

icebbs_demo

COLLABORATORS

	<i>TITLE :</i> icebbs_demo		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		October 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	icebbs_demo	1
1.1	icebbs_demo.guide	1
1.2	icebbs_demo.guide/Conventions	3
1.3	icebbs_demo.guide/Introduction	4
1.4	icebbs_demo.guide/Demo version	4
1.5	icebbs_demo.guide/Features	5
1.6	icebbs_demo.guide/Custom terminal	6
1.7	icebbs_demo.guide/Performance	6
1.8	icebbs_demo.guide/Requirements	7
1.9	icebbs_demo.guide/Quick start	8
1.10	icebbs_demo.guide/Distribution archive	8
1.11	icebbs_demo.guide/Install script	9
1.12	icebbs_demo.guide/Simple modem config	10
1.13	icebbs_demo.guide/Adding commands	10
1.14	icebbs_demo.guide/Using IceConfig	11
1.15	icebbs_demo.guide/The hailing file	11
1.16	icebbs_demo.guide/The configuration file	11
1.17	icebbs_demo.guide/The root directories	12
1.18	icebbs_demo.guide/Locating SIGs	12
1.19	icebbs_demo.guide/The feedback system	12
1.20	icebbs_demo.guide/Enabling the DOS shell	13
1.21	icebbs_demo.guide/Starting the BBS	14
1.22	icebbs_demo.guide/Serial ports	14
1.23	icebbs_demo.guide/Internal serial	14
1.24	icebbs_demo.guide/A2232 card	15
1.25	icebbs_demo.guide/ASDG DSB	17
1.26	icebbs_demo.guide/Multiface II and III	18
1.27	icebbs_demo.guide/ComPorts board	18
1.28	icebbs_demo.guide/AE modem	19
1.29	icebbs_demo.guide/Cables	19

1.30 icebbs_demo.guide/Null modem	19
1.31 icebbs_demo.guide/Nullmodem.device	20
1.32 icebbs_demo.guide/Required libraries	21
1.33 icebbs_demo.guide/Register settings	21
1.34 icebbs_demo.guide/Old modems	21
1.35 icebbs_demo.guide/Advanced modems	23
1.36 icebbs_demo.guide/MNP modems	24
1.37 icebbs_demo.guide/HST modems	25
1.38 icebbs_demo.guide/USR Dual Standard	26
1.39 icebbs_demo.guide/SupraFAX	26
1.40 icebbs_demo.guide/V32 modems	28
1.41 icebbs_demo.guide/V34 modems	28
1.42 icebbs_demo.guide/V42 modems	28
1.43 icebbs_demo.guide/High speed protocol	29
1.44 icebbs_demo.guide/Access control	30
1.45 icebbs_demo.guide/Levels	30
1.46 icebbs_demo.guide/Mixing levels	31
1.47 icebbs_demo.guide/Groups	32
1.48 icebbs_demo.guide/Number of groups	33
1.49 icebbs_demo.guide/DOS shell access	33
1.50 icebbs_demo.guide/Setting levels	33
1.51 icebbs_demo.guide/Privileges	34
1.52 icebbs_demo.guide/Time limits	35
1.53 icebbs_demo.guide/New users	35
1.54 icebbs_demo.guide/IceBBS.ACCESS files	36
1.55 icebbs_demo.guide/IceBBS.SHELLACCESS file	36
1.56 icebbs_demo.guide/IceBBS.MAXACCESS file	37
1.57 icebbs_demo.guide/IceBBS.MSGID file	37
1.58 icebbs_demo.guide/TRIM.LASTTRIM file	38
1.59 icebbs_demo.guide/IceBBS.SIGS file	38
1.60 icebbs_demo.guide/IceBBS.NEWSIGS file	38
1.61 icebbs_demo.guide/Newsigs file format	38
1.62 icebbs_demo.guide/Email SIG name	39
1.63 icebbs_demo.guide/SIG access control	40
1.64 icebbs_demo.guide/IceBBS.BINIT file	40
1.65 icebbs_demo.guide/Line 1	42
1.66 icebbs_demo.guide/Line 2	42
1.67 icebbs_demo.guide/Line 3	42
1.68 icebbs_demo.guide/Line 4	42

1.69 icebbs_demo.guide/Line 5	43
1.70 icebbs_demo.guide/Line 6 and 7	43
1.71 icebbs_demo.guide/Line 8	43
1.72 icebbs_demo.guide/Line 9	44
1.73 icebbs_demo.guide/Line 10	44
1.74 icebbs_demo.guide/Line 11	44
1.75 icebbs_demo.guide/Line 12	44
1.76 icebbs_demo.guide/Line 13	45
1.77 icebbs_demo.guide/Line 14	45
1.78 icebbs_demo.guide/Line 15	45
1.79 icebbs_demo.guide/Line 16	46
1.80 icebbs_demo.guide/Line 17	46
1.81 icebbs_demo.guide/Fields 1 to 11	46
1.82 icebbs_demo.guide/Fields 12 and on	48
1.83 icebbs_demo.guide/Use with TrapDoor	48
1.84 icebbs_demo.guide/Polling script	49
1.85 icebbs_demo.guide/Normal line	49
1.86 icebbs_demo.guide/IceBBS.DIRS file	50
1.87 icebbs_demo.guide/IceBBS.HAIL file	50
1.88 icebbs_demo.guide/IceBBS.ARBITRATE file	51
1.89 icebbs_demo.guide/IceBBS.NOASCII file	52
1.90 icebbs_demo.guide/IceBBS.FIDOCFG file	52
1.91 icebbs_demo.guide/ASCII menus	52
1.92 icebbs_demo.guide/ASCII account	53
1.93 icebbs_demo.guide/ASCII alias	53
1.94 icebbs_demo.guide/ASCII bye	54
1.95 icebbs_demo.guide/ASCII cd	55
1.96 icebbs_demo.guide/ASCII chat	55
1.97 icebbs_demo.guide/ASCII confer	55
1.98 icebbs_demo.guide/ASCII ctrl	55
1.99 icebbs_demo.guide/ASCII describe	56
1.100icebbs_demo.guide/ASCII dir	56
1.101icebbs_demo.guide/ASCII display	57
1.102icebbs_demo.guide/ASCII download	57
1.103icebbs_demo.guide/ASCII email	58
1.104icebbs_demo.guide/ASCII feedback	58
1.105icebbs_demo.guide/ASCII goto	59
1.106icebbs_demo.guide/ASCII lock	60
1.107icebbs_demo.guide/ASCII login	60

1.108icebbs_demo.guide/ASCII news	61
1.109icebbs_demo.guide/ASCII password	61
1.110icebbs_demo.guide/ASCII prompt	61
1.111icebbs_demo.guide/ASCII protocol	62
1.112icebbs_demo.guide/ASCII SIGs	63
1.113icebbs_demo.guide/ASCII term	64
1.114icebbs_demo.guide/ASCII top	65
1.115icebbs_demo.guide/ASCII upgrade	65
1.116icebbs_demo.guide/ASCII upload	65
1.117icebbs_demo.guide/ASCII users	65
1.118icebbs_demo.guide/ASCII who	65
1.119icebbs_demo.guide/ASCII xdown	66
1.120icebbs_demo.guide/ASCII xup	66
1.121icebbs_demo.guide/Error messages	66
1.122icebbs_demo.guide/MOTD	70
1.123icebbs_demo.guide/Makemotd	70
1.124icebbs_demo.guide/Motdinfo	71
1.125icebbs_demo.guide/NewSince	72
1.126icebbs_demo.guide/IsNetUp	72
1.127icebbs_demo.guide/Append	72
1.128icebbs_demo.guide/Account	72
1.129icebbs_demo.guide/Edituser	73
1.130icebbs_demo.guide/EU	73
1.131icebbs_demo.guide/Coupon system	74
1.132icebbs_demo.guide/Sharing lines	75
1.133icebbs_demo.guide/Pause line	75
1.134icebbs_demo.guide/Timed script files	76
1.135icebbs_demo.guide/Front doors	77
1.136icebbs_demo.guide/FidoNet	78
1.137icebbs_demo.guide/FidoNet scripts	79
1.138icebbs_demo.guide/FidoNet security	79
1.139icebbs_demo.guide/Multiple networks	80
1.140icebbs_demo.guide/FidoNet commands	80
1.141icebbs_demo.guide/Export FidoNet	80
1.142icebbs_demo.guide/Import FidoNet	82
1.143icebbs_demo.guide/Viewing FidoNet packets	83
1.144icebbs_demo.guide/Posting to FidoNet	83
1.145icebbs_demo.guide/Testing time of day	84
1.146icebbs_demo.guide/Waiting in scripts	85

1.147icebbs_demo.guide/Exporting netmail	85
1.148icebbs_demo.guide/Debugging scripts	86
1.149icebbs_demo.guide/UUCP	86
1.150icebbs_demo.guide/Finding a feed	87
1.151icebbs_demo.guide/Configuring for UUCP	87
1.152icebbs_demo.guide/UUCP support commands	88
1.153icebbs_demo.guide/NewsIn	88
1.154icebbs_demo.guide/UUExport	89
1.155icebbs_demo.guide/InMail	89
1.156icebbs_demo.guide/FwdEmail	90
1.157icebbs_demo.guide/Outmail	90
1.158icebbs_demo.guide/Internet tools	91
1.159icebbs_demo.guide/Searching	94
1.160icebbs_demo.guide/Find	95
1.161icebbs_demo.guide/MUIFind	96
1.162icebbs_demo.guide/FindScan	98
1.163icebbs_demo.guide/MySort	99
1.164icebbs_demo.guide/FindStat	99
1.165icebbs_demo.guide/FindBuild	99
1.166icebbs_demo.guide/Find databases	100
1.167icebbs_demo.guide/Find limits	100
1.168icebbs_demo.guide/File system	101
1.169icebbs_demo.guide/File format	102
1.170icebbs_demo.guide/Indirected storage	103
1.171icebbs_demo.guide/File tools	104
1.172icebbs_demo.guide/BBSJoin	104
1.173icebbs_demo.guide/BBSSplit	104
1.174icebbs_demo.guide/UJoin	105
1.175icebbs_demo.guide/AmiJoin	105
1.176icebbs_demo.guide/FMT	105
1.177icebbs_demo.guide/DelOld	107
1.178icebbs_demo.guide/Access	108
1.179icebbs_demo.guide/HashAccess	109
1.180icebbs_demo.guide/FileList	109
1.181icebbs_demo.guide/Store	109
1.182icebbs_demo.guide/Listing new files	109
1.183icebbs_demo.guide/Bulk programs	110
1.184icebbs_demo.guide/BulkSplit	110
1.185icebbs_demo.guide/BulkJoin	111

1.186icebbs_demo.guide/BulkTest	111
1.187icebbs_demo.guide/Bulk example	112
1.188icebbs_demo.guide/Message bases	113
1.189icebbs_demo.guide/Feedback messages	113
1.190icebbs_demo.guide/SIGs messages	115
1.191icebbs_demo.guide/SIG file format	116
1.192icebbs_demo.guide/FirstLinkItem struct	116
1.193icebbs_demo.guide/LinkItem struct	117
1.194icebbs_demo.guide/DataItem struct	119
1.195icebbs_demo.guide/VMSG struct	119
1.196icebbs_demo.guide/Traversing links	122
1.197icebbs_demo.guide/Find message by number	122
1.198icebbs_demo.guide/Find the next message	125
1.199icebbs_demo.guide/Find the previous message	126
1.200icebbs_demo.guide/Find end of file	126
1.201icebbs_demo.guide/The message reader	127
1.202icebbs_demo.guide/Support utilities	127
1.203icebbs_demo.guide/Triming SIGs	128
1.204icebbs_demo.guide/Adding SIGs	128
1.205icebbs_demo.guide/Mail	128
1.206icebbs_demo.guide/Electronic mail	130
1.207icebbs_demo.guide/EMail directories	130
1.208icebbs_demo.guide/Mail between users	131
1.209icebbs_demo.guide/Mail to FidoNet	132
1.210icebbs_demo.guide/Internet mail	133
1.211icebbs_demo.guide/EMail support commands	134
1.212icebbs_demo.guide/CheckMail	134
1.213icebbs_demo.guide/EMail	135
1.214icebbs_demo.guide/Forward	136
1.215icebbs_demo.guide/Send	137
1.216icebbs_demo.guide/Monitor	137
1.217icebbs_demo.guide/Conferencing	138
1.218icebbs_demo.guide/Configuring conference	139
1.219icebbs_demo.guide/Confer	139
1.220icebbs_demo.guide/Problems	140
1.221icebbs_demo.guide/News bulletins	140
1.222icebbs_demo.guide/IceBBS.NEWS file	140
1.223icebbs_demo.guide/News files	141
1.224icebbs_demo.guide/DOS shell	141

