed to run ParNet. Also the CD32 does contain a special high speed serial port that is apparently burried within the controller connector, so this might be another way of interfacing it.

AmigaDOS 3.1 contains the same CDROM file system that CD32 uses. So if you have upgraded to 3.1 you may not want to use another file system. Reports are that the third party CDROM solutions are more versatile (there are a lot of odd quirks with the ISO9660 format) than the Commodore CDROM file system, so if you find you are having problems with some CDROMs under 3.1 you might want to try another CDROM file system.

44.4 CDROM-FS

Canadian Prototype Replicas makes a product called the CDROM File System. This is a handler and a library that are installed in your system that allow AmigaDOS to read CDROMs written in the Hi-Sierra and ISO-9660 formats. AMUC has used³ this product quite extensively and has found it to work quite well appart from the odd enforcer hit that occurs when the software is used with a GVP Series II controller and a Sony drive. If you are using either the GVP controller or a Sony CDROM you probably want to use the Xetec product instead.

44.5 Xetec

Xetec makes a very nice CDROM file system. Normally they bundle it with sales of a CHINON drive; however, they also sell it separately

³ This was in 1992, so things may have improved.

18 The S:IceBBS.ARBITRATE file

IceBBS includes some optional access arbitration for CDROM drives to improve performance on single drives when two users are doing directories on different areas of the disk and on jukebox drives (like the Pioneer DRM-600 series) where several users are accessing different disks at the same time. This arbitration also allows one to reduce the CDROM software's buffers to the minimum if you are tight on RAM¹ without affecting performance much (and it allows one to reduce the number of buffers the send file server pre-reads from disk to a smaller amount, see Section 15.14 [Line 15], page 58). To add access arbitration to a particular device you add the name of that device to a file called: 'S:IceBBS.ARBITRATE', for example on AMUC Express we have a Pioneer DRM-600 and DRM-602 (both are 6-disk changers), so this file contains:

```
CD0:
CD1:
CD2:
CD3:
CD4:
CD5:
CD6:
CD7:
CD8:
CD9:
CD10:
CD11:
```

where CD0: to CD5: are in the DRM-600 and CD6: to CD11: are in the DRM-602.

The arbitration file is optional, if you do not need or want arbitration then this file does not even have to exist.

44 CDROM Drives

IceBBS can be used with CDROM drives so long as they are running a handler that makes them accessible as regular AmigaDOS drives. There are a few points to make note of here.

- You should use an AmigaDOS assigned directory for the root directory to the CDROM drive in the 'S:IceBBS.DIRS' file. This assigned name should start with an underscore character (see Chapter 16 [IceBBS.DIRS file], page 63) so that directory listings are read faster. The underscore tells the BBS not to open each file looking for the short description information, normally this will be fine since most CDROMs do not have IceBBS format files on them¹. For example, include the following in your startup-sequence:

```
assign _FishDisks: cd0:fishdisks
```

and then include '_FishDisks:' in the 'S:IceBBS.DIRS' file.

- Some of the available CDROM disks are not 100% public domain (or freely redistributable) in content. You should choose your root directories on the disk with care to avoid giving the world access to copyrighted material,
- Since the CDROM is a read only device the download counters will not work on its files and you will not be able to attach file descriptions to files on it. The 'bulksplit' (see Section 33.5.1 [BulkSplit], page 139) tool can be used to help with this,
- The regular 'access' command cannot be used on CDROM drives (because it cannot store it 'IceBBS.ACCESS' files on them); however, there is a replacement command: 'hashaccess' (see Section 33.3.8 [HashAccess], page 138) that stores the access control files on a writable disk some where else.

44.1 Using deferred assignment

If you are running AmigaDOS 2.0 or higher you can make use of the new 'defer' capability of the 'assign' command. This allows you to have more CDROMs than you have drives. The 'defer' option causes the assignment to wait until the CDROM is actually in the drive, this way you don't have to swap disks several times every time you reboot. For example, here is an excerpt from my 'user-startup' file:

```
assign _Library: DESKTOPLIB1: defer
assign _FishOnCD: FishMarket:Fishies defer
assign _GIFS: Gif_Galore1:gifs defer
```

¹ Which in the case of both AsimCDFS and Xetec might save you quite a bit.

¹ The AMUC CDROMs do have IceBBS format description files.

17 The S:IceBBS.HAIL File

This file (which must be kept in the ‘S:’ directory) is sent out the serial line when a modem connection is established (after a short delay) but before the error-correcting protocol is started. This file is intended to be used to:

- indicate to the caller that he has indeed reached a BBS (perhaps including the BBS name, access times and policy information),
- indicate to the caller that this is a special BBS that will require special software to make full use of the BBS,
- provide some information to the caller as to where he can get support from (perhaps another BBS or a voice number),
- and to indicate to the user how to go about starting the ASCII mode in case he does not yet have IceTERM.

When editing this file you will probably want to make sure that each line ends with both a CR and a LF. Most terminal programs on the Amiga don’t really care about this but programs for the PC do.

43 Format of Account Files

IceBBS account files are all stored in one directory the location of which is specified in the ‘S:IceBBS.BINIT’ file on, see Section 15.4 [Line 4], page 56. There is one account file per user name in the system. Each file is given a name that is derived from the user’s name by doing a 32 bit CRC checksum of the user’s name. This checksum is then converted into a signed decimal number and is used as the file name. With such a system the BBS can find the account information on any person very quickly; however, there is a small probability that two different names can produce the same code. If this happens the second person to try to log in under that name will appear to supply the wrong password and will be forced to change his name slightly to make it unique from the BBS’s viewpoint.

16 The S:IceBBS.DIRS File

This file (which is required to be in your ‘S:’ directory) is used by IceBBS to determine which directories you are going to allow users to download files from. You may list any number of directories here, by providing complete paths to each directory and separating the directory names by *white space* (ie. spaces, tabs or new lines). The directories may be on different devices, may even overlap and logical assigns can be used.

Access to the root directories can be restricted, as well any subdirectories within them can have different access levels set, see the ‘**access**’ command (see Section 33.3.7 [Access], page 137) for more information.

Examples of valid directory names are:

```
DH0:
DH2:BBSFiles/Graphics
BBS_files:
```

One small patch has been added to the file system, this is: the file system will not attempt to send short file descriptions for any file within a directory tree who’s name in the ‘S:IceBBS.DIRS’ file starts with an underscore (‘_’) character. This patch was added to speed directory scans on CDROMs, otherwise large directories can take a very long time to scan¹.

42 IceTERM icon problems

From time to time users encounter problems with the tool types stored in their IceTERM program’s icons. The icon for the terminal will usually be located in the same directory as the terminal program, however, CLI users may place it in their ‘S:’ or the ‘C:’ directories.

The terminal program also supports project icons (as well as tool icons). It will read the tool types first from the tool icon and then from the project icon, allowing the user to set defaults in the tool icon and then using customized project icons to do different things (like to call different IceBBS systems using different dialing scripts).

One occasional problem that your users may encounter is that for some reason (perhaps they have upgraded from an older IceTERM) some of the buttons in one of the windows are missing or are cut off. If this happens it is usually caused by an old snapshot tooltype in their IceTERM icon. Have them delete all the snapshot tooltypes and reconfigure their window layout.

A related problem that can occur when a user upgrades from an old version of IceTERM is that the snapshot menu stops functioning. This is because the new version of IceTERM no longer adds new tool types to the icon each time snapshot is used, instead it looks to see if the tool type already exists and if it does it updates the first copy it finds. The simplest way to get the problem fixed is to just delete all tool type duplicates.

Users quite often come across a bug in the AmigaDOS 1.3 WorkBench in which two tool types keep getting merged into one when the user attempts to edit them. The only known cure for this is to get a copy of a more recent version of the ‘**icon.library**’ and ‘**info.library**’. If a user says that this did not fix the problem, tell him to reboot the system using a WorkBench with those libraries on it.

¹ The reason for this is that IceBBS stores the short file description (and other information) in a small header that is prepended to the file, so for each file in the directory the BBS is scanning it must open the file and read the first chunk of it to find the short description. Since most CDROM drives are very slow at seeking from the place the directory information is stored to the place the start of the file is stored this information can take quite a long time to read (long by comparison to a regular hard drive).

4800 seconds (1 hour and 20 minutes), this is the zone mail hour for Calgary with an extra 10 minutes of padding time on either end since my clock sometimes drifts... If a user tries to log in during this time he will receive the ‘`linenotopen`’ message from the BBS and then be logged off in about 30 seconds. If the line is available all the time just set these two numbers to -1 and -1.

The third number (in this case) is a mode flag indicating that the BBS should open the serial port in shared mode and should leave watching the modem, answering, baud rate adjustment, and disconnecting up to another program such as ‘`TrapDoor`’ or ‘`WPL`’. The BBS will remain dormant on this line until it receives a special message at the message port ‘`IceBBS2`’ (the next field in the line). The name of this port is not important, just so long as it is unique. This special wake up message is sent by the ‘`BBSNOW`’ program which has the syntax:

```
BBSNOW portname
```

if you are using ‘`TrapDoor`’ as the front end in this mode add the following lines to your ‘`TrapDoor.cfg`’ file:

```
SHARED
BBSMODE SPAWN
BBSCOMMAND "dh2:bbs/bbs/bbsnow IceBBS2"
```

Note that if you are running several copies of ‘`TrapDoor`’ then each should use a different BBS port name, and those names should appear on the corresponding lines in the ‘`IceBBS.BINIT`’ file.

The ‘`BBSNOW`’ program acts by sending the necessary message to the appropriate BBS port and then waits for the message be be replied to, when the user logs out then IceBBS will reply to the message and the ‘`BBSNOW`’ program returns.

15.16.4 Example line 18 (polling script)

Line 18 is an example of the older method of networking by running a polling script file every so often. This is still supported (and may be necessary to get USENET). In this case there are 6 fields at the end of the line:

```
-1 -1 1800 3600 900 s:usenet_new
```

The first two are the BBS shutdown times. These are not usually necessary for a polling based system, but can be used. In this example the line is available all the time as indicated by the ‘-1 -1’.

The next number is the offset into the day at which the script file will be first run (ie. at 00:30), the next is the time delay before running the script file a second time (in this case 3600 seconds—one hour).

keep transmission times low you should keep the sizes of the miniature pictures down to the 100x100 mark.

41.6 Edituser Door

This is a door for remote sysops to maintain the user accounts with. It provides a nice MUI based user interface so that one can just click through the account files. It also provides *account templates* so that common account types and changes can be made at the click of a button.

41.7 CircleMUD Door

This door provides a user interface to the ‘`CircleMUD`’ program. It is essentially a reimplementaion of the ‘`TELNET`’ program, that is provided as part of the ‘`CircleMUD`’ distribution, for the IceBBS environment. As such, this door allows users to play in the MUD while performing other actions on the BBS.

41.8 ARexx Doors

IceBBS also supports doors that are written in ARexx. These doors are somewhat easier to write than the regular doors which must be coded in C. The ‘`RexxDoorKit`’ archive contains the instructions, examples and a simulator program⁴.

41.9 Internet FTP

41.9.1 What is FTP?

FTP is File Transfer Protocol, a system for transferring files between computers on the Internet. It is very popular because there are a lot of big *anonymous FTP sites* on the Internet. An anonymous FTP site is like a big BBS that allows you to log onto it and download files without having to register for an account. The IceFTP door allows the user of IceTERM to explore this side of the Internet from his Amiga without having to set up software such as AmiTCP and PPP or SLIP.

⁴ The simulator allows one to test doors without even running an IceBBS or IceTERM.

15.15 Line 16

Line sixteen is used to specify the command to run to send email into the internet (see Chapter 30 [UUCP], page 113). Normally this is something like:

```
uucp_c:sendmail
```

but it might be different if you have some special processing to do. Note that if your BBS is not running UUCP you should replace this line with a `~` character like:

```
~
```

so that the BBS knows to warn the user that the Send Internet button in TERM does not do anything useful.

15.16 Line 17 and beyond

The demo version of IceBBS will only use line 17, any other lines will be ignored.