1.225icebbs_demo.guide/Doors	142
1.226icebbs_demo.guide/Door tool types	143
1.227icebbs_demo.guide/Operation theory	144
1.228icebbs_demo.guide/RemFile door	145
1.229icebbs_demo.guide/Preview door	146
1.230icebbs_demo.guide/Edituser door	146
1.231icebbs_demo.guide/CircleMUD door	147
1.232icebbs_demo.guide/ARexx doors	147
1.233icebbs_demo.guide/IceFTP	147
1.234icebbs_demo.guide/IceIRC	148
1.235icebbs_demo.guide/Icon files	149
1.236icebbs_demo.guide/Account files	149
1.237icebbs_demo.guide/CDROM drives	150
1.238icebbs_demo.guide/Deferred assigns	151
1.239icebbs_demo.guide/CDTV	152
1.240icebbs_demo.guide/CD32	152
1.241icebbs_demo.guide/CDROM-FS	152
1.242icebbs_demo.guide/Xetec	153
1.243icebbs_demo.guide/AsimCDFS	153
1.244icebbs_demo.guide/AmiCDROM	153
1.245icebbs_demo.guide/CDROM sources	154
1.246icebbs_demo.guide/Walnut Creek	154
1.247icebbs_demo.guide/Fred Fish	155
1.248icebbs_demo.guide/AMUC	155
1.249icebbs_demo.guide/Removable drives	155
1.250icebbs_demo.guide/Priorities	156
1.251icebbs_demo.guide/ANSI codes	157
1.252icebbs_demo.guide/Escape sequences	157
1.253icebbs_demo.guide/Amiga ANSI	158
1.254icebbs_demo.guide/Cursor movement	159
1.255icebbs_demo.guide/Erasing	159
1.256icebbs_demo.guide/Colour graphics	159
1.257icebbs_demo.guide/Text attributes	161
1.258icebbs_demo.guide/Future enhancements	162
1.259icebbs_demo.guide/Glossary	163
1.260icebbs_demo.guide/Warranty	167
1.261icebbs_demo.guide/Updates	167
1.262icebbs_demo.guide/Copying	168
1.263icebbs_demo.guide/Support	168

1.264icebbs_demo.guide/No warranty	168
1.265icebbs_demo.guide/Disclaimer	169
1.266icebbs_demo.guide/Thanks	169
1.267icebbs_demo.guide/Tool types index	170
1.268icebbs_demo.guide/Command index	170
1.269icebbs_demo.guide/Concept index	172

Chapter 1

icebbs_demo

1.1 icebbs_demo.guide

IceBBS

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.

IceBBS is an alternative Bulletin Board System for the Amiga series of computers. Its distinguishing feature is the builtin session multiplexing that allows a user to do several things on the BBS at once. To make this possible a custom terminal program called IceTERM must be used to call the BBS. For those people who do not have a copy of IceTERM (which is freely distributable) an ASCII interface to the BBS is also provided. An additional program called IceOLR is used to provide an off-line message posting, reading and replying capability.

IceBBS also has a set of additional sysop tools which provide control over the BBS and also can be used to connect a IceBBS system to FidoNet or the Internet (via UUCP). These tools are not part of the demonstration package.

The main limitations of the demonstration version of the BBS are:

- * 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, in any one session a user can start up to three doors)
- * no support utilities, including the user account editor and the FidoNet and USENET support utilities.
- * only one phone line is supported.

Printed versions of the IceOLR and IceTERM manuals are available from the Author and may also be available from your local Amiga Dealer, IceBBS sysop, or Amiga Users Group. This 200 page manual includes about 40 figures and is printed in a small book format (5x8 inch pages) and coil bound so it lays flat open for ease of reference.

Copying	IceBBS is not distributable
Conventions	Typographic conventions
Introduction	What is IceBBS?
Quick start	Setting up quickly
Serial ports	Both internal and expansion
Required libraries	What you need to run IceBBS
Register settings	For various modems
Access control	Understanding the security and account limits
IceBBS.ACCESS files	Controlling access to the file areas
IceBBS.SHELLACCESS file	Controlling access to the DOS shell
IceBBS.MAXACCESS file	How many security groups do you need?
IceBBS.MSGID file	Generation of unique message IDs
TRIM.LASTTRIM file	Control file for the trim command
IceBBS.SIGS file	Old message bases, location and security
IceBBS.NEWSIGS file	New message bases, location and security
IceBBS.BINIT file	The master configuration file
IceBBS.DIRS file	Locating your file areas
IceBBS.HAIL file	Greeting callers
IceBBS.ARBITRATE file	Controlling CDROM swapping
IceBBS.NOASCII file	Removing the ASCII mode
IceBBS.FIDOCFG file	The node number
ASCII menus	The ASCII side of IceBBS
Error messages	Customizing the IceBBS error messages
MOTD	The message of the day
Account	The obsolete account editing program
Edituser	Editing user account files
EU	A point and click edituser
Coupon system	Automatic account upgrading and downgrading
Sharing lines	How to get IceBBS to share serial ports
FidoNet	The FidoNet support utilities
UUCP	The USENET and Internet support utilities
Internet tools	Direct support for the Internet
Searching	Searching for files by content
File system	The file areas
Message bases	The SIGs or message areas
Mail	Feedback to the sysop
Electronic mail	The email subsystem
Monitor	Watching what a user is doing
Conferencing	Joining the online conference
News bulletins	Adding on-line news and help files

DOS shell	Configuring the remote sysop's DOS shell
Doors	The expansion door system
Icon files	Information about the various tool types
Account files	Format of the user account files
CDROM drives	Using CDROMs with IceBBS
Removable drives	Using removable media drives
Priorities	System task priorities
ANSI codes	Escape sequences for coloured text etc.
Future enhancements	What's planned for the future
Glossary	Terms and definitions
Warranty	Legal terms and conditions
Thanks	Without which this would not have happened

Indices...

Tool types index	Index of tool types
Command index	Index of support commands
Concept index	Main Index

1.2 icebbs_demo.guide/Conventions

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 typewriter 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 used (1). 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.

----- Footnotes -----

(1) It will be the last character on the line.

1.3 icebbs_demo.guide/Introduction

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 Rockies (1), and the first general release of the BBS was made during the winter.

Demo version	Restrictions of the IceBBS demo version
Features	An overview of what's provided
Custom terminal	Your user's will need this
Performance	What sort of performance?
Requirements	What system is required to do this

----- Footnotes -----

(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.

1.4 icebbs_demo.guide/Demo version

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 FidoNet and USENET support utilities.
- * only one phone line is supported.

1.5 icebbs_demo.guide/Features

Features

=====

This is the sysop's manual for IceBBS (1), an alternative multi-line BBS system for the Amiga. IceBBS was designed with several goals in mind:

- * The software should support a fully-multiplexed, error-free, communications channel,
 - * The user should be able to do more than one thing at a time,
 - * The BBS should not require an accelerated machine to run more than one line,
 - * The BBS should not eat huge amounts of RAM,
 - * The BBS should easily support a mixed set of serial ports and modems, as well as providing support for the new fixed baud rate modems (between computer and modem) at the same time as the old style modems,
 - * The BBS should provide the basics (messages, sysop chat, conference mode, email and a file system),
 - * The BBS and terminal programs should offer some means of expansion, this is provided by the doors mechanism which allows new clients and servers that behave in the same way as the built in ones to be added,
-

- * The BBS should take advantage of the capabilities of advanced modems (error correction and compression) to off-load work from the Amiga's CPU,
- * The BBS should allow messages to be imported from and exported to the major networks (FidoNet and USENET),
- * Support CDROMS,
- * Support some access to the Internet.

At this point the software meets all of these goals.

----- Footnotes -----

(1) 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 baud, 24 hours a day (V32bis and HST) on (403) 284-2048 and (403) 284-5625 and in the evenings (mountain standard time) by voice at (403) 282-7990.

1.6 icebbs_demo.guide/Custom terminal

Custom terminal
=====

To realize the multiplexed serial connection (and the BBS multitasking capabilities it provides) it was necessary to develop a special terminal program. This terminal program currently is only available for Amiga computers. As this poses a problem for non-Amiga users an ASCII interface to IceBBS has been developed; of course this lacks some functionality and it has not been as extensively tested at this point in time.

The terminal program is also copyrighted software; however, it is freely distributable, so all your users can get a copy of it for free.

1.7 icebbs_demo.guide/Performance

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 baud(1), 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.

----- Footnotes -----

(1) 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.

1.8 icebbs_demo.guide/Requirements

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 once, or if they have selected a large editor (such as a commercial word processor (1)) to edit their messages with.

The IceBBS program will also run on a 512K machine, although more memory may be required when your users start multitasking its functions. At a minimum you should budget about 200K for the first line and about 100K per additional line.

----- Footnotes -----

(1) I had one user who used WordPerfect to do this.

1.9 icebbs_demo.guide/Quick start

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 subsystems please refer to the appropriate chapters.

Distribution archive	The contents of the file
Install script	Sorry, no Amiga Installer yet
Simple modem config	The basic modem setup
Adding commands	Where to put the {No Value For "IceBBS"} commands
Using IceConfig	IceConfig the configuration editor
The hailing file	Greeting users
The configuration file	Configuring the BBS
The root directories	The directories the users can access
Locating SIGs	About your SIGs
The feedback system	The old simple SIG
Enabling the DOS shell	If you have co-sysops
Starting the BBS	The last stages

1.10 icebbs_demo.guide/Distribution archive

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),

ErrorMessages

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

BBS_S

this directory contains five files for your S: directory, there is an example of a script file to handle FidoNet (see FidoNet) and USENET (see UUCP),

ASCII

this directory contains a working set of ASCII menus (see ASCII menus),

L

this directory contains the handler for the remote sysop shell
(see DOS shell),

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.

1.11 icebbs_demo.guide/Install script

The Install Script

=====

If you are setting up a single line IceBBS system and you are using a
normal 2400 baud modem (ie. one that does not support MNP or higher
baud rates) you can install IceBBS by just typing the commands:

```
CD your_directory_for_icebbs
lharc -m x IceBBS54
Install
```

Once the script has finished there are a few things left to do.
First you should add a line to your startup sequence file to mount the
VAUX: device, for example:

```
mount VAUX:
```

You will usually want to start IceBBS via your startup sequence, so
add another line to run IceBBS, you should probably add the BBS_C
directory to your command search path by using the AmigaDOS path
command. You probably will want to modify the s:user-startup file
rather than the startup sequence file, as this is the safer and
system-preferred way. Note that if the initial shell window will not
disappear after you have made your changes you should add some >NIL:
and <NIL: entries to the new commands between the command name and the
first parameter that the command takes.

Next you must configure IceBBS for your modems (see
Register settings). To do this you may need to make a few edits to the
S:IceBBS.DIRS (see IceBBS.DIRS file) and S:IceBBS.BINIT (see
IceBBS.BINIT file) files. The make_binit (1) command will make an
S:IceBBS.BINIT file for you taking into account the root directory you
specified, it will also make an S:IceBBS.DIRS file which lets the users
see the contents of the directory (2): <dir>BBSFiles. The script file
will also set up three SIG areas for the SIGs tool.

----- Footnotes -----

(1) This command is not yet available, sorry

(2) The default VBBS.DIRS file will make several directories accessible

1.12 icebbs_demo.guide/Simple modem config

Configuring Your Modem

=====

If you have chosen to just use the default installation, the next thing to do is to configure your modem. The basic commands that must be sent to the modem are:

```
AT&D3      ;hang up using DTR
AT&C1      ;pass carrier detect
ATV1      ;command results as text strings
ATX4      ;report most modem events
ATL0      ;minimum speaker volume
ATM0      ;turn off speaker
ATS0=2    ;answer phone on second ring
```

1.13 icebbs_demo.guide/Adding commands

Commands

=====

Install the IceBBS program and related commands (see Command index) (Confer, Mail, BBSSplit, 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.

1.14 icebbs_demo.guide/Using IceConfig

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 run⁽¹⁾ and the help facility requires that AmigaGuide or MultiView is installed.

----- Footnotes -----

(1) There 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.

1.15 icebbs_demo.guide/The hailing file

The S:IceBBS.HAIL File

=====

This file (which is required to be in your S: directory) is used by IceBBS to send an ASCII text welcoming banner to all who successfully connect to the system. This welcoming text may be as long as you wish. It is sent before the IceBBS protocol is started, so that non-Amiga users can see they have connected to a very special kind of BBS they cannot make full use of, and to give them instructions on how to start the ASCII side of the BBS.

1.16 icebbs_demo.guide/The configuration file

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. For more information see IceBBS.BINIT file.

1.17 icebbs_demo.guide/The root directories

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). Because of this the directory names cannot have embedded spaces in them. For more information see IceBBS.DIRS file.

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 Access control). To give selected users the keys to get past these locks you use the edituser command (see Edituser).

1.18 icebbs_demo.guide/Locating SIGs

The S:IceBBS.NEWSIGS File
=====

This file contains the locations of the various directories on your system where the SIG (1) 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 IceBBS.NEWSIGS file.

Also see Access control for more information about how IceBBS security works.

----- Footnotes -----

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

1.19 icebbs_demo.guide/The feedback system

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 Feedback messages) all that must be done is to edit the S:IceBBS.BINIT (see IceBBS.BINIT file) 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 Mail). It is strongly advised that these two files be on non-volatile media; otherwise, when you crash you will lose your message base. Do not worry about scanning speed if you are using floppies since IceBBS loads all unread messages it finds into RAM as soon as the user starts reading (1), and unless you are using 9600 baud (or faster) modems the modem will not be able to keep up with the disk. Of course if you are using high speed modems, why do you not have a hard drive?

The feedback system has a number of major limitations, these include:

- * The users can only backup by using the ASCII terminal mode,
- * Threaded reads are not implemented,
- * Old messages cannot be deleted from the message base (apart from deleting the entire message base).

----- Footnotes -----

(1) This only applies to the current version of the IceTERM reader, the ASCII reader does not work this way and a future version of the IceTERM program will not either.

1.20 icebbs_demo.guide/Enabling the DOS shell

Enabling the Remote Shell

=====

In order for your remote sysops to use the remote AmigaDOS shell (see DOS shell) you will need to do a few additional things. First copy the file VAUX-Handler into your L: directory. Second, add the small mountlist file that is found in the DEVS directory of the distribution archive to your DEVS:mountlist file. This will put the proper mounting information for the handler into your system. Next add to your startup sequence script file a command like:

mount VAUX:

to bring the handler online sometime before the command to run IceBBS. And last, create a file in S: called IceBBS.SHELLACCESS (1). See the chapter on the remote DOS shell for more information.

----- Footnotes -----

(1) It can contain anything.

1.21 icebbs_demo.guide/Starting the BBS

Running the BBS

=====

Once the previous configuration steps have been finished you run IceBBS just like any other CLI only program, type: RUN IceBBS. Depending on what other software you intend to run at the same time as IceBBS you may need to either run IceBBS at a higher priority or run the other software at a lower priority. If you are often running compute bound programs like a raytracer or an archiver like Lharc, or you make heavy use of the WorkBench, the easiest thing to do is to run IceBBS at a priority of 4, using the AmigaDOS command ChangeTaskPri 4 in the CLI or Shell window before starting IceBBS. It is recommended that you set the CLI stack to about 10K before starting IceBBS.

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.

1.22 icebbs_demo.guide/Serial ports

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.

Internal serial	The standard internal serial port
A2232 card	Commodore's 7 line card
ASDG DSB	ASDG's 2 line card
Multiface II and III	A German 2 line card
ComPorts board	An 8 port card from New York
AE modem	An internal modem
Cables	What ICE needs in a cable
Null modem	Direct connection
Nullmodem.device	Simulating a direct connection in software

1.23 icebbs_demo.guide/Internal serial

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 this (1). 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.

----- Footnotes -----

(1) This has been improved in AmigaDOS 2.0

1.24 icebbs_demo.guide/A2232 card

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 3.5MHz(1) (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 parts there is room here for an ambitious hacker to write a higher performance version (2) of this driver software.

The A2232 card performs quite well, its only weakness is the lack of official baud rates higher than 19.2K baud (there is an undocumented rate of 115K baud, but that does not work with handshaking and with a direct wired IceBBS connection it seems to result in about 25K baud transfers). The driver software also seems to have problems when 6 ports are being run at 19.2K baud with RTS/CTS handshaking (3). It handles 6 ports at 9600 baud fine, but the higher baud rate seems to cause some of the lines to go to sleep (4) for extended periods of time during transfers.

The A2232 card's driver software also has another bug, it does not drop the DTR line properly (5) when the serial device is closed. Since this is the hardware method of resetting a modem (by using the AT&D3 setting) any system using an A2232 may have to use the software method of ++ATZ instead. The easy way to test your serial port for this

problems is to fire up a terminal program, watch that the modem connected to a A2232 serial port shows its "TR" light go to on and then exit the terminal program, the "TR" light should now extinguish. If it stays on then the serial driver software has the bug. A more recent version of the driver exists that has fixed this problem, but it may not be shipped to end users for a while. There is another bug with the serial driver that can cause lines to stop responding. There is a software patch for this which has been added to IceBBS.

From browsing the A2232 manual's schematic diagrams of the A2232 card it appears that the 6551 chips are being driven by a 1.79MHz clock (actually one quarter of the Amiga's 7.1MHz clock). Now according to the only information (6) I have on the 6551 chip the clock speed required for this chip should be 1.8432MHz. So it looks like the actual maximum baud rate might be 18646 baud(7). Also it might be possible (8) to modify 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.

----- Footnotes -----

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

(2) Please someone, do this!

(3) As soon as you try running 3 lines at 19.2K baud the card starts to loose characters, this becomes worse as more lines are run.

(4) Of course this could be due to a bug in IceBBS or IceTERM, but they work fine with ASDG boards at higher baud rates even at the same time that the A2232 has stalled.

(5) At the time of writing of this manual this bug has been fixed in beta versions of the new driver and this driver has seen wide circulation.

(6) 6502 Assembly Language Programming by Lance A. Leventhal, Osborne and McGraw-Hill Inc. 1979

(7) There is a 1.8432MHz crystal installed on the board so the schematics might be completely misleading.

(8) This is a suggestion that may or may not work, I have not tried it with my board and I have no way of knowing if there will be some 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.

1.25 icebbs_demo.guide/ASDG DSB

ASDG - Dual Serial Board
=====

The ASDG dual serial card is a nice piece of hardware but the software is in need of some bug fixes. This beast is also a Zorro II card and it has two 9 pin (joystick type) connectors on the back panel. You can use standard AT serial cables with this card. The ASDG driver software is up to version 1.6, I advise you to send in your registration and get the most recent copy from ASDG since they have fixed some significant bugs in the earlier versions which were related to use with accelerator cards.

This card, on an accelerated Amiga, is capable of running two lines with IceBBS at baud rates of 153.6K baud, with an unaccelerated Amiga two lines at 38.4K baud is achievable. This card has four problems which I know of:

- * It does not coexist with TrapDoor 1.80 well. This is because of a bug in the way the card does either a CMD_CLEAR or a CMD_FLUSH,
 - * It drops characters from time to time. This only seems to happen
-

when there is handshaking involved as is the case when talking to high speed modems,

- * It suffers from some sort of mungwall incompatibility when it is first loaded from disk, if the serial port is opened before mungwall is started all seems to be fine.
- * It has a bug some where that will cause your system to crash after a period of time, the time this takes depends on the activity level of your system and the number of lines you have.

This card is not currently being manufactured by ASDG any more, so you will probably only find it on the used equipment market.

Summary: for lightly loaded one and two line systems this card will be fine; however, if you're trying to run a large system or stability is a concern then this card probably will not be satisfactory.

1.26 icebbs_demo.guide/Multiface II and III

Multiface II and III dual serial cards

=====

These are Zorro II cards that contain 2 serial ports and either one or two parallel ports. The cards work well in all Amigas (2000, 3000 and 4000). They run flawlessly with both IceBBS and IceTERM, even in heavily loaded environments. The only drawback with these cards is that they limit the total number of serial ports in a single machine to 10 (assuming you don't need a slot for a hard drive controller).

There may be a 4 port version of these cards but I have yet to see it.

1.27 icebbs_demo.guide/ComPorts board

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

1.28 icebbs_demo.guide/AE modem

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 control (1) 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.

----- Footnotes -----

(1) 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).

1.29 icebbs_demo.guide/Cables

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.

1.30 icebbs_demo.guide/Null modem

Direct Line Hookup

=====

It is also possible to run a direct serial connection between IceBBS and IceTERM. All you need is to make up a standard null-modem cable and connect it between the two serial ports. Next setup the S:IceBBS.BINIT

file to specify that the BBS serial port is a fixed rate line. Now to start the IceBBS protocol over the direct wire line, just type CONNECT and hit return in the terminal program's ASCII mode window, in a few seconds the BBS will send the hailing message and start the protocol. If you want a direct wired connection to run with the MNP/V42 protocol you should type a string that contains the MNP or V42 connect string, for example: CONNECT ARQ or CONNECT REL.

1.31 icebbs_demo.guide/Nullmodem.device

Simulating a null modem in software

=====

Rather than using two physical serial ports connected by a null modem cable it is possible to simulate the connection by software called the null modem device. This is a device that is placed in the DEVS: directory. It was written by Ian Hibbert who can be reached at:

```
UUCP: plunky@closet.wizdom.royle.org
FidoNet: 2:255/171.33
AmigaNet: 39:136/1.33
```

A typical configuration line for the S:IceBBS.BINIT file would be:

```
dhl:downloads nullmodem.device 0 38400 7 fixed XYZ +
    ATZ ATSO=1 mail:line2 -1 -1 -1
```

Note the two strings between the XYZ and mail: fields. The first string is the command to send to reset the modem after the BBS hangs up the line and sends an +++ sequence. Normally one puts an ATZ or an ATH command here. The second command is the string to send the modem to re-initialize it. In the case of the nullmodem.device one specifies an ATSO=1 which causes the device to auto-answer.

This makes it very easy to use nullmodem.device with IceTERM and IceBBS. Just configure the BBS line as shown above, and then use a dialing script like the following:

```
send ATD
cr
exit
```

with your IceTERM. If the BBS is using nullmodem.device unit number 0 as shown in the example, then IceTERM needs to be configured with the following tooltypes:

```
DEVICE=nullmodem.device
UNIT=1
```

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

1.32 icebbs_demo.guide/Required libraries

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.

1.33 icebbs_demo.guide/Register settings

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.

Old modems	The plain 1200 and 2400 baud units
Advanced modems	Higher speed units
MNP modems	Modems equipped with MNP
HST modems	USR's asymmetrical modulation
USR Dual Standard	USR's classic sysop modem
SupraFAX	High speed on a budget
V32 modems	The 9600 and 14400 baud modulation
V34 modems	As fast as can be
V42 modems	Modems equipped with V.42
High speed protocol	IceBBS's special protocol

1.34 icebbs_demo.guide/Old modems

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 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 (the only modem I have seen do this is an old Packard Bell 2400 baud modem),

AT&W This saves these configuration settings into the modem's memory.

The modem is now be set up for use with IceBBS.

----- Footnotes -----

(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.

1.35 icebbs_demo.guide/Advanced modems

Advanced modems

=====

These are the modems which you specify with a fixed baud setting in the S:IceBBS.BINIT file. This type of modem usually supports MNP or V.42 and it supports a fixed baud rate link between the computer and the modem. The general approach to configuring an advanced modem for use with IceBBS is to lock the baud rate of the linkage between IceBBS and the modem, enable CTS/RTS (1) flow control on that linkage and tell the modem to negotiate with the calling modem to establish the best protocol between them. IceBBS does not need to know or even care (2) that the link is an MNP link at 300 baud or an HST line at 14400 baud the protocol will still work fine. The following are the common settings that should be set for use with IceBBS:

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, the baud rate of the connection, and the error correction used in the connection,

ATV1

This makes the result codes be the English text strings,

ATX4

This makes the modem report all (3) events to us, the ones that IceBBS looks for at are the CONNECT, the baud rate string and the error correction protocol name,

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. (4)

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.

----- Footnotes -----

(1) The importance of flow control cannot be over emphasised.

(2) IceBBS can make good use of this information though, it will allow a faster version of the protocol to be used which will result in better throughput and less CPU loading on both ends of the connection.

(3) Users of US Robotic's Dual Standard and HST modems can use X6 here.

(4) 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.

1.36 icebbs_demo.guide/MNP modems

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:

AT%C1

Turn on compression (MNP level 5),

AT\J0

Turn off the automatic baud rate adjustment. This keeps the modem to computer connection at a fixed baud rate regardless of the actual modem to modem link speed. The GVC manual documents this incorrectly in one place and correctly in another so who knows what it is really supposed to be,

AT\N3

This enables the modem to negotiate for an auto reliable link, it first tries to get an MNP class connection and if that fails, it falls back to the old style connection,

AT\V1

Tells the modem to send extra result strings to describe the connection it got. This is not really necessary for IceBBS but it can be useful as a diagnostic,

AT\Q3

Tells the modem to use full CTS/RTS flow control between it and the computer, essential if you are wanting to go faster than 2400 baud,

AT\G0

This is a mysterious command, it seems that the G0 setting is required at least at the caller's end (when IceBBS is being run on old-style modems and the caller has an MNP modem), for this reason I recommend that MNP modems on the IceBBS end also be set to G0 (this is usually the default factory setting).

With these settings stored into the appropriate register set you should be ready to run IceBBS with an MNP type modem.