All the remaining lines specify how each of the modems and serial ports are configured for the system. There is one line for each serial port and there may be as many lines as you have the RAM and CPU to support. Each of these configuration lines may take three different forms depending on the additional functions of the line. The first eight fields are the same for all three configurations.

15.16.1 The first eleven fields

The first eleven fields are (in order from left to right):

- the name of a directory into which user uploads from this line will be placed (for multi-line systems you can use different directories for each line if you wish),
- the name of the serial port device that is to be used for this line ('siosbx.device' for ASDG's Dual Serial Board),
- the unit number (on the selected serial port) that this line is to use,
- the baud rate that is to be used by default, if this is a fixed baud rate modem (ie. MNP, V.32, or HST) this is the only baud rate used, if this is a normal modem then this is the baud rate that will be used when the modem is reset,
- this is a single character (either a 'n' or a '7') that specifies the type of handshaking that is to be used. The 'n' is for no handshaking (this will be fine for normal 2400 baud modems). The '7' is for seven wire (RTS/CTS) connections, which should be used with high speed

Currently there are several doors available for use with IceBBS and IceTERM. These are: 'preview' the IFF file previewer, 'remfile' the remote sysop's file maintenance tool, 'eu' the remote sysop's user editor, the voting door, the 'IceFTP' door, the 'IceIRC' door, 'GPCheess' a chess game, and the 'CircleMUD' multi-user dungeon adventure. Additional doors will be created in the future.

Additional programming information, including example source code for a working client and server, is provided on the distribution disk in the various 'toolkit' archives.

41.4 The RemFile Door

The 'remfile' door provides a co-sysop with the tools needed to maintain the file system. This tool is provided as a safer alternative to the remote sysop shell function. It provides the user with the ability to move, rename and delete files; to edit and add descriptions; and to test most types of file archives.

This door will need more stack than most since it will be running other AmigaDOS functions (like the archiver utilities) so its 'STACK' tooltype should be set to about 10000 bytes.

Since you will not normally be wanting to allow general users to have access to a tool that can allow them to damage the file system (by deleting files) you will want to pay particular attention to the 'ACCESS' tooltype. Remember the syntax is:

```
ACCESS=group|level
```

so do not do something like:

```
ACCESS=10|0
```

which would give everyone who has access level zero or higher in group 10 permission to use the tool, since *everyone* has access level zero in *every* group on the BBS this would give everyone access to the tool.

The 'remfile' door has one unique tooltype: 'ARC=' which you use to define the archivers co-sysops can use to manipulate files and what operations they can perform with them. The syntax for this tooltype is:

```
ARC=archiver|test|list|extract|
```

where 'archiver' is the name of the archiver that will be run. The 'test' string is the archiver's command line option for testing archive files. The 'list' string is the archiver's command line option for listing the contents of an archive. And the 'extract' string is the archivers command line option (or options) for extracting files from the archive. Note that you may wish to include multiple options here so that the

15.6 Lines 6 and 7

The sixth and seventh lines contain the names of the two files in which the *feedback* (see Chapter 35 [Mail], page 161) message base will be kept. The file names should include the full path to them. For more information on the format of these files see the *Message Bases* chapter, in particular the *Old Style – Feedback* section. These two files do not have to exist when the BBS is started, the first user postings to them will create them.

15.7 Line 8

Use this line to specify the name of your IceBBS system. This name is going to be used to create a subdirectory on the user's system to hold messages that come from (or go to) your BBS system. For this reason the name is limited to 31 characters and should not contain any characters which will confuse AmigaDOS—especially spaces and punctuation.

15.8 Line 9

The ninth line specifies the number of lines of monitor window history that is to be kept in RAM on the BBS machine. This information is sent out to the user whenever he opens a new Monitor window, or when the sysop runs the 'monitor' program (see Chapter 37 [Monitor], page 171) on his machine. Note that it might be a good idea to keep this buffer relatively small (say 25 lines) since it can take a while to transmit to users with 1200 baud modems.

15.9 Line 10

This line holds the name of the first ASCII menu file that the BBS is to run when a user logs into the BBS and requests an ASCII mode session. See Chapter 21 [ASCII menus], page 73 for more information on this.

15.10 Line 11

The eleventh line specifies the directory that is used to store the menus for the ASCII mode of IceBBS. This should be a full directory name including the AmigaDOS path to the directory.

41 The Doors System

41.1 Introduction

IceBBS supports a unique and powerful system of expansion called the Doors System¹ (or just doors for short). This system has been designed so additional clients and servers may be added to both the BBS and IceTERM ends of the communications link by just dragging icons and editing a few tool types.

The idea behind a IceBBS door is that there are two programs communicating transparently through the BBS and terminal packages. The connection that the BBS and terminal provide for the door is error free and transparently multiplexed in with other BBS and terminal functions. The user can start up and treat doors in the same way he treats any other BBS function.

The two programs involved in a door are called the client and server. (So now you can say you've used a Client-Server architecture). The client always runs on the terminal end of the connection and the server always runs on the BBS end.

Installation of a door (from the point of view of the BBS or terminal software) is quite simple, the user just drags the icon for the door client or server and drops it in the drawer where these are stored on his system. This applies to both the BBS and terminal ends. The directory where the door servers are stored on the BBS end must be identified in the 'S:IceBBS.BINIT' file, see Section 15.13 [Line 14], page 58.

41.2 Icon Tooltypes

When the user of IceTERM clicks on his 'Doors' button a door chooser is started, this will then scan the user's door directory and look for files which have icons attached. It then opens each icon in turn looking for ones with the appropriate tool types. These tooltypes are:

```
STACK
PRIORITY
MENUNAME
SERVER
ACCESS
```

¹ Other BBS systems provide expansion via doors, IceBBS is unique in that a single user can run multiple doors at the same time and he uses other built in BBS features.

15 The S:IceBBS.BINIT File

This file (which is required to be in your ‘S:’ directory) is used by IceBBS to configure its serial lines and to specify the various special directories and files that IceBBS requires. This is a line oriented file, so each line in it has a fixed purpose. There can be a variable number of serial port configuration lines, and these appear at the end of the file to allow for this. Spaces are only allowed on the serial port configuration lines to separate the multiple fields on them.

Here is an example of a ‘IceBBS.BINIT’ file for a three line system, the various lines are explained below¹:

```

1: dh2:bbsfiles/UserLogFile
2: 150
3: dh2:ErrorMessages
4: dh2:UserAccounts
5: dh2:UserMail
6: dh2:sigs_messages
7: dh2:sigs_headers
8: NameOfYourBBS
9: 200
10: FirstAsciiMenuFile
11: dh2:ascii_menus
12: dh1:AccessDir
13: dh1:DescDir
14: dh2:BBSDoors
15: 100
16: uucp_c:sendmail
17: dh2:receive2 serial.device 3 2400 n auto REL +
    ATZ AT dh2:line2 284-5625 10200 4800 -2 IceBBS2
18: dh2:receive3 serial.device 4 19200 7 fixed XYZZY +
    ATZ AT dh2:line3 284-2048 -1 -1 1800 3600 900 +
    s:usenet_new
19: dh2:receive1 serial.device 2 9600 7 fixed ARQ +
    ATZ AT dh2:line1 282-5171 -1 -1 -1
```

The sections that follow describe the use of the various lines in the above example.

15.1 Line 1

This line specifies the name of a file to which IceBBS will write a user access record. This file will be updated quite often—at least four times

¹ The line numbers are not part of the file, and ‘+’ characters indicate the line has been split at that point.

40 The Remote DOS Shell

An AmigaDOS shell, through which a sysop can work remotely using AmigaDOS commands, is included in IceBBS. In order for this shell to function the additional handler ‘VAUX-Handler’ must have been copied¹ into your ‘L:’ directory and the small mountlist for it should be added to the ‘mountlist’ file in ‘DEVS:’. Here is the mountlist entry for the handler:

```

VAUX:      Handler = L:vaux-Handler
           Stacksize = 8000
           Priority = 5
           GlobVec = -1

#
```

It may not need the full 8000 bytes of stack so if you are really worried about RAM you can try a smaller amount.

Before IceBBS is started you should execute a command similar to:

```
mount VAUX:
```

perhaps in the user startup script. If this is not done then no amount of clicking on the “Shell” button in IceTERM is ever going to bring up a DOS shell.

Before the BBS starts up a DOS shell it will check three things, first it will see if the user has level 255 access in all the security groups on the BBS (see Section 7.5 [DOS shell access], page 35), second it will check to see if the user’s account has the DOS shell bit enabled and third it will see if the file: ‘S:IceBBS.SHELLACCESS’ is present (it does not have to contain anything in particular, it just has to exist). If neither of these three conditions is met the shell will not be started.

One additional thing to note, since there is no way to click on gadgets in the windows that some commands may bring up upon being run from the CLI it is a good idea to run a program that will cancel requesters. This will prevent a shell from being locked up waiting for someone to insert a disk or hit the cancel button.

¹ AmigaDOS 2.0 users can always use the multiple assign feature to leave it in the IceBBS area.

and higher allow more access levels, the default is 16 but may be overridden by the sysop, by editing the value stored in 'S:IceBBSMAXACCESS' file (see Chapter 10 [IceBBS.MAXACCESS file], page 43).

39 News Bulletins

IceBBS (and the more recent IceTERM programs) supports a news bulletin facility that allows the sysop to place a number of reference text files on the system in an easily retrievable format.

39.1 The S:IceBBS.NEWS File

The sysop creates a file called 'S:IceBBS.NEWS' which contains a number of lines, each line contains two strings of characters, for example the file might contain:

```
dh2:news/newhere "Info for new users"
dh2:news/versions "Current software versions"
dh2:news/wishlist "Plans for future features"
```

The first string on each line is the name of the file which will be sent to the user. The second string is the text describing the file, this text will be placed in the menu that appears on the user's screen when she hits the 'News' button in IceTERM.

39.2 News files

Each news file is written in the same way as the error message files. The important thing to note is that the first line must contain four numbers, the top left corner's position (from the left and top of the screen) and the width and height of the window.

```
120 20 410 260
        So you are new here?
```

```
Hit the Chat button to talk to the sysop
(if he is around and paying attention).
```

```
Updates to all IceBBS related software are
kept in the IceBBS: directory, hit Get Files
to see this.
```

```
Files that have arrived in the last month
or so are kept in Files:@NewFiles.
```

```
The SIGs system is configured to only send
you a few SIGs by default, to get more SIGs
you must use the Subscribe button within
the Sigs window of IceTERM.
```

14 The S:IceBBS.NEWSIGS File

This file takes the place of the ‘S:IceBBS.SIGS’ file and serves two purposes: locating the files in which the SIGs are stored and providing the access control to them.

14.1 New SIGs file format

This file contains the locations of the various directories on your system where the SIG files are stored. It also contains the names that you wish to give those SIGs on the user’s machine. The third thing stored in this file is a status for each SIG, this is currently not used. The last things on each line are the access levels needed to subscribe to that SIG followed by two numbers which define who gets moderator access.

The file format is very simple, for example:

EEmail	dh1:testsig	man	0	0	0	0	1	255
Test_SIG	dh1:sigs/test	man	1	1	1	1	1	255
Amiga	df1:amiga	auto	0	0	0	0	2	1
General	dh1:sigs/general	auto	255	255	0	255	1	255
News	dh2:bbsStuff/news	man	7	2	4	1	3	0

Each line contains the specification of one SIG. Each line has a number of space separated fields. The first field is the name of the SIG as it will appear on the user’s machine—this name will become the name of a file¹ on the user’s machine—so keep it free of strange characters that will confuse AmigaDOS.

The second field is the full path specification of the directory on the BBS machine that will contain the two files (the link and data files) that will contain the messages and replies in this SIG.

Note that these fields cannot contain any spaces so you cannot use spaces in the names of the directories or the SIGs on your system.

The third field is the selector which specifies which SIGs the user must manually subscribe to before he receives any messages from them. This is currently unused, it is a remanant of the old SIGs system. This field used to contain either: ‘man’ (for manual mode) or ‘auto’ (for automatic mode). The automatic mode SIGs will be sent to all users until they create a ‘MySigs’ file. The manual mode SIGs will only be sent to users who have a ‘MySigs’ file and include the name of those SIGs in the file. Again, this field is not currently used, but something needs to be present in its place so include either ‘auto’ or ‘man’ as in the above example.