1.37 icebbs_demo.guide/HST modems

USR HST Modem

=====

IceBBS has now been tested with a pair of USR HST Modems, performance was good on unidirectional transfers (1630 CPS) but became poor when bidirectional transfers took place (about 350 CPS). The performance drop from the expected 800 CPS bidirectional can probably be attributed to the HST modem's asymmetric nature, it appears that the HSTs do a line retrain everytime that the high speed direction changes. It looks like the retrain process takes a significant amount of time to complete and hence we see a large drop in performance.

Note that the above figures are obtained when IceBBS and IceTERM are running in the error-corrected modem mode, if this protocol is disabled then some additional performance will be lost since IceBBS and IceTERM use a smaller packet size than the HST modem likes best.

For a BBS system I would strongly recommend not buying an HST modem, but rather getting a USR Dual Standard modem, as these now 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.

1.38 icebbs_demo.guide/USR Dual Standard

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.

1.39 icebbs_demo.guide/SupraFAX

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.

AT&F2

Select factor default settings for hardware handshaking,

ATW2

Should get the modem to report the connection speed rather than the speed of the serial port link (which is going to stay fixed),

AT&C1

Returns the current state of the carrier so the BBS can tell when the connection has been broken,

AT&D3

Get the modem to do a full reset when the BBS drops the DTR serial port line,

AT&K3

Enable the hardware handshaking,

AT\A3

Allows use of full MNP block sizing (this may not be too important),

AT\N3

Negotiate for the best connection, starting with V42, dropping back to MNP and then falling back to a regular non-corrected connection,

ATS0=2

Answer the phone on the second ring (so that all callers hear at least one ring),

ATS95=3

This convinces the modem to send the connection string as only one line (like other modems do),

AT&Y0

Make register set number 0 the default one,

AT&W0

Save these settings into register set 0, so that when the modem is reset or powered up again it remembers these settings.

These modems work pretty well now that Supra has upgraded the ROMs several times. There is a command to print out the ROM version, you might want to do this to check that you have the latest ROM version, and to upgrade if you have an older ROM. Their only failings are that older V32 modems have trouble connecting to them and the long distance callers do not always connect at the highest rate.

1.40 icebbs_demo.guide/V32 modems

V.32 and V.32bis Modems

=====

V32 is the international standard for 4800, 7200 and 9600 baud communications. V32 modems are available from a number of manufacturers, most V32 modems will also feature MNP and V42 error correction protocols. Note that V32 modems will only connect at 2400 baud when an HST modem is calling them. V32bis is the new international standard for communications at 12000 and 14400 baud, it will also fall back to the V32 speeds if necessary. The advantage a V32 or V32bis modem has over an HST modem is that it allows data to be sent in both directions 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 time (1).

----- Footnotes -----

(1) Without the line turn-around delay that will occur with HST modems.

1.41 icebbs_demo.guide/V34 modems

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.

1.42 icebbs_demo.guide/V42 modems

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.

1.43 icebbs_demo.guide/High speed protocol

High Speed Protocol

=====

IceBBS makes special provisions for V.42, MNP and HST connections. If such a connection is detected when the user logs in, IceBBS and IceTERM will start up a different version of the protocol that does not do the error correction (since the modems are doing this). When this is done the overhead on file transfers is reduced to about 2% and less CPU at both ends is required.

The error correction is almost entirely handled by the modems, but not quite. It appears that not all serial ports run without loosing bytes occasionally. Also, when a phone connection is broken some MNP modems return a brief burst of garbage to the computer, instead of falling silent. Both of these cases can be disastrous to a protocol like that used by IceBBS which requires that the data gets through intact.

As a result of this the high speed protocol still has to check for errors; however, at this point in time it does not have the ability to recover from errors. Therefore when an error is detected it has to break the connection. In the future a retry capability will be added to get around this limitation.

If you find that users complain a lot about IceTERM reporting an MNP Error and then hanging up the connection, you will have to turn off the high speed connection. This can be done by editing the string that the BBS looks for in the modem connection strings to decide if there is an error free link in place between the modems. This string is set in the S:IceBBS.BINIT file, all that needs to be done is to change this from its normal value (which might be ARQ or REL) to something that the modem does not return (like XYZ).

1.44 icebbs_demo.guide/Access control

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 SIG access control), 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.

Levels	Security levels and groups
Mixing levels	Selecting who gets or does what
Groups	Grouping access
Number of groups	How many security groups?
DOS shell access	Controlling access to the DOS shell
Setting levels	Modifying access levels
Privileges	Various account limits
Time limits	The user's time bank
New users	The values for a new account

1.45 icebbs_demo.guide/Levels

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.

----- Footnotes -----

(1) This is based on the AMUC Express BBS.

1.46 icebbs_demo.guide/Mixing levels

Mixing access levels

=====

It is also possible to mix and match the various levels so that a code of 1 0 0 1 would give a user access to the unique sysop functions but not to the areas that members and executive only have access to.

The access control system gets more versatile (and complicated) when you remember that each of the groups (there are four groups in the above examples) can each take on 256 different levels. The rule to remember is that if a user's account has not been set to a level equal to or greater than the level the sysop has set for the function (1) the user is trying to access then he cannot use that function.

For example if we add a fifth group (group number 4) to the previous example, this being used to control access to the doors. And if there are four doors on the BBS we might assign the following levels to each door:

0

for the IFF picture preview door,

1

for the voting door,

2

for the GPChess door,

10

for the remote sysop file maintainance door.

As everyone must have at least a level zero access in all the security groups this makes the preview door automatically available to everyone. Next consider a member who is given the access levels: 1 1 0 0 1. He will have access to both the voting door and the IFF picture previewer. If his level in group 4 is raised to 2 (1 1 0 0 2) then he will also get access to the GPChess door. However, he will not get access to the remote sysop file maintainance door (RemFiles) until his access level in group 4 is raised to at least 10.

----- Footnotes -----

(1) This includes: doors, individual SIGs and directories within the file system.

1.47 icebbs_demo.guide/Groups

Access levels in different groups

=====

It is also possible to give a user access to particular functions (or file areas or SIGs) in several ways. If, for example, we continue the previous example on the doors system. It is possible to tell a particular door that access to it can be granted based on more than one security group. For example we could automatically give every sysop access to the RemFiles door by setting an alternate access level of one in group 3 in addition to the above level 10 in group 4. This way, whenever someone is given sysop status by setting their group 3 level to 1 they automatically get access to the remote sysop door. Consider the following accounts:

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.

1.48 icebbs_demo.guide/Number of groups

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.

1.49 icebbs_demo.guide/DOS shell access

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 could just give themselves this level anyways so this is what the shell server looks for. So on AMUC's system a full sysop would have 255 255 255 255 255. Note that if you have no S:IceBBS.MAXACCESS file set you will have to specify 255 as the access level for all 16 groups before a user can get access to the DOS shell.

1.50 icebbs_demo.guide/Setting levels

Setting access levels

=====

To set the access levels for a particular user the sysop uses the edituser command or the eu program.

See Access and Edituser for more information on tools to control user access.

1.51 icebbs_demo.guide/Privileges

Privileges

=====

Starting with version 47, IceBBS includes several forms of system privileges. The first are the controls to enable or disable the various built in clients and servers in the user's IceTERM program. Each of the buttons of the main control pannel of IceTERM can be selectively enabled or disabled, making it possible for you to set up a default IceBBSNEWUSER account so that new users cannot do anything until you have validated them. The edituser command is used to adjust these settings, to adjust the settings for an individual user just type:

```
edituser -mBITS "user name"
```

and the edituser command will walk you through the various functions that can be enabled or disabled.

The second form of privilege control is the number of login sessions per day for a given user. Once a user has reached his account's login limit the BBS will not allow him to connect for more than 30 seconds until the next day is reached.

The third form of privilege control is the maximum process quota (set by NPROC from edituser). This is the maximum number of simultaneous tasks the user can perform on the BBS. Once he has reached this limit the BBS will not allow him to start doing something else until one of the current processes completes.

The fourth form of privilege control is in the download ratios. Through edituser you can assign maximum ratios of downloads to uploads for each user based either on byte counts or file counts. Additionally you can choose to disable these ratios entirely or to suspend a particular user's ability to download anything at all.

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.

1.52 icebbs_demo.guide/Time limits

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.

1.53 icebbs_demo.guide/New users

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 called IceBBSNEWUSER, if this exists then the BBS copies the contents of the new user account into the newly created account file. If the IceBBSNEWUSER account does not exist then the BBS uses its own built in

set of default settings. If you wish to have complete control over what a new user can and cannot do you should set up a IceBBSNEWUSER account, which you can do by logging in under that name or by using the account creation facility of the mail program.

1.54 icebbs_demo.guide/IceBBS.ACCESS files

The IceBBS.ACCESS Files

These files are written by the access command (see Access) 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.

1.55 icebbs_demo.guide/IceBBS.SHELLACCESS file

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 DOS shell).

1.56 icebbs_demo.guide/IceBBS.MAXACCESS file

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 Edituser) and access (see Access).

1.57 icebbs_demo.guide/IceBBS.MSGID file

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 vmsg.Expansion[4] (see VMSG struct) 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 maintainance of this file is completely automatic you do not have to do anything.

1.58 icebbs_demo.guide/TRIM.LASTTRIM file

The S:TRIM.LASTTRIM File

This file is written and maintained by the trim utility (see Trimming SIGs), 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.

1.59 icebbs_demo.guide/IceBBS.SIGS file

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.

1.60 icebbs_demo.guide/IceBBS.NEWSIGS file

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.

Newsigs file format	SIG name, storage and access parameters
Email SIG name	Required for private email
SIG access control	Restricting access to SIGs

1.61 icebbs_demo.guide/Newsigs file format

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:

```
EMail      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 file (1) 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 replys 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. These access levels work the same way that the access levels for the file system work.

The last two numbers specify a security group number and a level within that group that a user must have before he can use the moderator functions in the terminal program's SIGs tool. The moderator functions are things like message move and delete.

----- Footnotes -----

(1) The name will have either .LNK or .DAT appended to it.

1.62 icebbs_demo.guide/Email SIG name

The Email SIG's Name

=====

If you want your BBS to provide email you must have a SIG called EMail. This name is special and it cannot be changed as this SIG is structured slightly differently internally. Also all the other sysop

tools that create and process email must be able to locate this SIG to do their work.

In the example I have given everyone access to the email SIG (by setting at least one access level to zero). This is not strictly necessary, if you want to run a BBS where access to email is only available to certain groups of users you can just adjust the access levels.

1.63 icebbs_demo.guide/SIG access control

SIGS Access Control
=====

The IceBBS SIGs system has a multilevel access control facility that works on a SIG by SIG basis. Each SIG has a variable number of access level numbers which should be between zero and 255. These are the access levels needed to download messages from this SIG. The access control is the same scheme used for controlling access to file directories, that was discussed earlier (see Access control). Briefly, the BBS SIG system will compare the access levels set in the user's account file to the levels defined for a particular SIG in the S:IceBBS.NEWSIGS file, if there is at least one case where the user's level is greater than or equal to the corresponding level in the S:IceBBS.NEWSIGS file then the BBS will allow the user to read and write to that SIG.

In the above example S:IceBBS.NEWSIGS file the message areas: Amiga and General will be available to everyone, since all users have at least access level zero in all their groups. If a user has been given access 6 1 3 0 then he will also be able to read the Test_SIG but not the News SIG since none of his levels are high enough to access that group. Note that the General group is available to all users, regardless of what their access levels are set to, since the third security group's level has been set to zero and all users will have at least that level.

The above example shows only four access levels per SIG. For versions of IceBBS prior to 47 only four levels were allowed for SIGs. Versions 47 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 IceBBS.MAXACCESS file).

1.64 icebbs_demo.guide/IceBBS.BINIT file

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 below (1):

```

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.

Line 1	Log file name
Line 2	Conference history depth
Line 3	Error message directory
Line 4	User account directory
Line 5	User email directory
Line 6 and 7	Old feedback message base
Line 8	BBS system name
Line 9	Monitor history depth
Line 10	First ASCII menu file name
Line 11	Directory for ASCII menus
Line 12	Hashed access control directory
Line 13	Auxillary file description directory
Line 14	Door server directory
Line 15	Blocks to pre-read from disk
Line 16	UUCP sendmail command name
Line 17	Serial port and script configurations per line

----- Footnotes -----

(1) The line numbers are not part of the file, and + characters

indicate the line has been split at that point.

1.65 icebbs_demo.guide/Line 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 for each user connection--and will contain the time and user name name of each login and logout event along with the connection modem string and the baud rate. It will also identify ASCII mode login events. And the commands that ASCII users issue may also be logged.

1.66 icebbs_demo.guide/Line 2

Line 2
=====

This line contains a single number which specifies the number of conference messages that will be kept on the system. These messages are kept in RAM so if you are tight on RAM keep the number down. The maximum space taken by a single conference message is 256 bytes. See also Confer.

1.67 icebbs_demo.guide/Line 3

Line 3
=====

This line contains the name of a directory where the various IceBBS error message files are kept see Error messages. If you wish to customize the error messages you can edit these files.

1.68 icebbs_demo.guide/Line 4

Line 4
=====

This line specifies the name of the directory in which the user account files are kept. There will be one file created per user, but since these files are only (1) accessed by name the number of files in this directory is not important. Note that it is not a good idea to

store other files in this directory since the edituser program (see Edituser) will try to process them.

----- Footnotes -----

(1) With the exception of the edituser program.

1.69 icebbs_demo.guide/Line 5

Line 5
=====

The fifth line contains the directory which will be used to hold the user email directories. The BBS software will create the individual user directories as it needs them, but this directory must be in existence when the BBS is run. The individual user directories are given the same names as the user account files. The edituser and checkmail utilities can be useful when trouble shooting user email problems.

1.70 icebbs_demo.guide/Line 6 and 7

Lines 6 and 7
=====

The sixth and seventh lines contain the names of the two files in which the feedback (see Mail) 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.

1.71 icebbs_demo.guide/Line 8

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.

1.72 icebbs_demo.guide/Line 9

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 Monitor) 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.

1.73 icebbs_demo.guide/Line 10

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 ASCII menus for more information on this.

1.74 icebbs_demo.guide/Line 11

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.

1.75 icebbs_demo.guide/Line 12

Line 12
=====

Line twelve specifies the directory used for auxillary hashed access control files (see HashAccess). These can be used to add access restrictions to directory trees located on read-only media. The bulksplit and bulkjoin tools can be used as an alternative to this.

1.76 icebbs_demo.guide/Line 13

Line 13
=====

Line thirteen specifies the directory used to store additional file descriptions for files that are located on read-only media. The bulksplit and bulkjoin tools can be used as an alternative to this.

1.77 icebbs_demo.guide/Line 14

Line 14
=====

Line fourteen specifies the directory that is used to store the door server programs (see Doors) and their icons. It is a good idea to make this directory accessible from the WorkBench (as a drawer icon), this way a new door can be installed at any time by just dragging its icon into the drawer and the tool types that allow you to control the doors (1) are easily accessible.

----- Footnotes -----

(1) Particularly the access control tool type.

1.78 icebbs_demo.guide/Line 15

Line 15
=====

Line fifteen contains a single number, which is the number of blocks that the BBS reads at a time from disk when sending files to users. The blocks are 200 bytes (or so) each, so a value of 50 will cause the file sender to load about 10K bytes at a time. If you are in a heavily loaded system and have spare memory, increasing this number will help. Where increasing this number really helps is when you have a multi-line BBS with CDROMS attached (see CDROM drives), since CDROMS are such slow devices (when seeking) you want to minimize the number of seek and read operations. This is especially true when you are using a CDROM jukebox type drive (1) where the disk that is currently in the drive may have to get swapped often.

----- Footnotes -----

(1) The DRM-600 and DRM-604X from Pioneer are examples of this drive type.

1.79 icebbs_demo.guide/Line 16

Line 16
=====

Line sixteen is used to specify the command to run to send email into the internet (see UUCP). 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.

1.80 icebbs_demo.guide/Line 17

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.

Fields 1 to 11	The first 8 parameters
Fields 12 and on	The remaining parameters
Use with TrapDoor	Configuration for use with TrapDoor
Polling script	Getting a line to run a polling script (UUCP)
Normal line	Nothing fancy happening on this line

1.81 icebbs_demo.guide/Fields 1 to 11

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 modems. The importance of using handshaking with the higher speed modems cannot be overemphasized,
 - * this selects the type of modem that is being used, it may be either auto for the normal auto-baud rate changing modems (ie. when the modem gets a CONNECT 1200 it expects to talk to the computer at 1200 baud), or else it is fixed for the more advanced modems which can keep the computer to modem serial link at the same speed regardless of the actual modem to modem connection speed,
 - * this specifies the modem result code that is used to enable the faster IceBBS protocols for MNP or V42 connections. The REL is for a GVC or older Supra modem, and the ARQ is for a US Robotics Dual Standard or newer Supra modem. To disable the high speed protocol all that must be done is to set this field to some string (such as XYZ) that the modem does not return. Currently(1) it is recommended that you do not use this high speed protocol, so set this field to XYZ,
 - * this is the first of two strings that are to be sent to the modem to reset it, no spaces are allowed, up to 63 characters long,
 - * this is the second of two strings that are to be sent to the modem to reset it, no spaces are allowed, up to 63 characters long,
 - * this is the name of the directory where FidoNet email messages sent by users will be placed until the email2cut program is run,
 - * this is the name or other identifier of this line. This will be displayed by the monitor tool. Normally one would put either the phone number or type of modem here. This can be up to 31 characters long, no spaces are allowed.

----- Footnotes -----

(1) v54 of IceBBS.

1.82 icebbs_demo.guide/Fields 12 and on

Additional fields

The remaining fields on the configuration line depend on the additional functions that this line must perform. There are currently three distinct modes of operation available as illustrated in the example.

1.83 icebbs_demo.guide/Use with TrapDoor

Example line 17 (Use with TrapDoor)

Line 17 is the case for a system that is using TrapDoor (or WPL (1)) to answer that line:

```
10200 4800 -2 IceBBS2
```

The first two numbers are the time and duration of a bbs not available period, this starts at 10200 seconds past midnight (02:50) and runs for 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.

----- Footnotes -----

(1) WPL replaces the now obsolete welmat.

1.84 icebbs_demo.guide/Polling script

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).

The last number is the time of the day at which the script file must be run and any user on that line will be preempted to do so. This has been set to 00:15, as it was not important in this application.

The last parameter is the name of the script file that is to be run.

1.85 icebbs_demo.guide/Normal line

Example line 19 (a normal line)

Line 19 is an example of an ordinary line which does not do any running of scripts and for which the BBS (not a front door program) manages the modem.

```
-1 -1 -1
```

in this case the last three numbers are just -1.

1.86 icebbs_demo.guide/IceBBS.DIRS file

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 Access) 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 scan (1).

----- Footnotes -----

(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).

1.87 icebbs_demo.guide/IceBBS.HAIL file

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.

1.88 icebbs_demo.guide/IceBBS.ARBITRATE file

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 RAM(1) 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 Line 15). 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.

----- Footnotes -----

- (1) Which in the case of both AsimCDFS and Xetec might save you

quite a bit.

1.89 icebbs_demo.guide/IceBBS.NOASCII file

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.

1.90 icebbs_demo.guide/IceBBS.FIDOCFG file

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

1.91 icebbs_demo.guide/ASCII menus

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 Line 10) of the S:IceBBS.BINIT file.

If you do not want to have any ASCII side to your BBS then see IceBBS.NOASCII file.

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 ASCII protocol.

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.

ASCII account	Modify and display account settings
ASCII alias	Define new commands
ASCII bye	Logout command
ASCII cd	Change directory command
ASCII chat	Enter into a chat with the sysop
ASCII confer	Enter into the multi-user conference
ASCII ctrl	Define hot keys
ASCII describe	Show descriptions about a file
ASCII dir	Directory command with various options
ASCII display	Type a file to the user's screen
ASCII download	Select a file and download it
ASCII email	Enter the email sub-system
ASCII feedback	Read and write messages in the old feedback system
ASCII goto	Exit the current menu and start another
ASCII lock	Prevent the user using builtin commands
ASCII login	Get the user's name and password
ASCII news	Show files in the news directory
ASCII password	Change your password
ASCII prompt	Define the menu text
ASCII protocol	Select another protocol
ASCII SIGs	Read the new SIGs and Email
ASCII term	Attempt to start IceTERM protocol
ASCII top	Change directory to the list of root directories
ASCII upgrade	Cash in a coupon for more access
ASCII upload	Start an upload of a file
ASCII users	Show who else is online
ASCII who	Show information about a certain user
ASCII xup	Upload a file with an XPR protocol
ASCII xdown	Download a file with an XPR protocol

1.92 icebbs_demo.guide/ASCII account

Account
=====

Not yet implemented.

1.93 icebbs_demo.guide/ASCII alias

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.

Aliases also have a primitive form of parameter substitution, any additional text that the user types after the alias command will be saved and then appended to the alias substituted command allowing you to alias the dir command as d and then still allowing the user to type d df0: to effect a dir df0: command.

The syntax is:

```
alias perm|temp user substitute
```

where, alias is the name of the command, followed by either a perm or temp keyword to define whether this is a permanent alias or a temporary one. The user string (a single word only) is a command name that the user will type to trigger the 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 alias 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 alias.

1.94 icebbs_demo.guide/ASCII bye

```
bye  
===
```

Causes the BBS to log the user out and hang up the line.

1.95 icebbs_demo.guide/ASCII cd

cd
==

The cd command allows the user to walk around the BBS file directory tree, in a similar fashion to the AmigaDOS CD command. The cd command respects the access limits that the sysop has placed on the file tree. It only allows the user to start in the directories listed in the S:IceBBS.DIRS file and will allow him to step into any subdirectories of those root directories, but only if he has the appropriate access level.

The current implementation only allows one level of directory change per command. It will also trap all attempts to use the : character and will prevent the user from accessing private parts of the system through excessive use of the / character.

1.96 icebbs_demo.guide/ASCII chat

chat
====

Not implemented yet.

1.97 icebbs_demo.guide/ASCII confer

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.

1.98 icebbs_demo.guide/ASCII ctrl

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.

1.99 icebbs_demo.guide/ASCII describe

```
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 the file (in bytes), and the long description (if any). If the file does not have any of this information attached to it, the BBS then displays the nofiledesc error message file (1) to the user.

----- Footnotes -----

(1) Found in the ASCII menu files directory.

1.100 icebbs_demo.guide/ASCII dir

```
dir
===
```

The dir command displays the list of files and subdirectories in the current BBS directory. This command takes five optional parameters

(all can currently be abbreviated to their first letter),
time
 shows the AmigaDOS date and time stamp for the file,

size
 shows the AmigaDOS size (in bytes) of each file, note this may be
 slightly larger than the actual file size reported by the describe
 command,

dirs
 shows only subdirectories within the current directory, no
 information about files are shown,

files
 shows only the files within the current directory, no information
 about subdirectories are show,

new
 causes all the items to be sorted in date order (newer files are
 displayed first, allowing the user to use a CTRL-C to abort the
 listing once old files are being listed.

1.101 icebbs_demo.guide/ASCII display

display
=====

This function displays a file to the user, it will only show files that are in the ASCII menu's directory so the user cannot wander around your system displaying files at will. Note that this means that any user can display both the menu definition files and the files that just contain text for display purposes. If this poses a problem (1) then you can use the lock command to prevent the user from executing any command except those that are currently defined by a temporary alias.

----- Footnotes -----

(1) It might be best for me to just create a separate directory for files that will be displayed by the display command

1.102 icebbs_demo.guide/ASCII download

download
=====

This command takes a file name as its sole parameter and starts an Xmodem (1) download of that file to the user's machine. The user can abort an Xmodem transfer by typing a string of CTRL-X characters into his terminal, or by breaking the connection.

----- Footnotes -----

(1) Xmodem is the only file transfer protocol hard coded into IceBBS's ASCII mode. Support for other protocols (including ymodem and zmodem) is available through the external XPR protocol libraries (see ASCII protocol).

1.103 icebbs_demo.guide/ASCII email

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 ASCII SIGs for more information.

1.104 icebbs_demo.guide/ASCII feedback

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 forward 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 redisplay 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.

A typical set of aliases for the message system might be:

```
alias temp n feedback next
alias temp p feedback prev
alias temp g feedback goto
alias temp l feedback last
alias temp f feedback first
alias temp a feedback again
alias temp e feedback enter
alias temp r feedback reply
ctrl temp m feedback next
ctrl temp p feedback prev
ctrl temp f feedback first
ctrl temp l feedback last
```

This set binds all the various feedback command functions to single ASCII characters, as well it implements a number of them as control keys, including read next message as the return key.

1.105 icebbs_demo.guide/ASCII goto

goto
====

This command takes one parameter, the name of the new menu file that should be run. The ASCII menu system allows the user to move between menus as he pleases, subject to the links between them that the sysop has defined.

If a goto command appears in a menu file where it will be executed it causes IceBBS to immediately change to the next menu file. This can be useful when you want to establish a few default aliases and then switch to a common menu file. An example of this is file which is the first menu file executed upon ASCII login on my BBS:

```
alias perm logout bye
alias perm lo bye
alias perm quit bye
alias perm help display help
alias perm new goto new
alias perm main goto main
alias perm term goto getterm
```

```
display welcome
login
goto main
```

Additionally, the goto command will often be found in alias commands where it will be executed when the user decides to select a different menu.

1.106 icebbs_demo.guide/ASCII lock

```
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.