The remaining numbers (except for the last two) are the access levels that the user is required to have to see and download from the SIG.

38 Conferencing

IceBBS supports a simple multi-user conferencing feature.

38.1 Configuring

This facility works from a local memory buffer on the BBS machine. Through the ‘S:IceBBS.BINIT’ file the sysop configures the maximum number of lines of text that will be stored in the BBS history buffer for the conference (see Section 15.2 [Line 2], page 56). When a user connects to the conference all of these lines will be sent to him, and the BBS will add a line announcing his arrival and will send a copy of it to all users who are in the conference. Since all this information is sent to the user it is usually a good idea to keep this to a reasonable number (say below 200).

On the user’s machine there is also a history buffer (which he sets the depth of). This allows him to enter the conference as soon as he logs into the system and then go and do something else, secure in the knowledge that he will not miss anything if he is too busy to swap back into the conference.

When a user sends a message to the conference (up to about 200 characters of text) that message gets entered into the BBS conference history buffer (and if the buffer is full the oldest message is deleted). The message is then sent to all users who are currently on-line. This ensures that all users see the messages in exactly the same order. When a user logs out of the conference the BBS will send a good bye message to the conference so others know he is gone.

38.2 The Confer Command

This is a sysop utility that communicates with IceBBS through a named message port and allows the sysop to participate in conferences. It provides the same user window and message window interface that the terminal program does. It is provided as a separate utility to save space in IceBBS and to allow for easier upgrading.

The program examines its icon tooltypes for some configuration information. It looks for the icon first in the current directory, then in ‘S:’ and then in ‘C:’. The tooltypes are:

CONFHISTORY=
sets the maximum number of lines of scroll back in the conference window,

¹ The name will have either ‘.LNK’ or ‘.DAT’ appended to it.

13 The S:IceBBS.SIGS File

This file is no longer needed, it was used to control access to the old SIGs system that was available in pre-53 versions of IceBBS.

37 Monitor

This program implements an activity monitor window (similar to the one the IceTERM program provides) that the sysop can run on his Work-Bench. The program takes a number of default configuration values first from the icon¹ and then from the command line. The icon tooltypes are:

MONHISTORY=

This specifies the number of events that are to be kept in the local scroll back buffer,

LEFTEDGE=

This specifies the position of the left edge of the window when it opens, this is in pixels,

TOPEDGE=

This specifies the position of the top edge of the window when it opens, this is in pixels,

WIDTH=

This specifies the width of the window when it opens, this is in pixels,

HEIGHT=

This specifies the height of the window when it opens, this is in pixels.

The command can also pick up these parameters from the command line with command line switches, any number of these may appear in any order.

-bNNN

The number of lines of events to maintain in the scroll back history buffer,

-xNNN

The initial horizontal position of window,

-yNNN

The initial vertical position of window,

-wNNN

The initial width of window,

-hNNN

The initial height of window.

¹ As usual the program looks first for the icon in the current directory, then in 'S:' and then in 'C:'.

12 The S:TRIM.LASTTRIM File

This file is written and maintained by the ‘**trim**’ utility (see Section 34.2.8.1 [Trimming SIGs], page 160), which is responsible for deleting old messages from the SIGs message bases. If ‘**trim**’ refuses to run you might want to delete this file, but in normal circumstances there is no need to touch or create this file.

36.5.3 Forward

This command forwards email from one user to another, this is useful when deleting an old account and moving unread mail to the new account. The command syntax is:

```
forward from to [-rdirname] [-d] item1 [item2]
```

Where:

from	is the name of the user we are forwarding from,
to	is the name of the user we are forwarding to,
item1	is the email message number we are forwarding other items may also be forwarded but are optional,
-r	use this directory for the email (override ‘S:IceBBS.BINIT’),
-d	delete the file after copying it to the destination account.

36.5.4 Send

The ‘**send**’ command has been removed, it was present because the ‘**email**’ command could not be run without a window opening, and hence, could not be used by remote sysops. The ‘**email**’ command has been fixed to prevent this from happening.

11 The S:IceBBS.MSGID File

The file 'S:IceBBS.MSGID' stores a single long integer containing the next message identification (MSGID) sequence number to be allocated. If the file does not exist it will be automatically created and seeded with the current time. When a message is entered into the SIGs (by batch upload, via the online reader or via the OLR in sysop mode) it will be assigned the current MSGID and the contents of 'S:IceBBS.MSGID' will then be incremented by 1. The MSGID is stored in 'vmmsg.Expansion[4]' (see Section 34.2.5 [VMSG struct], page 149) of the message. When a message is exported by 'export' or 'uuexport' this stored message ID will be printed into the appropriate area of the message, if the 'Expansion[4]' is 'NULL' (ie. an older message) then it will be assigned a fresh message ID at that time. MSGIDs have also been added to net mail (FidoNet email) via the new version of 'email2cut' and the updated 'fidopost' program.

Again, creation and maintenance of this file is completely automatic you do not have to do anything.

```
checkmail "The User Name"
```

If the user name is not provided then 'checkmail' will default to the name set by the 'USER=' tool type found in its icon⁸. This allows 'checkmail' to be run from an icon. If the 'USER=' tool type is not found then 'checkmail' defaults to 'Sysop'.

'Checkmail' exits with a return code of 0 if no mail is waiting and a return code of 5 if there is mail waiting, so that it can be used in AmigaDOS script files to set the 'WARN' condition when mail is waiting for a particular user.

36.5.2 Email

This utility allows the sysop to post and read email to and from users. It also handles sending and receiving files. It takes its input from both the icon (stored in the current directory, 'S:' or 'C:'), the command line and the 'S:IceBBS.BINIT' file.

The icon tooltypes are searched for first and are:

- USER= Is used to specify the user name you wish to use as sysop, if not specified it will default to 'Sysop',
- EDITOR= Is used to specify the name of the editor you wish to use to edit replies to messages with, if not specified it will default to 'Ed',
- TEMPDIR= Is used to specify the directory where you will allow this program to create temporary files, if not specified it will default to 'RAM:', this is needed so that the editor can be used to create new messages or edit replies,
- FILEDIR= Is used to specify the directory where you wish received files to be placed, if not specified it will default to 'RAM:'.

The program determines the directory in which user mail directories are to be created from the 'S:IceBBS.BINIT' file. The command syntax is:

```
email [-flag1] [-flag2] ... [file] [user1] [user2] ...
```

where you can specify a number of command line flags followed by the name of the file (either a file to be sent or a file containing a message to be sent) and a list of user names to which the message or file is to be sent. Note that if the file name or a user name contains a space character you must enclose it in quotation marks. The command line flags are:

⁸ which it searches for in the current directory first, then the 'S:' and then the 'C:' directories.

10 The S:IceBBS.MAXACCESS File

This file is written by the sysop to set the number of independant access groups that he wants to have on the BBS. It should contain a single number (for example: '126'), and if the file is not present the BBS will default to having 16 access groups. This file is accessed by a number of tools, including 'edituser' (see Section 25.1 [Edituser], page 95) and 'access' (see Section 33.3.7 [Access], page 137).

In older versions of IceTERM the user list used to be requested automatically; however, when the user lists got large this proved to cause too many problems for users with low speed modems so this feature was redesigned. Also the older email lists were more bulky because there was unused space within them, this space was intended for storing a public encryption key, but since the 'PGP' program does this so well the reserved space has been removed.

Note that you should not make the email storage directory publically visable to the file system, as this will allow other users to download email.

36.3 Between BBSs via FidoNet

The IceTERM program's email facility can also be used by a user to send, receive and reply⁵ to FidoNet netmail messages. As far as the user is concerned these are much the same thing as mail between users on the same BBS, with two exceptions: he must type in the name and FidoNet address of the user he is sending mail to (for replies this is done for him). In order for the BBS to supply this function two support programs must be run at various times of the day.

The first of these is the 'email2cut' program (see Section 29.5.8 [Exporting netmail], page 111) which converts the messages from IceBBS users into FidoNet messages. This program scans a IceBBS outbound mail directory (specified in the 'S:IceBBS.BINIT' file, see Section 15.16.1 [Fields 1 to 11], page 59) and converts any messages that it finds into appropriately named FidoNet '*.CUT' message files. This program should be run in your FidoNet script before the mailer program⁶ is used to call other BBSs.

The second program this is needed is the 'import' program (see Section 29.5.2 [Import FidoNet], page 108). This program will take the FidoNet messages that have been received from other systems and unpack them and deliver them to the appropriate user mail directories (creating directories as needed). This program should be run after the FidoNet mailer has been run. Note that this program no longer places email into the user's email directory, it now writes it to the email SIG.

'Outmail' is also needed to process the contents of the email SIG and forward on any new messages it finds there to the appropriate destination on the Internet or FidoNet.

⁵ If the BBS runs the 'email2cut' program users will be able to send FidoNet mail this way still; however, any replies to their mail will appear only in the email SIG as that is where the new version of the 'import' command puts them now.

⁶ TrapDoor or WPL.

9 The S:IceBBS.SHELLACCESS File

As an added precaution against sysops setting up an IceBBS system and forgetting to set the access control to the DOS shell function correctly the 'S:IceBBS.SHELLACCESS' file was added. If this file does not exist then no one will be able to start the DOS shell.

In order to enable the DOS shell function you need to create a file called 'S:IceBBS.SHELLACCESS' (it does not have to contain anything in particular) and then set up the additional access controls (see Chapter 40 [DOS shell], page 177).

36 Electronic Mail

The BBS supports four modes of electronic mail (email) between users, these are message and file exchange between users on a single IceBBS system, message exchange between users on different systems connected by FidoNet, and message exchange between users on different systems connected via the Internet.

36.1 Email directories

The message and file exchange between users on the single IceBBS system is fully self-contained¹, no additional software is needed. This system works by creating a private mail directory for each user the first time he receives a letter. These private mail directories are given the same name as the user's account file² and are all stored in the system email directory (see Chapter 15 [IceBBS.BINIT file], page 55). When a user logs into the BBS the server will check his mail directory and send him a message (see Chapter 22 [Error messages], page 85), if there is waiting mail.

Similarly, when a user receives a file via email the system will place that file into a private directory³, creating the directory if it does not already exist. When the user logs into the BBS a message will be sent to him if he has any files waiting. The user can download these files at any time (resume is also supported in case there is line trouble) by using the Get Files tool in IceTERM and selecting the special '**Emailed files...**' directory⁴. Emailed files stay on the system until the user uses the Delete File button in Get Files. Note that directory downloads of the user's email directory are also allowed, and that these are resumable.

The sysop can create a special type of emailed file, using the '**email**' utility, that will be sent to a user automatically as soon as he opens the email tool. This is a one-time transfer, if the user cancels the transfer or breaks the connection before it is complete the file will be deleted and the transmission cannot be resumed. This is only useful for junk mail

¹ This is the old email system, it is being phased out in favour of the email SIG system, but it will continue to be supported for a while until the email SIG is more convenient to use.

² See the '**edituser**' command.

³ This directory has the same name as the email letters directory, except it has an '**F**' appended to it.

⁴ This magic directory name is automatically mapped to each users private mail directory, so there is no way for a user to download files from another user's directory.

8 The IceBBS.ACCESS Files

These files are written by the ‘access’ command (see Section 33.3.7 [Access], page 137) to set the access levels that the user must have to see and download files from the directories the access files are located in. The file format is quite simple, just an array of bytes, where each byte is one group’s access level. The number of bytes that are written to each file depends on the number stored in the ‘S:IceBBS.MAXACCESS’ file, and if that file is not present it will default to 16.

For example if you have a user with the following access levels: ‘2 3 0 0’, the following table shows what happens when he attempts to access directory with any of the following access control settings:

‘0 0 0 0’	The user gets access. This is the default access level for a directory that does not have access controls. Remember though, for a user to see the contents of subdirectories he must first get past the access controls in the parent directories.
‘4 4 1 0’	The user gets access because of the last zero. Remember that all users have at least access level zero.
‘2 5 5 5’	The user gets access because of the first level, the fact that he does not have enough access in the other three groups does not matter, he only needs sufficient access in one group.
‘1 5 5 5’	The user gets access, his 2 in the first group is more than the access requirement.
‘3 5 5 5’	The user does not get access. The 3 in the first group exceeds the 2 that the user has.
‘255 255 255 255’	The user does not get access. In fact he would have to be a full sysop (with DOS shell access) to get in here.