1.107 icebbs_demo.guide/ASCII login

```
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.

1.108 icebbs_demo.guide/ASCII news**news**

====

Not yet implemented.

1.109 icebbs_demo.guide/ASCII password**password**

=====

This command takes no additional parameters and will allow the user to change his password. It will first prompt him for the old password and then prompt him for the new one twice. The password will only be changed if the old password was entered correctly and the same new password was entered twice.

1.110 icebbs_demo.guide/ASCII prompt**prompt**

=====

This command takes one parameter, the name of a file that is to be displayed to the user everytime he executes a command in the current menu. This is the file that you use to define the actual menu text that is displayed to the user. An example of a prompt file might be:

```
----- IceBBS Main Menu -----
Return here with "main"   Get help with "help"
    m = feedback messages
    f = files              term to get TERM
    ! = conference        bye to logout
    new = goto new files   quit to logout
-----
```

Note that you should keep these files short, since they are going to be sent to the user a lot of times, and if your editor only ends lines

with line feed characters you should probably add carriage returns to the line ends as well.

Also note that if you wish to have multiple sets of menus, perhaps for new users, advanced users and special groups you can. All you must do is have an upper level menu that allows the user to select the set he wants to work with and then use a goto command to jump to the first menu in the appropriate set.

1.111 icebbs_demo.guide/ASCII protocol

protocol
=====

The new XPR upload-download system requires that you install some XPR libraries (of the protocols you want to run-such as zmodem) in your libs: directory. These libraries always start with XPR and include these:

- * xprxmodem.library for XModem file transfers, this may work better with some PC terminal programs than the XModem that is built into IceBBS,
- * xprymodem.library for YModem file transfers,
- * xprzmodem.library for ZModem file transfers.

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.

1.112 icebbs_demo.guide/ASCII SIGs

```
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,

prev
    non-threaded read previous message,

high
    return to high water mark,

first
    jump to the first message in the current SIG,

last
    goto last message in the current SIG,

up
    threaded reading: go up a message (ie. to the parent),

down
    threaded reading: go down a message (ie. to the child),

thread
    threaded next message,

back
```

```
        threaded backup (reverse or previous) message,

goto
    goto a message by date    (not yet implemented),

enter
    enter a new message,

reply
    reply to the current message,

again
    type the current message again,
```

To maintain the users subscription list these are the commands:

```
sub add signame
    add the named SIG to the subscription list,

sub drop signame
    remove the named SIG from the subscription list,

sub full
    shows the full list of SIGs the user has access to on this BBS,

sub
    shows the list of SIGs the user has subscribed to.
```

To move between sigs:

```
change next
    goto next SIG in subscription list,

change prev
    goto previous SIG in subscription list,

change to signame
    change to SIG by name.
```

It is recommended that you build a set of menus that bind these commands through a series of simple prompts to keys using the alias (see ASCII alias) and ctrl (see ASCII ctrl) commands.

1.113 icebbs_demo.guide/ASCII term

```
term
====
```

Not yet implemented.

1.114 icebbs_demo.guide/ASCII top

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.

1.115 icebbs_demo.guide/ASCII upgrade

upgrade
=====

This command (which takes no arguments) will prompt the user for the serial numbers of an upgrade coupon (see Coupon system). 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.

1.116 icebbs_demo.guide/ASCII upload

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.

1.117 icebbs_demo.guide/ASCII users

users
=====

Not yet implemented.

1.118 icebbs_demo.guide/ASCII who

who
===

1.119 icebbs_demo.guide/ASCII xdown

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 ASCII protocol). 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.

1.120 icebbs_demo.guide/ASCII xup

xup
===

This is the command to upload a file using the currently selected XPR protocol (see ASCII protocol) from the user to the BBS. Note that if the XPR protocol supports wildcards then they can be used to transfer a batch of files.

1.121 icebbs_demo.guide/Error messages

Error Message Files

There are about 30 text files which must reside in the error messages directory (1). 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 dimensions (2). 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 Time limits,

cantfindnewsitem

this message is sent by the news system if one of the news bulletin files cannot be found, see News bulletins,

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 Indirected storage,

emailfileswaiting

if a user has files waiting in his email files directory when he logs in this will be sent, see EMail directories,

emailwaiting

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

exceededbratio

this is sent when a user attempts to download a file that would exceed his bytes downloaded to bytes uploaded ratio, see Privileges,

exceededddaily

this message is sent to the user if he attempts to start a new download when he has already downloaded more bytes in this day than his account's daily download limit allows, see Privileges,

exceedednratio

this is sent when a user attempts to download a file that would exceed his number of files downloaded to number of files uploaded ratio, see Privileges,

exceededquota

this is sent when a user tries to start more tasks than he is allowed to, see Privileges,

filenotdeleted

this is sent when a user attempts to delete a file he is not allowed to. Normally users can only delete files that are in their private emailed files directory,

imposter

this message is sent to any user who attempts to access IceBBS under an existing user's name, but with an incorrect password, see Edituser,

linenotopen

this is sent when a user tries to connect to a line that is currently in its shut down state, for example when it is FidoNet mail hour, see FidoNet and Use with TrapDoor,

motd

this message is sent to all users (except new users) whenever they log into IceBBS, use it to send a message of the day. See MOTD

which describes the process of producing a version of the motd which is tailored to the user,

newuser

this is a special welcoming message sent to all users new (first time connections), useful to send some helpful hints about the system,

nodlallowed

this is sent when a user who has been denied access to the get files tool attempts to use it,

nodooraccess

this is sent to a user who has tried to use a door to which he has not been given access to,

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 Doors,

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 File format,

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 Line 16,

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 DOS shell

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 Preview door,

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 RemFile door,

toomanycalls

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

unknownuser

this message is sent by the email subsystem when it finds that the user is trying to send mail (or a file) to a user who is not registered on the system. This can happen if the user is using an old version of the user list and the account that he is trying to send email to has since been deleted,

The only thing about these files that is at all tricky is that the first line of each file must only contain four numbers. These are used to specify the size of the viewing box window that will be opened on the user's end of the connection. These numbers specify the position on the screen of the top left corner of the window (in pixels from the left edge and pixels down from the top) and the width and height of the window also in pixels. An example message file follows:

```
120 10 400 180
#####
#
#           Welcome to IceBBS, New User!      #
#
#   As a new user of the system, your         #
#   access level is restricted and your        #
#   access time is limited to 1 hour per      #
#   call.                                     #
#
#   Also, your message counter has been       #
#   fully reset, so there are about 600       #
#   messages to read... If you just want     #
#   to skip them all, so you only see new    #
#   messages the next time you log in you    #
#   should select the message section, wait  #
#   for the first message to arrive and      #
#   then hang up the modem.                  #
#
#####
```

Note that the box formed by the # characters are actually part of the file that will be sent to the user, you don't have to include them in your own files. Also note that it may be possible to crash a caller's system (3) by specifying unreasonable values for the initial viewing window's position and size, so keep those dimensions to within the standard workbench size which is 640 pixels wide and 200 pixels high.

----- Footnotes -----

(1) See the S:IceBBS.BINIT file chapter for information on how to define the location of this directory. The default is ErrorMessage 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.

(3) This should no longer be possible, so let me know if you ever do this.

1.122 icebbs_demo.guide/MOTD

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.

Makemotd	Customizing the message of the day
Motdinfo	Adding user account info to the motd
NewSince	Adding a list of new files
IsNetUp	Checking to see if a PPP connection is up
Append	Building up the message of the day

1.123 icebbs_demo.guide/Makemotd

Customizing

=====

The login message that is sent to IceTERM users can be customized (1) 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:

```
.key tfile,user
stack 10000
copy dh2:errormessages/motd <tfile>
dh2:bbs/util/motdinfo <tfile> <user> -1
dh2:bbs/util/newsince <tfile> <user> -2 Files:New
```

This script first copies the standard motd file into the custom file, then it calls on the motdinfo command to append some account limit information to the output file (the tfile), and then it uses the newsince command to scan the new files directory and add the names of any files that are new (to this user) to the output file. When this is done the script file will return and the BBS will read the output file, send it to the user (in the form of a standard scrollable text list) and then delete it.

It is certainly possible to write other utilities that add additional information to this dynamically built message of the day. For example a fortune cookie program or additional statistics on BBS usage.

----- Footnotes -----

(1) With version 47 and higher of IceBBS.

1.124 icebbs_demo.guide/Motdinfo

Motdinfo
=====

This command is intended to be called from a makemotd script and to add some information about the user's BBS account to the message of the day.

1.125 icebbs_demo.guide/NewSince

NewSince
=====

This command is intended to be called from a makemotd script to add a listing of the files that are new on the BBS since the user last logged in. It does this by scanning a directory that you specify for files with an AmigaDOS time stamp newer than the last time the user logged in. Any the names of files it finds will be appended to the output file and depending on the level of detail desired file dates and sizes can also appear.

1.126 icebbs_demo.guide/IsNetUp

IsNetUp
=====

This is the command you can use to check to see if a PPP internet link (or other device-type link) is up and running. It is designed to be part of the makemotd script file so that it adds a message about the current status of the network to the message of the day.

1.127 icebbs_demo.guide/Append

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.

1.128 icebbs_demo.guide/Account

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 Edituser). 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.

1.129 icebbs_demo.guide/Edituser

Edituser

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 EU), the coupon system (see Coupon system) and the chapter on access controls (see Access control).

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 previous (2) 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.

----- Footnotes -----

(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.

1.130 icebbs_demo.guide/EU

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 Edituser door), 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 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.

The one thing that is a bit tricky is the security section-as this is a list. You edit the security area by clicking on the line you want to change and then editing the access level in the string requester at the bottom. To check mark the access level so that the apply template function will change that row of a user's account you double click the appropriate row in the list. This will cause a little left pointing triangle to appear on the right of the row indicating this item will be applied to a user's account. This is shown in the following figure, where access levels 0 and 5 have been make active in the new.template:

The templates are stored in a single directory, if you start eu from the command line you can specify the directory thus:

```
eu tmplt=some_directory_name
```

if you start eu from the WorkBench then you need to include a TMPLT=dirname tool type in the eu icon.

----- Footnotes -----

(1) Obviously a Black Adder fan.

1.131 icebbs_demo.guide/Coupon system

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 coupons(1) 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.

----- Footnotes -----

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

1.132 icebbs_demo.guide/Sharing lines

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 this(1).

Pause line	From the control window menu
Timed script files	Running script files on certain lines
Front doors	Sharing a line with a front door type program

----- Footnotes -----

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

1.133 icebbs_demo.guide/Pause line

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.

1.134 icebbs_demo.guide/Timed script files

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 IceBBS.BINIT file). 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 a time of 25200 (that's 7 hours times 3600 seconds in an hour). If you started running IceBBS at 6 AM then the script file would be called about 1 hour after you started IceBBS. However, if you started running IceBBS at some time after 7 AM the script file would not be run until 7 AM the following day. Essentially this is an offset to the first time the script is run after the BBS is started, beyond that point the second parameter is more important.

The second number is the time to wait between running the script file once and running it again. This number can be as low as 900 (wait 15 minutes between scripts) or as long as you wish. Typically you might wish to call a news feed node 4 times a day (a value of 21600, which is 6 hours times 3600 seconds per hour) or perhaps just once a day (use 86400 seconds). If you want the script file to be run just once a week (1) you would use 604800 seconds (which is 7 times 86400). This value will not pre-empt a user who is online at the time the script should be run again. The script will be delayed until the user logs out

normally, because of this the third number is provided.

The third number is a time at which the line must be shut down and the script run. This is a time in seconds after midnight, so to run a script at 2:45AM you would use 9900. This value is pre-emptive and will cause the BBS to terminate the session of any user who is on that line when this time is reached. Note that this is done by adjusting the users session limit at the time he logs in, so if he is running IceTERM version 86 (or higher) his count down timer will reflect this. This is primarily useful for FidoNet applications where the Fido mail hour must be respected.

Strictly speaking the script name is simply a command name that you could enter at the AmigaDOS prompt. You may also wish to use the script facility to run a single AmigaDOS command, such as DIR once a day. It should also be possible to include command line parameters, in which case you need to enclose the command name and the parameters with quotation marks like:

```
"dir dh2:downloads opt a"
```

Also note that for the script to run you need to be using a shell such as the AmigaDOS 1.3 shell, or ARP's AShell which recognize script files as executable files and you need to have the script files script bit set with the protect +s command. If you do not specify the full path name to where the script file is stored it might not run, depending on your version of AmigaDOS or the presence of ARP (2).

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.

----- Footnotes -----

(1) Perhaps to call a remote site to grab new files...

(2) Those who are running AmigaDOS 1.3 with or without ARP, or AmigaDOS 2.0 should have no problems.

1.135 icebbs_demo.guide/Front doors

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 Use with TrapDoor). 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"
```

1.136 icebbs_demo.guide/FidoNet

FidoNet and the BBS

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, TrapDoor (1) is a good choice. WPL or JamMail(2) (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.