35 Mail

So that sysop’s can read the messages that are posted in the old style (feedback) message base without having to log into their own systems a utility called ‘mail’ has been provided. This program runs from the CLI (and should also work from the WorkBench) it provides the same user interface that the users of the terminal program get, without having to connect through the BBS. It keeps track of when you last used it by updating the last read message number in the account file for the sysop.

The ‘mail’ command gets its configuraton information from its icon¹ and the command line. The mail icon needs to have the following tool types defined:

USER	This is used to specify the sysop’s name (which will be placed in replies and postings) and also for access to account file so that the last message read counter can be saved between sessions. Note that this can be any character string that will fit into 63 characters. If an account does not exist for the given name one will be created by ‘mail’ when it is run,
PASSWORD	This may be any sequence of characters up to 63 long. The reason this is needed is so that an account can be created if it is needed,
EDITOR	This is used to specify the editor to be run when the sysop wants to post a new message or compose a reply to an old one. Due to the AmigaDOS ‘Execute()’ funciton ² the editor must reside in the ‘C:’ directory. If no editor is specified the ‘Ed’ editor will be used,
TEMPDIR	This is used to specify a directory where the ‘mail’ program can safely create temporary files. The ‘mail’ utility will create temporary files when the sysop attempts to post new messages or reply to existing ones. If this is not specified then the default is taken to be ‘T:’,
KEEPFILE	This is used to specify the name (and path to) a file into which any messages the sysop decides to keep will be placed.

Note that any additional tooltypes (as might appear in a regular terminal program icon) will be ignored, so it is safe to just make a mail icon by copying an existing terminal icon.

The command line switches the ‘mail’ command understands are:

¹ The program will search the current directory, then ‘S:’ and then ‘C:’ to find the icon.

² This probably does not apply when running AmigaDOS 2.0.

The fifth form of privilege control is through the ability to restrict the maximum number of bytes a user can download in any one day. You may disable this feature if you wish.

7.8 Time Limits

IceBBS also includes a time bank and hourly session limit facility by which you can moderate usage of the BBS. This facility is based on four time bank parameters plus the twenty-four hourly limits per user.

The way accounts work is as follows. Each user has a time bank, in which any unused time accumulates. There is a per-user value that sets the upper limit to the amount of time the user can accumulate in his bank. There is a daily allocation amount that sets the amount of seconds that will be added to the user's time bank at midnight. (If a user does not log in for 3 days he will get 3 times the daily amount added by the next time he logs in).

When a user logs in, the system first determines the session limit based on the time of day he connected at, it then checks to see if that limit exceeds the amount of time the user has in his bank account. If it does, then IceBBS resets the session limit down to the amount remaining in the bank. IceBBS then checks the limit against the courtesy time limit, if the revised limit is less than the courtesy amount the system increases the session limit to the courtesy amount.

The system then checks the number of times the user has called in the last day, and if the user has called more than his allowable maximum the system sets his session limit to 30 seconds (so that he can see the warning message before being logged off).

Finally the system will check to see if there is a FidoNet mail hour session occurring within the users session limit, and if there is it will reduce the users limit appropriately.

All this is managed on a per-user basis by the 'edituser' command. The time bank facility allows users to save up for a big download and it also allows you to give special users (perhaps paying ones) extra credits, either as a lump sum or as an extra daily amount. The hourly session limits allow you to adjust peoples' access time based on the time of day their connections take place to make the BBS easier to connect to during peak hours by reducing session limits during those times.

7.9 Account Defaults

When a new account is created by a user logging in with a name that the BBS does not recognize the BBS attempts to find an account

34.2.6.4 Finding the end of file

The following function positions both files to their logical (not necessarily physical) ends so that the next writes to the file will lay down a valid new message. It returns the offset into the data file ('.dat') at which writing will take place.

```
ULONG SeekNextFree(BPTR lnk,
                   struct FirstLinkItem *firstlink,
                   BPTR dat)
{
    struct LinkItem local;
    struct DataItem data;

    /* find the last message's link, and
       advance over it so the next write
       will follow it.
    */
    FindMessageNo(lnk, &local, firstlink->Last,
                  firstlink);

    /* now seek into the data file to look for
       the current last message's data. Skip over
       that so that we are ready to write new data
       to the data file.
    */
    Seek(dat, local.Offset, OFFSET_BEGINING);
    Read(dat, &data, SIZE(DataItem));
    Seek(dat, data.Length, OFFSET_CURRENT);
    return(local.Offset + data.Length
           + SIZE(DataItem));
}
```

34.2.7 The Sysop's Message Reader

In order to make this system work the user is going to require a rather different kind of message reader. This is provided in the off-line message reading utility called IceOLR. The same program is used by both regular users to read messages they download via the 'BatchSigs' function of IceTERM and by sysops to read the same message bases the BBS server and support utilities maintain. This program is documented in a separate manual and AmigaGuide document.

'1 1 0 0 2' This gives a member access to all the doors except the 'RemFiles' door,

'1 1 0 0 9' Again, this gives a member access to all the doors except the 'RemFiles' door,

'1 1 0 0 10' This gives a member access to all the doors including the 'RemFiles' door,

'1 1 0 1 0' This gives a member sysop status, and hence gives him access to the 'RemFiles' door, but he does not get access to either the voting door or the GPChess door. He does have access to the IFF picture previewer since that only needs a level zero access in group 4.

'1 1 0 1 1' This gives a member sysop status (he can use 'RemFiles'), and he can use all the other doors except the GPChess door.

Remember, groups are independant, so if a user does not have enough access in one group but does in another he will be granted access, he does not have to have sufficient access in all the groups to use a certain function.

7.4 How many security groups?

The file 'S:IceBBS.MAXACCESS' can be used to change the default number of groups to anything you want. Note that each group takes at least one byte per user in the system, so keep things reasonable.

If this file is not present then the BBS will default to using only 16 access groups. If the file is present then it should contain a number in plain text form. This number will be used to set the number of access groups on the BBS. For example if the file contains:

6

then the BBS will have only six different access control groups (which would be sufficient for the AMUC Express examples above). If the file contained:

221

then there would be 221 access control groups on the BBS.

7.5 DOS Shell access

A special access level is required to be able to use the remote DOS shell. This is level 255, and it must be set in every single group. The reason for this is that as soon as someone is given DOS shell access they

```

        /* the item we found was less than the one
           we are searching for, so we must move
           the lower bound up to this point.
        */
        lower = target;
    }
    else
    {
        /* the item we found was above the one we
           are searching for, so in this case we
           want to bring the upper bound down
        */
        upper = target;
    }
}

/* we could have the case where lower and
   upper are the same, so we need to take a
   look at what's there, actually as we look
   at both the upper and lower bounds at the
   start we should never get here...
*/

Seek(file, lower
      * SIZE(LinkItem), OFFSET_BEGINING);
if (SIZE(LinkItem)
    == Read(file, link, SIZE(LinkItem)))
{
    if (link->MsgNo == MsgNo)
    {
        /* got it! */
        return(TRUE);
    }
}
return(FALSE);
}

```

34.2.6.2 Finding the next message

The following function is very similar to the previous function, except it is used to find the *next* message after a particular date, this can be used to find the first new message since the user last used sigs, and

7 Access Control

The demo version uses a fixed set of pre-set account settings which cannot be changed.

The topic of access control includes several things: actual security issues relating to the file and SIGs systems (see Section 14.3 [SIG access control], page 52), system privilege issues and access time limits.

Access control is not included as part of the demonstration version of IceBBS.

Security on the IceBBS system is provided in a form that is quite customizable. A number of the functions of the BBS have access control including: the file system, the SIGs message bases, the doors and the remote DOS shell. As well, the individual functions of the main control panel can be disabled on a user by user basis and each user's time and process quota's can be individually controlled.

7.1 Access levels

The access control system is built around a small array of access levels for each user. The individual levels in this array range from 0 to 255, if a system function is set to level zero it will be completely open to all users, and at level 255 that function is accessible to only your most privileged users. By default the BBS will have 16 different security groups (numbered 0, 1, 2, ... 15), within each group users can be assigned access levels from 0 to 255.

For example, suppose you have four groups of users:¹ free users, members, executive, and sysops. Each group is allowed access to different areas of the BBS. The free users get the least access while the sysops have access to just about everything. You might decide to use the first security group for free access, the second for member access and so on. Since there are four different user groups you still have 12 different security groups unused. If we ignore the 12 unused groups (they are all set to zero) then the four different access settings (groups 0, 1, 2, and 3) look like:

'1 0 0 0'	This is the setting for the free users,
'1 1 0 0'	This gives members all the rights of free users plus some member specific functions,
'1 1 1 0'	This gives the executive access to everything the members and free users get, plus perhaps something more,
'1 1 1 1'	This gives sysops access to everything.

```

Seek(file, upper
      * SIZE(LinkItem), OFFSET_BEGINING);
if (SIZE(LinkItem)
    != Read(file, link, SIZE(LinkItem)))
{
    /* no need to adjust seek position */
    return(FALSE);
}

if (link->MsgNo == MsgNo)
    return(TRUE); /* found it */
if (link->MsgNo < MsgNo)
    return(FALSE); /* requested message too new */

/* now we have valid start and end positions
   we need to search
   */

```

¹ This is based on the AMUC Express BBS.

at the same time at full speed, this is very useful for IceBBS because it allows a user to send and receive data at full speed at the same time⁷.

6.8 V.34 Modems

These are the newest and fastest modems on the market. For a while now a number of manufacturers have followed ZyXel's lead and started producing modems that initially ran at 16.8K baud, then 19.2K and now 21K. These are interim modulation designs that have seen some use in the long distance FidoNet scene. The international standard for the next generation of high speed modems (beyond V.32bis) is called V.34. This standard allows for modulations as high as 28.8K baud. Because this standard took so long to set a number of manufacturers sold pre-V.34 modems, these are usually termed V.fast or V.fc (fast class). Now that true V34 modems are widely available one should avoid getting a v.fc type modem, as these often have problems connecting to V34 modems.

The main problem with V34 communications is not the modem, it is the serial ports driving the modem. The internal serial port is not very suitable for the 38.4K or 56K baud connection that these modems are suited for, the ASDG card will loose bytes at these speeds, and the A2232 card cannot run faster than 19.2K baud. However, it looks like at least the Multiface and ComPorts cards should do a fine job at these speeds.

6.9 V.42, V.42bis and MNP Modems

V.42 is an error correcting protocol that runs between a pair of modems, it is very similar to MNP level 4. V.42bis is an extension to V.42 that adds compression to the communications channel, thereby increasing the throughput. It is similar to MNP level 5 except that it is somewhat smarter, it will detect when compressed data is being sent and get out of the way unlike MNP 5. Since V.42 and MNP are both protocols they can be used on any physical link, this is why you get 2400, 9600, 14400 and now 28800 baud modems using them.

Manufacturers have been making some grandiose claims about the gains these compression schemes will make, the claim for MNP 5 is typically a factor of 2 and the claim for V.42bis is usually a factor of 4. In reality, the only time you are going to see these levels of compression is when you are sending straight uppercase text, typically 30-40% compression gains are what you should expect to see.

⁷ Without the line turn-around delay that will occur with HST modems.

thus allowing the moderator to change his mind. Messages are actually only deleted by a special tool that the sysop runs when he feels the need.

Also this scheme allows FidoNet and USENET messages to be easily imported into a IceBBS message base, and the replys and new postings extracted for transmission.

34.2.6 Traversing links

To traverse the linkage file for a particular SIG I use a binary search technique on the '.lnk' file to find the message number of interest. If you are reading the messages in the order of posting then you can just keep reading the next 'LinkItem' from the file; however, if you want to follow the thread linkages you will have to search somehow.

The following section contains some of the code that can be used to search for messages.

34.2.6.1 Finding a message by number

This function is used to locate a particular 'LinkItem' by message number, it seeks to the right spot, and reads in the 'LinkItem'. It knows that message number 1 is the FirstLinkItem. It returns 'TRUE' if the requested ID was found.

The first link item is used to help in the bisection process. If it is 'NULL' then the routine will read in the extra data it needs but this requires an extra seek and read, so if you are following a chain you might want to pre-read the first link and pass it into the routine.

The routine will position the file so that the next link that is read is the one that follows the specified 'MsgNo' (date), regardless of the function returning 'TRUE' or 'FALSE'.

Note that this function does not look at whether a message is deleted⁴ or not, that is up to the caller.

```
int FindMessageNo(BPTR file,
                  struct LinkItem *link,
                  ULONG MsgNo,
                  struct FirstLinkItem *fli)
{
    struct FirstLinkItem localfli;
    ULONG lower, upper, target;
```

⁴ When a message gets marked as deleted it will still remain in place, only when the message base is trimmed will it actually disappear from the message base.

support V.32bis (which gives you 14.4K in BOTH directions at the same time) as well as HST protocol, and the price differential (on the USR sysop program) is now quite small.

6.5 USR Dual Standard Modem

For the USR Dual Standard Modems (very popular in the FidoNet world) the suggested settings are:

- AT&B1 This will set the modem so that it does not change the speed of the serial connection between it and the computer,
- AT&H1 This enables hardware handshaking, do not even think of trying to run without this at speeds over 2400 baud,
- AT&A3 This enables a reasonable level of connection type reporting about the protocol that is used,
- AT&X6 This selects a reasonable level of connection speed and status reporting,
- AT&R2 This turns on the other half of hardware flow control, I am not certain this is actually needed (it might only apply to synchronous mode connections) but it can't hurt,
- AT&K3 This sets the modem to negotiate for the *best* method of compression. If the connection is a V42 link then you always want to enable the V42bis compression scheme. If the connection is an MNP link then you want the MNP5 compression scheme if you are transferring a lot of text; however, you don't want the MNP5 compression scheme if you are doing a lot of compressed file transfers since it may actually expand the files a bit reducing throughput. For this reason the '&K3' setting will enable V42bis if the caller requests it, but will refuse an MNP5 connection if it is requested. If you have high volume SIGs on your BBS you might want to use '&K1' and leave the choice up to your callers.

6.6 SupraFAXModem V32bis

We have used a pair of these modems on the AMUC Express BBS for over four years now. These settings seem to do the job, but there might be some other changes needed. The only problem I experienced was that the modems want to return the connection reports on multiple lines rather than as a single string like all other modems do. There is an S-register (number 95) that fixes this, if you execute the following commands in the order they are given you should be fine.

- CreationDate the date at which this message was created, if the message came from an outside network then a date in the network header would have been extracted and placed here. If the message was posted by a batch SIG upload then the date is the date at which the user used IceOLR to create the message. The date is in conventional C-library (ctime) format,
- Author the name of the author of this message, stored as a null terminated string in a fixed length 64 byte field,
- DelBy the date this message should be deleted by, it may not physically get deleted at that date, but the server will stop showing it to users then. Again, stored in seconds since the Amiga clock began,
- ReadCount the number of times the message has been read, not currently updated,
- Archive Various bits to indicate to a secondary utility whether to set aside this message at a later date in an archive area because someone of authority feels it is useful. You need co-moderator or higher access to set these bits. Not implemented yet,
- Expansion some fields for expansion purposes, currently only Expansion[4] is being used, this is used to store the CRC of the 'Message-ID' string of USENET messages (this is used by 'newsin', see Section 30.3.1 [NewsIn], page 114, to reconstruct the thread linkages between messages as they arrive), the same functionality will be added to the FidoNet 'import' utility too, see Section 29.5.2 [Import FidoNet], page 108. For messages that originate locally this field is used to store a unique message ID number generated from the 'S:IceBBS.MSGID' file, see Chapter 11 [IceBBS.MSGID file], page 45.

The following six fields all indicate the length (in bytes) of each of the following fields. These fields are concatenated together between the message header structure and the message body. So to get to the message you need to seek past the fixed header and then do an additional advance of:

Subject+Address+ReplyTo+OldSubject
+Keywords+Network

bytes, to skip over the variable length fields.

- AT&C1** This makes the modem tell the computer the true state of the carrier tone, IceBBS needs this setting so that it can tell when a user has broken the connection. If this setting is not used the modem will not be reset for about 7 minutes after the user has broken the connection (or the protocol may not start),
- AT&D3** This makes the modem do a hardware reset when the BBS figures out that the line has been dropped. Note that early versions of the Commodore A2232 card software had some problems with this signal line, they still work (I use one) because IceBBS also sends the modem a '+++ ATZ' sequence to help reset the line,
- AT&R2** The modem will use the CTS and RTS line,
- AT&S1** The DSR line is used according to RS232 specifications,
- ATS0=2** This tells the modem to answer the phone and establish the connection on its own. IceBBS uses this method rather than polling the Ring Indicator pin like some BBS packages do because polling is a bad thing. Note that I have set my system to answer on the second ring because if you answer on the first ring the phone may not have rung on the caller's end and his modem may not respond correctly.⁶

Advanced modems may have several register banks to save custom settings in to. The commands to select a particular bank (on a GVC MNP modem) are 'AT&Y0' to use bank 0, and 'AT&Y1' to use bank 1. You will need to decide which bank you are going to use and then store the modem settings into that bank by using the appropriate 'AT&W0' or 'AT&W1' command and also set the active bank by using the appropriate 'AT&Y0' or 'AT&Y1' command.

6.3 MNP Modems

After having applied the previous modem configuration commands you are now ready to define how your modem will behave in the MNP world. The strategy here is to set up your modem so that it will try to establish a reliable and compressed line (REL/COMP) with the caller's modem and if that fails it will automatically fall back to a non-MNP mode. To do this use the following commands:

⁶ I have seen this happen with an old Packard Bell 2400 baud modem and I have had a number of people experience this with various modems when trying to connect to the US Robotics Dual Standard modems which answer the phone on the first ring.

```
#define LI_KILLED (1L << 2L) /* message is
                                marked as killed by the
                                user's kill filter, the
                                user's reader knows not
                                to allow replies to this
                                message. */
#define LI_PERMANENT (1L << 3L) /* this is a
                                permanen| message, do not del */
#define LI_LOCAL (1L << 4L) /* this is a
                                message that originated
                                locally, so should be
                                considered for export */

};
```

The two CRC values are used so that the SIG server (on the BBS) can quickly kill filter the user's requests. These are 32 bit CRCs computed by first upper casing the string in question (not including the null byte terminator) and the using the same hash function that the user account file names are computed from.

34.2.4 The DataItem structure

Each 'VMSG' in the data file is preceeded by a 'DataItem' structure, this structure serves to identify the type and length of the data block that follows. The 'DataItem' structure is:

```
struct DataItem
{
    ULONG Type; /* the type of this message,
                (usually uncompressed)
                */
#define MSGTYPE_PLAIN 0 /* unpacked message */
#define MSGTYPE_PP 1 /* powerpacker packed message */

    ULONG Length; /* the length of this item (the
                  number of bytes that follow)
                  */
};
```

34.2.5 The VMSG structure

The 'VMSG' blocks are virtually unchanged from the old SIGs system, except in the old system each message used to be stored in a separate file.

The message file header block contains:

6 Modem Register Settings

There are a number of types of modem on the market today. The main distinguishing features are: price, quality, speed and special protocols. IceBBS supports most modems on the market today.

6.1 Old style modems

These are the modems which you specify with an ‘auto’ baud setting in the ‘S:IceBBS.BINIT’ file. This type of modem does not support MNP, nor does it support a fixed baud rate link between the computer and the modem. Often these modems are described as being *Hayes compatible*. To set up such a modem for use with IceBBS the easiest thing to do is to restore the factory default settings by executing the ‘AT&F’ command and then send the following commands:

ATL0	This selects the lowest volume for the modem’s speaker,
ATM0	This turns the modems speaker off at all times,
ATQ0	This turns on result codes, IceBBS needs these so it can see when someone has connected to the modem,
ATV1	This makes the result codes be the English text strings,
ATX4	This makes the modem report all events to us, although the only one that IceBBS looks for at the moment is the CONNECT string,
AT&C1	This makes the modem tell the computer the true state of the carrier tone, IceBBS needs this setting so that it can tell when a user has broken the connection. If this setting is not used the modem will not be reset for about 7 minutes after the user has broken the connection (or the protocol may not start),
AT&D3	This makes the modem do a hardware reset when the BBS figures out that the line has been dropped. Note that early versions of the Commodore A2232 card software had some problems with this signal line, they still work (I use one) because IceBBS also sends the modem a ‘+++ ATZ’ sequence ¹ to help reset the line,
ATS0=2	This tells the modem to answer the phone ² and establish the connection on its own. IceBBS uses this method rather

¹ This is not the best thing to always do, a sysop configurable modem reset string will be added in a future version

² This may change in the future.

```
{
    ULONG One;      /* always the number 1L */
    ULONG First;    /* this first message number
                     present in this sig */
    ULONG Last;     /* the last message number
                     present in this sig */
    ULONG Next;     /* next message at this level
                     in the tree, this will
                     in fact be the first root
                     level posting in this sig.
                     */
    ULONG Prev;     /* previous message at this
                     level in the tree, this will
                     be the most recent root
                     level posting in the sig.
                     (the Last message is truly
                     the last message posted
                     to the sig, but it might
                     be a reply to another message)
                     */
    ULONG Number;   /* the number of messages that
                     are currently in the sig
                     NOT including the FirstLinkItem
                     (it does not count) also NOT
                     including any empty space left
                     behind by the TRIM utility at
                     the end of the file.
                     */
    ULONG Reserved1; /* set to NULL */
    ULONG Reserved2; /* set to NULL */
    ULONG Reserved3; /* set to NULL */
};
```

34.2.3 The LinkItem structure

The following is the structure that each SIG’s linkage file is composed of, these provide the linkages between messages to support the message threading functions and the dates for scanning purposes.

```
struct LinkItem
```


5 Required Libraries

The following libraries and devices (apart from your particular serial port devices) are required for IceBBS to run. If IceBBS does not startup check that these are in your ‘LIBS:’ directory.

```
LIBS:ARP.library
LIBS:translator.library
LIBS:icon.library
DEVS:narrator.device
DEVS:serial.device
```

or whatever alternative serial device you have specified in
‘S:IceBBS.BINIT’

```
L:VAUX-Handler
```

Note that IceBBS will startup without the ‘`translator.library`’ and ‘`narrator.device`’ being present in the system, in which case users will be unable to use the ‘Talk’ function. The ‘VAUX:’ device also needs to be mounted. The demo version of IceBBS does not use the ‘VAUX:’ handler.

count for you (or use the ‘`edituser -p -iu`’ command),

- If you want to check what you just posted, use the ‘`edituser`’ utility and reset your last message read to zero.

34.2 New Style: SIGs

This section describes the format, storage and retrieval methodology of IceBBS special interest groups (SIGs). During the life of IceBBS three types of message bases have been tried. The first was the feedback system, which is still in use. The second is now referred to as the *old SIGs* system³, this is no longer in use. The third system is the *new SIGs* system which is described in this section.

The bulk of email is also handled here. See the commands relating to it in the FidoNet (see Section 29.1 [FidoNet], page 105) and UUCP (see Chapter 30 [UUCP], page 113) chapters. There are a few unique points about the email SIG, these will be raised where appropriate. The most important point is that the email SIG must always be called *email*, so it will always occupy the files:

```
email.dat
email.lnk
```

34.2.1 The Storage and Retrieval Methodology

Each SIG is stored in two files, a linkage file (suffix ‘`.lnk`’) and a data file (suffix ‘`.dat`’). The data file is a concatenation of the same ‘VMSG’ format messages which used to be stored in individual files under the old system (the VMSG format is documented later). There is an additional header that is added to record the length of each message and the type, this allows for future storage of messages in compressed form (although that can be done right now by using the XPK compressed file system).