FidoNet scripts	Running script files with a line
FidoNet security	Preventing bad packets entering the system
Multiple networks	Feeding SIGs from more than one network
FidoNet commands	The various FidoNet support commands

----- Footnotes -----

(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.

1.137 icebbs_demo.guide/FidoNet scripts

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).

1.138 icebbs_demo.guide/FidoNet security

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 software (1) 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.

When the FidoNet import command is run, one of its options is to check the packets before processing to see if they are marked as secure (see Import FidoNet). If this option is enabled then only packets that are marked secure will be unpacked, any others will be left alone. The reason for this is that other nodes can still call and send mail, even though they do not have session passwords defined; however, any such piece of mail could contain bogus packets so sysop intervention in the unpacking is advised. If the secure packet check option is disabled then all packets found in the inbound directory will be unpacked and imported into the various message bases.

Note that the combination of session passwording and only processing secure packets is not really enough to protect a system against importing bogus packets. The risk here is that one of the trusted systems with which you have set up a session password could send bad packets containing messages addressed to users on your system or to an echo that system does not normally exchange with you.

The IceBBS support software does not currently provide a way around this problem.

The interested sysop might want to explore other solutions to the security problem, perhaps by using a package like PGP to ensure authenticity. For example the remote system would encrypt any packet it sends with its private key, this ensures that packet can only be decrypted by applying the remote system's public key. This guarantees that the packet did indeed come from that remote system.

----- Footnotes -----

(1) Such as TrapDoor or WPL.

1.139 icebbs_demo.guide/Multiple networks

Multiple Nets
=====

The import and export commands supply some support for acting as an echo-conference hub (either within the same net or between different zones). The primary limitation to this is that low priority mail can not be transmitted at this time.

1.140 icebbs_demo.guide/FidoNet commands

FidoNet Commands
=====

This section describes the commands that are provided with IceBBS to support FidoNet activity. Not all of these are needed for the basic installation (the minimum set is just export, email2cut and import); however, the others can be useful for diagnostics and day to day operations. To make a fully functioning FidoNet environment you will also need to install a front end mailer, such as TrapDoor or WPL.

Export FidoNet	Exporting new messages from SIGs
Import FidoNet	Importing FidoNet messages to SIGs
Viewing FidoNet packets	Examining the contents of FidoNet packets
Posting to FidoNet	Creating FidoNet message packets directly
Testing time of day	Testing the time in script files
Waiting in scripts	A replacement for the AmigaDOS wait command
Exporting netmail	Exporting email messages to FidoNet
Debugging scripts	Echoing to the internal serial port

1.141 icebbs_demo.guide/Export FidoNet

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 beginning 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-BYs 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 be rescanned the same MSGID lines will be written (see IceBBS.MSGID file).

If there is a configuration problem that prevents export from running it will return a result code of 5 which can be used to abort the script file. It returns 0 if the run was successful.

1.142 icebbs_demo.guide/Import FidoNet

Import

This program is designed to handle the FidoNet message importing process. It can unarchive arc-mail (1) and can unpack *.PKT files (2) and sort through the resulting messages and place them in the appropriate places (either SIGs for echo mail, or a user's email directory for netmail). Usually this program is run near the end of the FidoNet script (after the TrapDoor activity is finished) but before control is returned to IceBBS.

The FIDOCFG file (see IceBBS.FIDOCFG file) needs to be configured for import to work properly.

Import works by scanning the FidoNet inbound mail directory (3) for *.PKT files. It will parse the messages out of each PKT file it finds and for each message that is extracted it will decide whether the message is private netmail (in which case it is placed into the IceBBS email SIG) or an echo mail conference message. If the message is part of an echo mail conference it will extract all the control information and store that in the message's VMSG header, as well it will figure out the SIG area the message belongs to and put the message into that SIG's message base (4).

The program will strip out CR and soft CRs and convert them to LFs, as well it will break long lines into LF terminated short lines. Any LFs in the original will be discarded.

If the program finds any archived mail packets in the inbound directory it will unarchive (5) them one at a time into the TEMPDIR and then will process the resulting *.PKT files via the above steps.

All *.PKT files and archive files that are successfully processed will be deleted.

If you are receiving an echo mail conference feed then there is another feature of import that will be of interest. As import unpacks messages and places them into the correct SIG directory it will check each message 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.

----- Footnotes -----

(1) Also called compressed mail.

(2) Uncompressed mail bundles.

(3) TrapDoor usually uses MAIL:InBound for this.

(4) SIG message bases are stored in two files, the actual text and header information goes into the .dat file while the message linkage information goes into the .lnk file.

(5) Using the archiver specified by the -a command line switch.

1.143 icebbs_demo.guide/Viewing FidoNet packets

Fido

This is a debugging utility that can be used to inspect the contents of FidoNet packed message files (1) (*.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.

----- Footnotes -----

(1) To inspect compressed mail files you will need to unpack them first.

1.144 icebbs_demo.guide/Posting to FidoNet

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 zonegate (1).

Since the out of net and zone routing makes use of what is termed low priority net mail it may not work in all FidoNets and other facilities may have to be developed later.

----- Footnotes -----

(1) This has not been extensively tested at this point.

1.145 icebbs_demo.guide/Testing time of day

FidoTime

This program is designed to be called from a script file and return a level 5 AmigaDOS result (a warning level) if it is time for the FidoNet commands to run; otherwise, it returns a level zero. The AmigaDOS Warn command can then be used to check to see if the current time was between the two times specified and the appropriate section of the script file can be executed.

The program takes two parameters, both in the 24 hour clock, the first is the hour when FidoNet mail starts, and the second is the hour when it ends.

So if your site accepts FidoNet calls between 05:00 and 07:00 (ie starting at 05:00 and ending at 06:59:59) then you provide a 5 and a 7. If your system starts processing at 23:00 and ends at 02:00:00 the next day you provide a 23 and a 2.

An example script file usage might be:

```
FidoTime 3 4
IF WARN
  RUN TrapDoor answer
  WAIT 5 secs
  TrapTell "rings 1"
  TrapTell "rings 1"
  WAIT until 04:00
  TrapTell quit
ENDIF
```

It can be used for other purposes besides FidoNet, I use it to schedule when it is a good time to run the SIGs aging utility trim.

This command should probably be made to take minutes and seconds as well as the hours.

1.146 icebbs_demo.guide/Waiting in scripts

SafeWait

This is an alternative to the AmigaDOS wait command. This command takes one parameter, a time in seconds past midnight that the command should wait until before returning. This can be useful in script files because it will return immediately if the current time is already past the time the command is supposed to wait for; whereas, if you were to use the AmigaDOS wait command your script file would wait until the time arrived the next day before proceeding.

1.147 icebbs_demo.guide/Exporting netmail

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 Outmail 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.

Pkt2EMail

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

1.148 icebbs_demo.guide/Debugging scripts

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 occuring on that port.

1.149 icebbs_demo.guide/UUCP

UUCP and the BBS

UUCP (1) 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.

Finding a feed	The first step to USENET
Configuring for UUCP	Basic script files
UUCP support commands	The UUCP interface commands for IceBBS

----- Footnotes -----

(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.

1.150 icebbs_demo.guide/Finding a feed

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 (1). If you cannot find someone to feed you the news groups you are interested in then there is nothing (2) 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 publications (3) 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 are companies who will sell you access to the Internet (4) In North America some public libraries are starting to provide access to the Internet, these services are usually provided under the name FreeNet.

----- Footnotes -----

(1) A feed is a site who is willing to supply you with access to USENET.

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

(3) Such as: Boardwatch Magazine or the BBS Callers Digest

(4) Usually through PPP or SLIP connections, although some offer UUCP feeds too. The fees that ISPs charge have been dropping rapidly in the last few years.

1.151 icebbs_demo.guide/Configuring for UUCP

Configuring for UUCP

=====

Once you have a news feed and you are setting up UUCP (1) there is one additional thing to note, you must (2) run UUCP in a dial out mode if you are going to have it share one of your serial lines with IceBBS. Since you are running it in a dial out mode you do not need to use the Getty program and your script file (which will run UUCP) will only need to call the UUCico program. Nothing additional needs to be run in the background and any serial port (or device) may be used with UUCico. The

following (3) is a portion of the script file (4) that I use to obtain my USENET feed, this is run every two hours or so on my system through the IceBBS timed scripting feature on one of my lines.

----- Footnotes -----

(1) It should be possible to use AmiTCP through a PPP (or SLIP) to an ISP who has an NNTP (Network News Transfer Protocol) server to obtain your USENET feed. This is a theory, let me know if it works.

(2) Apparently WPL will accept incoming calls from both FidoNet and UUCP sources so this is not strictly true.

(3) Remember that the + character is used to indicate that a command is too long to fit on one line and has been split.

(4) An additional example is included in the distribution archive in the BBS_S directory as the file: usenet_script.

1.152 icebbs_demo.guide/UUCP support commands

UUCP Support Commands

=====

The following subsections detail the various UUCP support commands that are provided with IceBBS.

NewsIn	Importing USENET news into a SIG
UUExport	Exporting new postings from a SIG to a news group
InMail	Importing Internet email for users
FwdEmail	Special email forwarding for special users
Outmail	Exporting Internet and FidoNet email

1.153 icebbs_demo.guide/NewsIn

NewsIn

NewsIn is a conversion utility designed to move USENET news that have been received by UUCP into IceBBS SIG link and data files.

NewsIn, like the import command, will automatically make copies of messages that are found addressed to users of the BBS. These copies are placed in the users' email SIG. As some users find this feature annoying the CC: Usenet button in IceTERM is provided so they can turn it off.

1.154 icebbs_demo.guide/UUExport

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.

1.155 icebbs_demo.guide/InMail

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 Outmail) to allow replies to messages in this SIG to be sent back to the mailing list.

1.156 icebbs_demo.guide/FwdEmail

FwdEmail

This allows one to automatically forward email from the email SIG (and the old email system, and emailed files) to other accounts on other machines on the internet via UUCP.

1.157 icebbs_demo.guide/Outmail

Outmail

This program handles both the export of USENET (internet) email and FidoNet email from the email SIG. As such it can even cross route messages for you, for example: if you don't have a FidoNet feed yet you still want to be able to send messages to FidoNet sites via internet.

Info server
.....

The outmail command provides support for the information by email server (see InMail) with the -mi command switch. By using the -s mode of inmail the contents of any messages that are sent requesting information will be placed into a SIG. Now if someone replies to one of these messages that reply will only be seen locally. But if outmail is run on that SIG the reply will be forwarded to the person who requested the information in the first place.

Sharing a mailing list
.....

The -mp command switch activates outmails mailing list sharing mode. If you have several users on your BBS subscribing to the same high traffic mailing list you should consider using this(1) as it can cut down the size of the email SIG. What you do is to add another SIG to the BBS which will be used to store messages from the mailing list (for example lightwave_sig). You then subscribe to the mailing list on behalf of the BBS using a convenient user name that is not in use on the BBS, say common_account. You then run inmail to scan the UUMAIL: directory looking for messages addressed to name you subscribed to the mailing list under (ie. common_account). You then have inmail place these into the common SIG, thus:

```
inmail -s common_account lightwave_sig
delete uumail:common_account
```

This places the message received from the mailing list into a single SIG that your users can subscribe to. When a user makes a posting into this SIG, or when he replies to a message in the SIG you want those 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.

Hosting a mailing list

.....

The difference between sharing a mailing list between several users on the same BBS and hosting an entire mailing list(2) 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 to(3).

----- Footnotes -----

(1) Note: at the time of writing this has not been heavily tested so take care.

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

(3) An almost useless feature at the present as inmail does not handle receiving mail from a FidoNet address yet, but it's planned for.

1.158 icebbs_demo.guide/Internet tools

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 UUCP) and FidoNet (see FidoNet) 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 Wide Web, perhaps the most hyped, and certainly fastest growing software in the world. A hypertext based information retrieval system that spans the globe and anyone can use and contribute data to.

At the current time(1) 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.

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.

There is now a rapidly growing industry called Internet Service Providers (ISPs) whose business is to sell connections to the Internet. So finding a point of access is not difficult (at least in major cities in North America). However such access is not inexpensive, but the costs are dropping very rapidly. In North America I expect the cost of a 14400 or 28800 baud full time connection to the Internet to eventually drop to about \$70-100 per month, unless the phone companies start providing permanent connections, in which case(2) the prices could drop still further. However, I expect the phone companies will attempt to make fat profits for a while at least.

An ISP can provide you with varying levels of connection, each of these would be at a different cost. The most common is a direct modem connection using PPP or SLIP as the protocol. These are available at different speeds, as permanent or dial up on demand, and may be billed

by the month, day or hour. At a higher price are ISDN and special dedicated line connections. These can take you to 64K or even 128K baud rates. Beyond this are special leased lines which are what the ISPs use to get their feeds. The cost of these is pretty high (perhaps \$1000.00 per month and up). Names like T1 and T3 are types of leased lines.