The messages are stored in the data file called, it will be named ‘`signame.dat`’ where ‘`signame`’ is the name of the SIG. The data file contains:

```
a struct DataItem
a VMSG
a struct DataItem
a VMSG
and so on...
```

³ There were several user utilities for this system: ‘`read`’, ‘`age`’, ‘`VBBSReader`’, and ‘`GPRReader`’, all of which are now obsolete.

and all will work nicely. You can even use several IceTERMs at once if you need to experiment with multiplayer type games.

34 Message Bases

IceBBS implements two types of messages bases; a very simple one (the old style one) capable of only supporting one SIG and a more complex one capable of supporting many SIGs. The simple message base is called the ‘Feedback’ section in IceTERM¹ and the complex message base is accessed through the ‘Sigs’ and ‘BatchSig’ buttons in IceTERM. Version 54 and on of IceBBS add private email to the SIGs system, allowing for batch downloading and off line creation and reading of email.

34.1 Old Style: Feedback

When the user of IceTERM hits the *Feedback* button IceTERM starts a message reader and then sends a command to the BBS telling it to send all the new messages. When this command is received by the BBS a server is started which loads all the new data and adds it to the transmission queue. The client end of feedback which is running on IceTERM will receive this continuous stream of data and print it on the users display as he hits the return and space keys. Occasionally the user will send back fresh postings which the feedback server running on the BBS will add to its files.

For the moment the server will automatically stream all new data to the user as soon as the server starts, so any other things arriving from the user are fresh postings. Postings currently have a header containing the time and date of posting and the user’s name. All postings are stored as two files, a header file containing fixed length records and a data file which the header file indexes into. The records in the header file² are the following structure:

```
struct MsgHeader
{
    long messagenumber;
    long messagelen;
    long fileposn;
    long time;
    long replyto;
    UBYTE username[64], topic[64];
}
```

¹ Currently feedback is only available from the menu attached to IceTERM’s control panel. Eventually the feedback system may be removed completely, but since it does not need any configuration by the user it will stay for a while yet.

² This is subject to change without notice at the moment, so don’t write any utilities for it.

4.5 ComPorts 8 port card

This board contains 8 serial ports which can be run up to 56K baud. It is in use on two IceBBS systems, including AMUC Express which runs 7 phone lines and one direct null modem connection. This is made by:

AMIGO Business Computers
192 Laurel Road,
E. Northport,
NY 11731, USA
Phone: (516) 757-7234

4.6 Applied Engineering's Internal Modem

This is an internal modem card, it also has send fax capabilities and with an upgrade can be made to receive fax. I have had two users use this with the terminal program, but I have not tried it with IceBBS.

If you get one of these cards make certain you have the most recent versions of the ROMS on it. The earlier 1.0 and 1.1 ROMS did not handle multicharacter I/O and would not run the IceBBS protocol.

Curiously enough these cards support MNP but do not have any means of flow control¹⁰ between the card and the computer. As a result you cannot really make much use of the MNP feature since if you run the card faster than 2400 baud (to use the MNP compression) lost data occasionally results.

In general I would advise against using internal modems on the Amiga, it seems that the normal serial card and external modem is a more reliable (and ultimately more flexible and less expensive) route to take.

4.7 Serial Cable Requirements

Serial cables used with IceBBS must pass the carrier detect line (CD) and the Data Set Ready (DSR) line. IceBBS can also use the RTS/CTS flow control (seven wire protocol). IceBBS does not use the ring indicator (RI) line. The easiest way of providing the correct cabling is to use a standard 25 pin cable, or in the case of the ASDG board a standard 9 pin cable (often called an AT Serial Cable). The cables provided with the A2232 card supply the correct hookups too.

¹⁰ As strange as this sounds one of my IceBBS users has actually called AE several times about this (as he got new versions of the ROMS).

Furthermore, since the path to the files is going to be hard coded into the descriptions you want to use a special logical (say 'BBSFiles:') so that you can move the raw files to a different drive or subdirectory at some later date.

The command to do this is:

```
BulkSplit dh2:bbs/files dh3:rawfiles +
          dh2:bbs/desc BBSFiles:
```

once the command has been run the root directory file⁴ must be changed to point at the 'DH2:bbs/desc' directory rather than the 'DH2:bbs/files' directory and the following assign must be made:

```
Assign BBSFiles: DH3:rawfiles
```

so the indirection mechanism can find the actual files.

When 'bulksplit' encounters a file that does not have a IceBBS header it creates a default header in the description tree for that file, this way *all* the files can be accessed from the description file tree.

It also writes a 'FILES.BBS' and '00_INDEX.TXT' file in each directory of the file tree so that users of non-IceBBS systems can benefit from the short descriptions.

It also generates warning messages about any directory names which exceed 8 characters, since these (and their subdirectories) cannot be accessed on MSDOS machines with CDROMs.

It also ignores any *indirected IceBBS files*, ie. if you place a IceBBS description header containing an indirection to a file somewhere else (perhaps on a CDROM) these files will be filtered out.

'Bulksplit' also takes the '-k' command line parameter, if this is present then it will not delete the source files as the copying takes place. This will mean that you end up with over twice the disk space in use but you still have the *master* copy around which can be useful if you have the extra drive space...

⁴ 'S:IceBBS.DIRS'

one of these cards to run at a maximum baud rate of 38.4K by removing the second divide by two circuit from the ACIA chip's clock line.

The following message, which appeared in the AMY_TECH newsgroup gives an alternate way of doing this:

Author: Murray Rivett
 Creation Date: Mon Nov 28 09:22:00 1994
 Address: Matthew Mathers
 Subject: Re: C= 2232 at speeds >19200

In a message dated 22 Nov 94 21:05:44, Matthew Mathers wrote:

I was wondering if there is any way to make the Commodore 2232 serial card transfer data at a faster rate than 19200? Is there an updated disk with the 2232 drivers on it that allows it? I have a 25MHz 68040 3000T so I know the bus and processor aren't the slowdown. Any help would be greatly appreciated.

Yes, I have two A2232's at 38,400. The clock for 1.8432 Mhz needs to be changed to 3.6864 Mhz. The serial driver should be 33.13 from Wb2.1 and is the same driver as on the Wb3.1 Install disk.

Set all software that uses the board to 38,400 and the driver will happily comply. This works well to speed up ANSI and text and I get about 3600 cps with a Supra14.4Fax modem. The board has worked at full speed (38,400) with sending out one port and receiving on another.

This info I gleaned on Internet about 2 years ago from George Robbins, an engineer from C= at that time. He also thought 4x was possible but I was unable to get my modems to talk properly to the card so went back to 2x the base frequency of 1.8 Mhz.

By the way, one of the pins is very hard to unsolder without overheating everything so take care and use patience.

To summarize, this is a good card to use on a multiline BBS which only has a few high speed lines.

other unforeseen software problem with the A2232 serial driver as a result of this even if the hardware modification does seem to work. Still it might be an interesting and useful project; infact this serial card has a lot of potential for hacking.

out what's new on the system by just opening the 'NewFiles:' directory and showing the files sorted by date.

33.5 The Bulk Programs

These are some programs I wrote to make converting a IceBBS file tree into a CDROM for general use easier. They are also useful to regular sysops who want to place a variety of CDROMs online.

The problem with a IceBBS file system is that both the BBS file information and the AmigaDOS file are stored in one single file. This makes the file useless to a regular person or a sysop who is not using IceBBS. To work around this limitation, while also gaining some additional benefits for IceBBS sysops an additional capability was added to the IceBBS file system. This is file indirection, whereby a IceBBS file containing only the decription information can be shown to the IceTERM user, and when that user selects the file for download the BBS then looks up the *actual* name of the file to be sent to the user from within the description file. The advantage this gives the IceBBS sysop is that the AmigaDOS files can be stored on CDROM while the description files (which are a lot smaller) can be placed on a regular hard drive. This makes directory scanning a lot faster, and it allows the download counters to be updated (which they couldn't have been if the file and description were on the CDROM).

33.5.1 BulkSplit

The 'bulksplit' command is used to take a directory tree containing IceBBS format files and split those files into their header and AmigaDOS file parts and store the resulting files in two separate directory trees: one for headers and one for files.

When 'bulksplit' encounters a file that does not have a IceBBS header it creates a default header in the description tree for that file, this way *all* the files can be accessed from the description file tree.

There is also an additional mode that can be used to scan a directory tree (such as an existing CDROM) which only contains AmigaDOS files and thus create header files for that directory tree. This mechanism will also recognize certain special files, such as 'FILES.BBS', '00_INDEX.TXT' and '00_FILE.LST' and automatically extract the descriptions from those files and place them into the header files that get created.

The splitter will also create 'FILES.BBS' and '00_INDEX.TXT' files in the destination directory tree and it will print warning messages whenever it encounters a directory name longer than 8 characters (as these cause problems with MSDOS systems).

4 Serial Ports

IceBBS will work with most serial ports available today. It was developed using the internal serial port, the Commodore A2232 7 port board, the ASDG Dual Serial board and the ComPorts 8 port card. It has also been run with the Multiface II and III boards.

4.1 Internal Serial Port

The internal Amiga serial port has an important hardware limitation; it will only buffer a single byte for transmit or receive. This lack of buffering means that the Amiga must service an interrupt for every single character that is transferred. There is a considerable software overhead that results from this; so much that the system is strained to the limit to run at 19.2K baud. With an accelerated Amiga it is possible to run at 19.2K baud, but with a normal 68000, multitasking response will suffer severely if you attempt to do this¹. It is also very important to use hardware flow control (RTS/CTS) when running at high baud rates to protect against lost data.

If you are considering running a BBS system with more than one serial line, or you are contemplating using high speed modems it is strongly recommended that you use an expansion serial port board instead of the built in serial port.

4.2 A2232

The Commodore A2232 serial card adds an additional 7 serial lines to the Amiga. It fits in a Zorro II expansion slot and connects to the real world via 7 mini-DIN connectors on the rear card tab. Commodore also supplies 7 cables which go from mini-DIN to standard 25 pin RS-232 connectors, so all you need to get are modems (the 25 pin connectors are the right sex to connect to modems). This card has an on-board 65C02 processor which is clocked at 3.5MHz² (one half of the Amiga's clock speed), 16K of ram (which is shared between the 65C02 and the Amiga) and 7 of the 6551 ACIA serial chips.

Note that there are no ROMS on this card, so the driver code for the 65C02 must be loaded by the Amiga from the serial.device replacement that Commodore supplies. Since these are all standard well-documented

Move and split

move the files in the list to another location, in the process split them into their AmigaDOS part and IceBBS header part,

Copy and split

copy the files in the list to another location, in the process split them into their AmigaDOS part and IceBBS header part.

33.3.6 DelOld

This command is used to delete all the files from a given directory that are older than a specified number of days. The syntax is:

```
DelOld days dir1 [dir2 ...]
```

'days' is the number of days to keep files around. 'dir1' is the name of the directory to delete files from. Multiple directories can be specified, separated by spaces. To specify the current directory use ".".

This command is useful in helping to clean out old files from directories, such as the directory where copies of new files are kept.

33.3.7 Access

The 'access' command is used to set the access control fields for a particular directory. There are a variable number of classes of users (groups of users). You can assign an access level requirement independently for each group of users. Each user's account has the corresponding access levels (see Chapter 7 [Access control], page 33) which you can assign with the 'edituser' program (see Section 25.1 [Edituser], page 95). For a user to see the name of the directory and be able to change directory into it and download files from it at least one of his access group levels must be greater than or equal to one of the corresponding levels assigned by the 'access' command.

This also means that even though you can set a lower access requirement in a subdirectory of some given directory there is no way a user can get to that subdirectory if one of the parent directories he has to pass through first has a higher access requirement. It also means that you can gradually filter out users by increasing the access requirements as they progress into deeper subdirectories.

Access control can even be added to the root directories specified in the 'S:IceBBS.DIRS' file so that if there are entire trees you want to restrict access to, you can.

The syntax of the command is just:

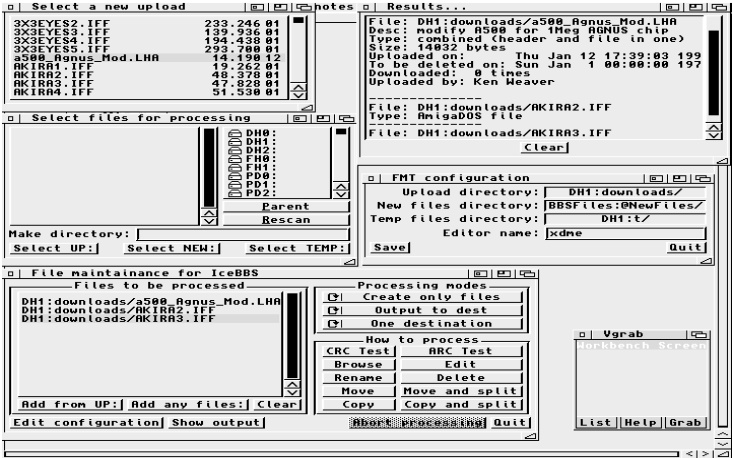
¹ This has been improved in AmigaDOS 2.0

² This is what the schematics included with the documentation indicate; however, it is possible this is only being clocked at about 1.8432MHz.

As IceBBS starts up you should see a burst of disk activity followed by the appearance of a number of line monitor windows on your WorkBench. If IceBBS fails to start, look for missing libraries, low memory, serial devices that are missing or are already in use and double check the 'S:IceBBS.BINIT' file.

33.3.5 FMT

'FMT' is a tool that was written to address the problems of validating, sorting and describing files on an IceBBS system. It uses MUI for its GUI.



The FMT Tool

The above figure shows 'fmt' with many of its sub-windows open. The main control pannel is in the lower left. It has three main areas:

- a list displaying the files that have been selected so far for processing. Files can be removed from this list by clicking on them or by hitting the 'Clear' button below the list. Files can be added to this list from either a simple requester that just displays the contents of you new uploads directory (ie. 'UP:') or from a full file requester. These two requesters appear above the main program window at the left of the picture,
- a set of three cycle gadgets to select the processing mode,
- and a set of ten buttons to actually perform the processing.

On the right side of the picture are two more windows, the top most one is the results window (which is opened by clicking on the 'Show output' button) and the one below it is the configuration window where you can customize the configuration to match your system.

33.3.5.1 Processing modes

The three processing modes cycle gadgets allow you to customize the way most of 'fmt's processing functions work. The options are:

Create only files

when 'fmt' is moving or copying it will not create icons to go with the files it creates,

the directory names cannot have embedded spaces in them. For more information see Chapter 16 [IceBBS.DIRS file], page 63.

Additional access control is provided by setting up access locks at key points in your directory structure. This is done using the ‘access’ tool (see Chapter 7 [Access control], page 33). To give selected users the keys to get past these locks you use the ‘edituser’ command (see Section 25.1 [Edituser], page 95).

3.9 The S:IceBBS.NEWSIGS File

This file contains the locations of the various directories on your system where the SIG⁴ sections for the SIGs tool (in IceTERM) are stored. It also contains the names that you wish to give those SIGs on the user’s machine (these will be the directory names on the user’s machine). The third thing stored in this file is a status for each SIG, this allows certain SIGs not to show up on the user’s machine until he subscribes to them (this is useful for protecting beginners from very high volume SIGs like usenet). The fourth thing stored here is the access control information for each SIG. For more information see Chapter 14 [IceBBS.NEWSIGS file], page 51.

Also see Chapter 7 [Access control], page 33 for more information about how IceBBS security works.

3.10 Setting up the Feedback Section

The current version of IceBBS only supports two forms of message bases, the original very crude message base (called Feedback) and the more advanced SIGs message base.

To set up the feedback message base (see Section 34.1 [Feedback messages], page 143) all that must be done is to edit the ‘S:IceBBS.BINIT’ (see Chapter 15 [IceBBS.BINIT file], page 55) file to include the full name and path of the message text and index files you wish to use, and to post a message or two into the system either with IceTERM (by dialing in) or with the sysop’s reader ‘mail’ (see Chapter 35 [Mail], page 161). It is strongly advised that these two files be on non-volatile media; otherwise, when you crash you will loose your message base. Do not worry about scanning speed if you are using floppies since IceBBS loads all unread

When a user uploads a file to IceBBS he is given the opportunity to provide a long file description. If he declines to do this the file will be sent as a regular AmigaDOS file (with no header). If the long description is provided the file will be prepended with a IceBBS file header on the user’s machine before it is sent to the BBS system. Such a file will arrive with a filled out header and long description already in place.

33.2 Indirected Storage

The IceBBS file system allows the description information to be stored in a different location from the actual file. This was done so that files that are located on CDROMs could have descriptions and working download counters. This also has the advantage that as the header information is located on a hard disk the directory scanning works faster than if the CDROM’s directory is read directly. This is particularly important in multi-line BBS systems which use CDROM drives. And especially in systems which use jukebox type CDROM drives.

Since CDROMs are becoming so important in the BBS world there are a few additional tools which work with file descriptions en-masse. These are the ‘bulksplit’, ‘bulkjoin’ and ‘bulktest’ tools.

The ‘FH_SOMETIMES’ and ‘FH_OFFLINE’ modes of indirect storage have not had any of the supporting software written yet. The idea behind them is that descriptions for every file will always be accessible, but when a user attempts to download a file he will be told that the file will be requested for him and that there will be a delay. The request will be logged to a system file somewhere and a request processing utility which might be run once a day or so will retrieve the file and place a copy of it either in the user’s emailed files directory or in a common requested files directory (which would get aged out every week or so). The most likely solution is to use the user’s emailed files directory since then a new security loop hole does not appear.

33.3 File Tools

These are the primary tools available for manipulating IceBBS format files and control access to areas of the directory tree.

33.3.1 BBSJoin

To allow the sysop to add long descriptions to regular AmigaDOS files the ‘BBSJoin’ utility is provided. This utility will take an existing

⁴ SIG is an abbreviation for Special Interest Group, which in BBS terminology is a place where users discuss topics of mutual interest.

```

ATLO      ;minimum speaker volume
ATMO      ;turn off speaker
ATSO=2    ;answer phone on second ring

```

3.4 Commands

Install the IceBBS program and related commands (see [Command index], page 221) ('Confer', 'Mail', 'BBSplit', 'BBSJoin', 'Edituser' and perhaps IceTERM) into a directory that is within your command search path. You can always just copy these to your 'C:' directory, but it is recommended that you keep them in a separate directory so that they can be more easily replaced when future IceBBS upgrades become available.

To put them in their own directory all you need to do is to make a directory for the commands, copy the commands into it and then use the 'Path' command to add that directory to your CLI or Shell's command search path to make this permanent. Edit the startup-sequence and shell-startup files (found in your 'S:' directory) to include a path command similar to:

```
path dh0:icebbs/bbs_c add
```

to do this.

3.5 Using IceConfig to configure the BBS

The 'IceConfig' utility, shown in the following picture, allows you to edit the majority of the IceBBS configuration files and it even features online help. The help requires that you place the icebbs.guide file in the same directory as the 'IceConfig' utility, then to get help on anything you just position the mouse pointer over the item (in 'IceConfig') and hit the 'HELP' key. 'IceConfig' requires MUI to run³ and the help facility requires that 'AmigaGuide' or 'MultiView' is installed.

³ There is a bug in MUI v2.2 that prevents help from working under AmigaDOS 2.x, but it does work quite nicely under 3.0. This bug is fixed in MUI v2.3.

33 File System

IceBBS stores files for users to download in regular AmigaDOS treed directory structures. The roots of these directories are stored in the 'S:IceBBS.DIRS' file, and any files and subdirectories within those directories are available for the users to download. Before a user can see or enter a directory IceBBS will check inside that directory for a 'IceBBS.ACCESS' file and if one is found the user must meet its access levels. This is one of the access control mechanisms in IceBBS, for more information see Chapter 7 [Access control], page 33.

33.1 Format

IceBBS stores the actual files in two formats: as regular AmigaDOS files and as IceBBS files; although the user only sees them as AmigaDOS files. IceBBS distinguishes between these two file types by looking at the first four characters of the file. If it sees the characters 'VBBS' then it treats the file as a IceBBS file, otherwise it assumes it is an AmigaDOS file.

If the file is an AmigaDOS file then IceBBS sends the 'nofiledesc' error message¹ to the user (which usually says something like: *sorry there is no long file description for this file*), and then will send the full file to the user if he decides to download it anyway.

If the file is a IceBBS file then the BBS loads in the rest of the header of the file (format shown below²) and extracts the long file description and sends it to the user. If the user decides to download the file IceBBS will extract the AmigaDOS portion of the file (starting after the 'LongDesc' and lasting for 'Length' bytes to the user.

```

struct FileHead
{
    ULONG Type;           /* 'VBBS' file type */
    ULONG Length;         /* the length of the actual
                           file */
    ULONG UploadTime;     /* the time at which is was
                           uploaded */
    ULONG ExpiryTime;     /* the time for the automatic
                           expiry */
    USHORT Bits;          /* various mode bits: */

```

¹ From the error messages directory.

² The format of these headers have changed a bit since they were first documented.

3 Quick Start

This is a quick guide to help you get your IceBBS set up quickly. Following this procedure should get you a working IceBBS system. For more detailed information on the various configuration options and sub-systems please refer to the appropriate chapters.

3.1 The Distribution Archive

IceBBS is distributed as an LZH archive file. When this file is unarced it will create an installation script file and number of separate subdirectories:

ReadMeFirst

this file contains important new information,

BBS_C

this directory contains the IceBBS program and the related commands for your 'C:' directory (see [Command index], page 221),

ErrorMessages

this directory contains the various ASCII text files that IceBBS sends to users when system errors and related events occur (see Chapter 22 [Error messages], page 85),

BBS_S

this directory contains five files for your 'S:' directory, there is an example of a script file to handle FidoNet (see Section 29.1 [FidoNet], page 105) and USENET (see Chapter 30 [UUCP], page 113),

ASCII

this directory contains a working set of ASCII menus (see Chapter 21 [ASCII menus], page 73),

L

this directory contains the handler for the remote sysop shell (see Chapter 40 [DOS shell], page 177),

DEVS

this directory contains the mountlist entry for the handler for the remote sysop shell,

DOC

this directory contains updated documentation that has been created or revised since the printing of this manual,

Install

this script file will install IceBBS in a directory that you specify.

```
dh0:databases/dh2database
dh0:databases/dh3database
dh1:special
```

the system will add the appropriate extensions ('.hsh', '.occ', '.fnm', '.cfg') to the names in the 'S:IceBBS.FIND' file when accessing the databases.

32.7.1 Why use these databases?

The database that the find utilities build and use is designed to allow the user to rapidly find all the files in a given directory tree that any word appears in. It is designed for speed. Typically it only takes a single seek in the database to determine the list of all the files that the word you are searching for is located in. Once this list is loaded it takes a couple more seeks per file in that list to determine the full path and actual file name. Thus, on a hard drive it takes less than a second to find the first occurrence and only a couple more seconds to print the list of all occurrences of a given word.

Note: a word is any string of alpha numeric characters, ('a...z' plus '0...9') three to 31 characters long, it is not case-sensitive.

32.7.2 How large are these databases?

The actual size of the databases is adjustable by the sysop when the database is being built. Useful databases are in the range of 1Mbyte to 5Mbyte depending on the size of the directory tree being indexed. For example the database for the software directory of the AMUC CDROM #1 when built with a 100000 word dictionary is a maximum of about 5Megs, by adjusting the cutoff point for words that occur in too many files (some words appeared in over 1000 files) this can be dropped to about 3.5Megs without impairing the searching capability significantly.

32.8 What are the limitations?

Since the system is based on a large hash table there is a chance that two or more words will be represented by the same place in the table. This means that if you search on a single word you might find more files than the word you are looking for actually appears in. However, there will be no other files on the system that contain the word you are looking for that are not in the list. Because of this the system allows you to sharpen your searches by looking for combinations of words (see the '&' and '-' operators). The other factor that affects how many *extraneous files* are reported to contain the file you are looking for is the size of

2.4 Performance

As far as performance is concerned, IceBBS is capable of running 6 to 8 lines on a stock 2000HD at 2400 baud on each line. With an '030 accelerator card in an Amiga you should be able to run 8 lines at 9600 baud³, as I have run 4 lines at 19.2K and 2 lines at 38.4K in a direct wired test. The actual maximum transfer capability depends a bit on what the users are doing since IceBBS will allow users to upload and download files at the same time, and when doing so they can nearly double the load on the system.

2.5 Requirements

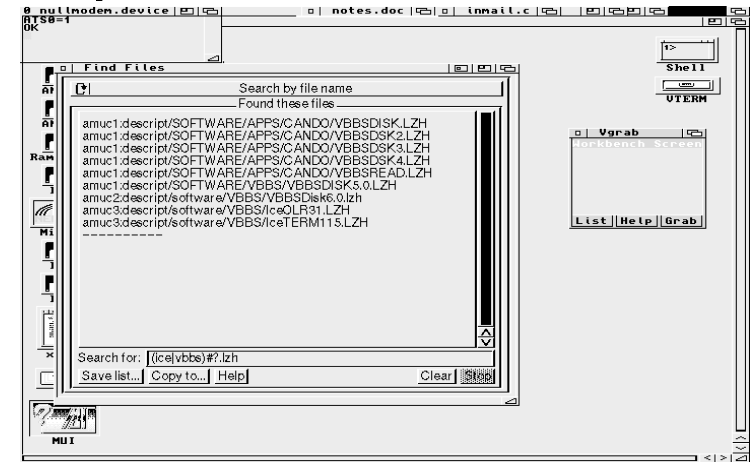
IceBBS will support most serial expansion boards and internal modems. The system has been tested with ASDG's dual serial board (at rates of up to 153.6K baud), Commodore's A2232 card (6 lines at 9600 baud work fine, more than 4 lines at 19.2K appears to choke the card), the Multiface II and III cards, the ComPorts 8 line card.

Note that while IceBBS will work with the internal serial port, using this port at baud rates greater than 9600 baud is not recommended unless you are running with an accelerated Amiga (it does in fact work, but you will find multitasking to be sluggish at times). You will be much better off to get a serial expansion board and to use that instead.

IceBBS has also been run with a variety of modems, including the SupraModem 2400, PackardBell 2400, GVC 2400 with MNP, some Motorola V.32 modems, the USR Dual Standard, Sportster and HST models, and Zyxel modems. It would be nice to hear how it runs with Telebit modems.

As for memory requirements IceBBS and IceTERM are quite frugal. The IceTERM program will just run on a 512K machine, so most of your users will have no memory problems with it. The only time when they can run into trouble is when they attempt to do too many things at

interesting when you start including multiple levels of parenthesis and use the not operator: '~'.



Searching by File Name

32.3 FindScan, the first pass

'FindScan' recursively scans a given directory tree and builds three of the four file finder database files for that tree. Each directory tree has its own database so that one only needs to rebuild the database for the file trees that change. Because of this you need only build a database for a CDROM² once.

In order to maximize the effectiveness of this searching system 'FindScan' will examine and index the *contents* of all the files it encounters including any arc, zoo, zip, lha, compress, gzip or arj archives, which it will unpack to a temporary area and then examine the contents of. The routine will skip binary files such as: exes, gifs, mods etc. If it encounters an IceBBS header it will index the long description in the header and then split the file from the header and index its contents. It also handles the indirected type of IceBBS header, where the header and file are stored separately.

32.4 MySort

This is a simple replacement for the AmigaDOS 'sort' command. The reason I wrote it was I needed a much faster sort (with fewer features) for sorting the large '.key' files that 'findscan' generates. These

³ AMUC Express currently runs 7 lines with one V34, three V32bis, one HST and two 2400 baud modems. The processor is an old GVP'030 25MHz card with 8Megs of RAM installed in an A2000. The serial card is a ComPorts 8 line card. The machine is networked to my development system via an A2065 ethernet card running commercial AmiTCP and Envoy software. About 2Gigs of hard drive and 13 CDROMs are attached to the GVP Series II SCSI card and the Commodore 2091 card. This is about as much as you can get that size of machine to do.

² The AMUC CDROMs come with a pre-built database.

2 Introduction

The products: IceBBS, IceTERM and IceOLR form an *integrated communications environment* for the Amiga user. This is where the term *Ice* comes from. The *I* in *Ice* could also be taken to mean *intuitionized*, but that is rather Amiga specific. Ice is also something that is cool and slick, it can also be dangerous so watch out. The IceBBS was also developed in the *icy* cold of the Canadian Rockies¹, and the first general release of the BBS was made during the winter.

2.1 The Demo Version

The demonstration version of IceBBS is freely distributable. The following are the most important limitations of the demo version:

- no access control is provided, users get as much time as they want and access to all the features,
- no file access restrictions, all users can see and download the full set of files without restriction,
- no remote DOS shell function (without access control this would be very dangerous),
- at most three doors can be in use at once, this allows you to see what a door can do, (actually a user can, in one session, start up to a maximum of three doors)
- no DOS Doors (they require extra support software that is included in the full version only),
- no support utilities, including the user account editor and the Fi-doNet and USENET support utilities.
- only one phone line is supported.

2.2 Features

This is the sysop's manual for IceBBS², an alternative multi-line BBS system for the Amiga. IceBBS was designed with several goals in mind:

¹ Technically in the Canadian equivalent of the Russian steppes. But you can see the Rockies from Calgary, and there is great skiing to be had there.

² This is a preliminary document and I would appreciate any feedback you might have as to content or structure. Also note that there are functions in IceBBS, IceTERM and the support utilities that are still quite fluid and I would appreciate any ideas you might have. Support for this software is available through IceBBS at speeds up to 14400

```
amuc1.cfg
amuc1.fnm
amuc1.hsh
amuc1.key
amuc1.occ
```

the `key` (keywords) file is optional. You need to put these files in one directory, for example `dh1:pd/amuc/database`, then you need to create the file `S:IceBBS.FIND`¹ and put one line in it containing the path to the database and the rootname of the database, in this case it would look like:

```
dh1:pd/amuc/database/amuc1
```

3. if you have AMUC CDROM 2, the name of this CDROM name is `AMUC2:`. If you start `MUIFind` with `AMUC2:` in a drive `MUIFind` will see it and will load the file:

```
AMUC2:S/IceBBS.FIND
```

from the AMUC CD. If you have other databases you want included in the searching you should copy the `IceBBS.FIND` file off of `AMUC2:S` into your `S:` directory and then edit that copy to include the name and path information of the other databases on separate lines. (`MUIFind` looks in `S:` before looking in `AMUC2:S`, so once there is an `S:IceBBS.FIND` file in your `S:` it will not see the one on `AMUC2:`).

32.2.2 Usage of MUIFind

32.2.2.1 Search by file contents

When the *Search by file contents* mode is selected the `MUIFind` display is split into two searching functions. The upper area is for searches on keywords and combinations of keywords. Typically you can take a guess at a likely keyword and try it here. If `MUIFind` fails to find anything you can use the other searching function to look for a keyword by using AmigaDOS wild cards.

¹ If your `MUIFind` does not work try copying the `IceBBS.FIND` file to `VBBS.FIND` which is what it used to be called.

1 Conventions

This is the Sysop's Manual for the demo version of IceBBS, as such it is an edited version of the full manual. It contains the information necessary to configure and run the demo version. As well, it contains partial documentation on the other areas of the IceBBS system that are not included the demo version, so that you can determine for yourself if the full version contains features you need.

The documentation for the demonstration version is available in two forms: as an AmigaGuide file and as a PostScript file ready for printing on a PostScript printer.

This manual is written in Texinfo, a macro package from GNU for use with TeX. There is both a printed copy and AmigaGuide version of this document.

In this manual a few typographic conventions are used.

- Things you type into the computer (like modem commands, script files, AmigaDOS commands and file names) are printed in a *type-writer* like font, like this command: 'DIR',
- Comments within script files and data files are written in italics like: *this is a comment*. Generally you should not type in the comments,
- Some of the configuration files have lines that are too long to print without breaking the line. To show that a line has been split and the next line should be added to it without a line break the '+' character is used¹. Do not type this character into your files,
- In some of the command descriptions the '|' character is used to separate several alternate items which you must choose between,
- In some of the command descriptions the '[]' (square brackets) are used to surround optional parameters for the command,
- Ellipses ('...') are used to indicate sections that can be repeated,
- Spaces in configuration files and AmigaDOS commands and scripts are important, the case (upper or lower) of characters may or may not be important.

32 Finding Files

One of the biggest problems facing BBS users and sysops today is finding the file they want out of the thousands of files that are on a particular BBS. IceBBS uses a pre-built database which contains the words that can be searched for and lists that identify which files each word is contained in. This system results in very fast searching for the user.

The user interface to the file finding system is provided by two buttons in the Get Files tool of IceTERM. These are the Find and Wild Find buttons. When pressed they will summon a string requester where the user types in the string he wants to search for. These strings for the Find button can contain the same logical operators as the 'FIND' utility uses. The Wild Find button allows AmigaDOS wild cards to be used to match against file names on the BBS.

32.1 Find, the sysop interface

This is the sysop's file finder tool, it is run from the CLI and the syntax is:

```
Find [-flags] "words & to & find"
```

Where the flags are:

- dDBname is the root name of a database. Find will need to access the files: 'DBname.cfg', 'DBname.hsh', 'DB.fnm', and 'DBname.occ'. This parameter is optional, if it is not included then Find will search all the databases that are listed in the 'S:IceBBS.FIND' file.
- s if this flag is supplied then the program will prompt you to adjust the security levels needed to access the databases that have been specified with the '-d' flag or loaded from the 'S:IceBBS.FIND' file. This works in the same way as the security set by the access command; note that if a user gets access to a database he can search for and download any file referenced by the database.

the rest of the command line are the words to be searched for. Each string will be parsed by 'find' and each word within the string will be searched for in the database(s). This produces a table of files for each word. Find can apply several operators to combine these tables and hence allow you to search for files which must, may or cannot contain several words. The operators are:

¹ It will be the last character on the line.

IceBBS, The Sysop's Guide

for demonstration version v56
April, 1995.

Stephen Vermeulen

US\$500-600) anyway, they should probably pay for themselves within a year.

To talk to your ISP over a basic modem connection requires you use something like PPP (Point to Point Protocol), SLIP (Serial Line Internet Protocol) or CSLIP (Compressed SLIP). All of these are available as either freeware or shareware.

You then need to run a TCP/IP protocol stack. On the Amiga there are two of these: AmiTCP and AS225R2. AmiTCP used to be freeware but has since become commercial, it is by far the most popular because of it's freeware roots. It also has a programmers interface that is very similar to the model used on UNIX machines, for this reason it is fairly easy to port UNIX networking software to AmiTCP. The IceBBS IRC and FTP doors both are written to use the AmiTCP version 4.0 (or higher) interface.

The AS225R2 product was one that Commodore was working on at the time they filed for bankruptcy. Commodore sublicensed it to a number of developers and at least two commercial versions will be available soon. The IceBBS IRC and FTP doors will not work with these products.

There are two other networking stacks available commercially, these are Envoy (from IAM, email 'info@iam.com') an Amiga-only product and DLAN-FS a DECNET based system. Either of these can be run across the same network as AmiTCP, allowing you to use the best features of both.

31.2 Internet real time conferencing

An IceBBS system can be configured to allow users to access and participate in real time conferencing over the Internet through IRC (Internet Relay Chat). This service is provided through the 'IceIRC' door (see Section 41.10.2 [IceIRC], page 184). To configure this is very easy once you have AmiTCP installed and running. Just drag and drop the icon for the 'IceIRC' server and adjust the 'access=' tool type if necessary.

31.3 File transfer

'IceFTP' (see Section 41.9.2 [IceFTP], page 184) is an IceBBS door that will provide a point and click (directory utility like) user interface to the Internet FTP tool. When this door is installed and AmiTCP is running your users can access file servers around the world, browse those directory trees and download whatever they want. Since 'IceFTP' like other IceTERM tools can be run several times at once by a single user, users can browse one site while downloading from others.