If you happened to live beside an ISP you could even just sling an ethernet cable over the fence.

If you live within about a half mile of an ISP you might be able to bypass the phone company altogether by using a pair of radio modems. One such pair was written up in the November 1994 issue of Boardwatch magazine (pp. 26-35). This modem is manufactured by:

Metricom
980 University Ave
Los Gatos, CA 95030.

Voice: 408-399-8200
FAX: 408-354-1024
Email: info@metricom.com

These Ricochet wireless modems use the FCC Part 15 spread spectrum radio band (902-928MHz) and with less than 1 watt of output they do not require licensing in the USA. They can be pretty fast, Boardwatch reported about 77000 baud, but a pair of them will cost about US\$1000.00, which should pay for itself in under two years through phone savings. And since they are replacing a pair of high speed modems (say about 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.

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 IceIRC). 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.

File transfer

=====

IceFTP (see IceFTP) 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.

Is the net up?

=====

This is a question a lot of your users will start asking you if you are only using an intermittent dial up connection. To answer this question automatically as part of the MOTD generation the program IsNetUp was written (see IsNetUp).

----- Footnotes -----

(1) April 16, 1995.

(2) Since the connection would only need one phone line from you to the switchboard, rather than three: from you to the switchboard and then from the switchboard to to the ISP and then from the ISP back to the switchboard.

1.159 icebbs_demo.guide/Searching

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.

Find	The sysop's stand alone file finder
MUIFind	A stand alone finder with a MUI GUI
FindScan	First phase of the database builder
MySort	Faster sorting of large files
FindStat	Prints some statistics on the database
FindBuild	Builds the final database files
Find databases	What's happening inside
Find limits	Limitations of the system

1.160 icebbs_demo.guide/Find

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:

A & B

The & (and) operator takes the lists of words contained in the two files and produces a new list containing only the words in both files. This is the set intersection operator,

A | B

The | (or) operator takes the lists of words contained in the two

files and produces a new list containing all the words in both files. This is the set union operator,

A - B

The - (minus) operator takes the lists of words contained in file A and removes from that list any words which are also contained in file B. This is the set difference operator.

As well parentheses are available to allow you to group several operations:

A & (B - C | (D & F))

will evaluate D & F first to produce X,

A & (B - C | X)

it will then evaluate B - C to produce Y

A & (Y | X)

it will then evaluate Y | X to produce Z

A & Z

and will finally evaluate A & Z.

If there are no parentheses then evaluation proceeds from left to right as was the case when it evaluated B - C | X above.

1.161 icebbs_demo.guide/MUIFind

MUIFind
=====

MUIFind is a GUI based file finding utility for use on IceBBS systems and for regular users who have an AMUC CDROM (see AMUC) and its cross reference database. MUIFind uses MUI for its user interface, so you will need to get a copy of MUI installed to use it.

Installation

There are three cases for installing this utility:

1. if you run an IceBBS system just run MUIFind and it will read your s:icebbs.find file to discover the location of your databases.
2. if you have AMUC CDROM 1 you need to get a copy of the database files for that CD, these are present on AMUC CDROM 2 or they may be downloaded. There will be 5 database files named something like:

amucl.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 dhl: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:

```
dhl: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:.

Usage of MUIFind

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.

A combined search on several keywords can also be done. The search string will be parsed by MUIFind and each word within the string will be searched for in the database(s). This produces a table of files for each word. The finder 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 the same as those described previously in the find command (see Find).

When using the lower area to search for keywords by using AmigaDOS patterns you enter the pattern into the string gadget and hit return. For example the pattern:

```
delay#?
```

will cause it to search for all the keywords that start with delay. If you enter:

```
#?delay#?
```

MUIFind will look for all the keywords that contain delay. You can use the full capabilities of AmigaDOS pattern matching here.

```
Search by file name
.....
```

When the Search by file name mode of MUIFind is selected then there is a single string requester at the bottom of the window. Into this one types an AmigaDOS wild card string and upon hitting return MUIFind will search the databases for any file names that match the pattern. Full AmigaDOS pattern matching is available so that typing:

```
mui#?(1|2|abc)#?
```

will cause a list of all the file whose name starts with mui and follows that by a 1, 2 or abc later in the name. This gets particularly interesting when you start including multiple levels of parenthesis and use the not operator: ~.

----- Footnotes -----

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

1.162 icebbs_demo.guide/FindScan

```
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 CDROM(1) 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.

----- Footnotes -----

(1) The AMUC CDROMs come with a pre-built database.

1.163 icebbs_demo.guide/MySort

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 files need to be sorted so that MUIFind's keyword searching feature works better. The syntax is just:

```
mysort input output
```

where input is the name of the file to be sorted and output is the name of the file to place the results in. Since mysort does all its sorting in memory you need to have enough RAM for it to load the whole file, plus some overhead to sort in.

1.164 icebbs_demo.guide/FindStat

FindStat, trimming the database
=====

This program is used to analyse the word count information produced by FindScan so that you can select certain parameters that prune the data base. There are two reasons you might want to do this. The first is that pruning the database will save you some disk space without impairing its usefulness much. The second is that words which have a high occurrence count can cause problems for the users of your system. This is because the system sends the user the full path and file name for each match it finds, so if the user specifies a word that appears in a thousand files he's going to get a lot of output. As it happens these very common words are not very useful for a searching point of view anyway, so it makes sense to delete them from the system (they are a sort of information noise).

1.165 icebbs_demo.guide/FindBuild

FindBuild, building the database
=====

This program is responsible for building the occurrence file, which contains the tables that list which files each word in the database index occurs in. This takes very similar command parameters to the FindScan program, in fact the parameters provided should be identical with the exception of the flags.

1.166 icebbs_demo.guide/Find databases

The list of databases
=====

Both the user interface to the file finding system through IceTERM and the sysop interface through the FIND command allow for automatic searching of more than one database. The list of databases is stored in the file:

S:IceBBS.FIND

This file is a white space separated list of the path and root names of all the databases in the system. For example if you have three databases the file might look like:

```
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.

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.

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.

1.167 icebbs_demo.guide/Find limits

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 the dictionary that the sysop built the database with. With a smaller dictionary (like 10000 words) there will be many more collisions than a larger dictionary like 100000 to 1000000 words). Note that larger dictionaries only cost the sysop file storage space, they do not slow the searching process significantly.

Memory usage

The routines that perform the finding of words in a completed database use quite a small amount of memory, so should not pose a problem. The FindScan, FindStat and FindBuild routines all require a single chunk of memory which is 4 times the number of words in the database's dictionary (ie. the same size as the completed .hsh file). So for the default 100000 word dictionary you need at least 400K of free memory to run. If you want to produce a million word dictionary you will need at least 4Megs of free RAM, this usually means you must be running on a system with at least 8Megs of RAM installed.

1.168 icebbs_demo.guide/File system

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 it's access levels. This is one of the access control mechanisms in IceBBS, for more information see Access control.

File format	The IceBBS file format
Indirected storage	Separate descriptions and files
File tools	Tools to manage the file system
Listing new files	Generating a listing of new uploads
Bulk programs	Processing whole directory trees

1.169 icebbs_demo.guide/File format

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 message (1) 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 below (2)) 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: */
#define FH_CRCVALID      (1 << 14) /* the CRC is valid, ie.
                                   the file uploaded OK, Pre-89 versions of TERM
                                   and anonymous uploads do not set this bit
                                   */
#define FH_LOCN_MASK     0x03
    /* these bits are used to specify where the file is
    */
#define FH_INPLACE       0
    /* a regular IceBBS file, the file follows
    the long description
    */
#define FH_INDIRECT      1
    /* indirect access, the data following the
    long description is really the full path and
    name of the AmigaDOS file to be sent to the
    user. This is useful for CDROMs
    */
#define FH_SOMETIMES     2
    /* like FH_INDIRECT except the if the media is
    "offline" (maybe on tape or something) so
    the file request should be logged to
    a file for future processing (not implemented
    yet)
```

```

    */
#define FH_OFFLINE      3
    /* the file is not in online storage, but the
       "file" area contains the requesting info
       that is to be written to the request file.
       For example this might include FTPing or
       FREQing instructions.
    */

    USHORT CRC;          /* 16 bit CRC of the file */
    ULONG Downloads;     /* the number of times
                           downloaded */
    ULONG DescLength;    /* the length of the long
                           description */
    UBYTE User[64];      /* name of the user who
                           uploaded it */
    UBYTE ShortDesc[64]; /* the short description */
    UBYTE LongDesc[2];   /* the long description...
                           variable sized */

};

```

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.

----- Footnotes -----

(1) From the error messages directory.

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

1.170 icebbs_demo.guide/Indirected storage

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.

1.171 icebbs_demo.guide/File tools

File Tools
=====

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

BBSJoin	Making an IceBBS file out of an AmigaDOS file
BBSSplit	Extracting the AmigaDOS file
UJoin	Joining a description file with a file
AmiJoin	Joining an Aminet description file with a file
FMT	File maintainance tool
DelOld	Deleting older files
Access	Setting the access levels for a directory
HashAccess	An alternate form of access control
FileList	Making a listing of files with descriptions
Store	To help with script file debugging

1.172 icebbs_demo.guide/BBSJoin

BBSJoin

To allow the sysop to add long descriptions to regular AmigaDOS files the BBSJoin utility is provided. This utility will take an existing AmigaDOS file and a second file containing a long description and combine them into a single IceBBS format file. This program takes a number of parameters.

1.173 icebbs_demo.guide/BBSSplit

BBSSplit

To allow the sysop to extract long descriptions and regular AmigaDOS files from IceBBS format files the BBSSplit utility is provided. This utility will take an existing IceBBS file and extract from it one or both of the AmigaDOS and long description files.

1.174 icebbs_demo.guide/UJoin

UJoin

UJoin was written to automatically add descriptions to files. A more recent attempt at the same thing is AmiJoin (see AmiJoin). UJoin will scan the contents of its input directory looking for files that end in .desc. If it finds one of these it looks for the matching file without the .desc and joins the two together. The resulting combined file is copied into the output directory and the two input files are deleted. Credit for the upload can be assigned with the -c switch.

1.175 icebbs_demo.guide/AmiJoin

AmiJoin

The AmiJoin command was made to join .readme files from Aminet with their corresponding AmigaDOS files to make VBBS format files. Like the UJoin command it processes an entire input directory, looking for pairs of files with the same base name but where one file ends with .readme. When it finds a pair it extracts what information it can from the .readme and places that information into the VBBS header of the file in the output directory, it then deletes the input files. The -c flag can be used to set the uploader credit line. The command's syntax is:

```
amijoin -iINPUTDIR -oOUTPUTDIR [-cCredit_Where_Due]
```

The format of the .readme files is quite a bit more flexible than in the case of ujoin. Aminet uses a header that's rather line the RFC-822 specification for email on the Internet. The amijoin command looks for a line starting with short: and uses it for the short file description. It then places a copy of the full .readme file into the VBBS long description.

1.176 icebbs_demo.guide/FMT

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 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.

Processing modes
.....

The three processing modes cycle gadgets allow you to customize the way most of fmts 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,

Create icons too

when fmt is moving or copying it will create icons to go with the files it creates,

Output to dest

the files that are created will only be written to one destination directory,

Output to dest + NEW:

the files that are created will be written to both a destination directory and a new files directory. The new files directory is specified in the configuration editor. This makes placing a copy of the files into both the new files directory (which can be scanned by the newsince command, see NewSince) and the proper permanent directory in the file system very easy.

One destination

if all the files in the to be processed list are to be placed in one directory then this setting will only ask you for the directory

once,

Many destinations

otherwise, select this setting and you will be prompted to select a destination directory for each file in the list.

The various processing methods are described below:

CRC Test

test the files against the internal CRC value to see if they suffered damage during the upload. Note that it is possible for someone either with an old version of IceTERM, by uploading with out description or by uploading from the ASCII side of the BBS to send a file without the internal CRC value,

Browse

view what information is stored about each file, including its type and any short or long descriptions,

Rename

rename the files in the list,

Move

move the files in the list, that is copy them to a destination and then delete the originals,

Copy

copy the files in the list,

ARC Test

test the files, by using the appropriate archiver tool's test mode, to see if any are corrupt,

Edit

edit the long or short descriptions of the files in the list,

Delete

delete all the files in the list,

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.

1.177 icebbs_demo.guide/DelOld

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.

1.178 icebbs_demo.guide/Access

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 Access control) which you can assign with the edituser program (see Edituser). 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:

```
access
```

The access command will then prompt you to enter the appropriate information.

Access needs to read the S:IceBBS.MAXACCESS file (see IceBBS.MAXACCESS file) to determine the maximum number of access levels to the BBS, and if this file is not present it will default to 16 levels.

1.179 icebbs_demo.guide/HashAccess

HashAccess

This is an alternative command to the access command. It is a new addition intended to be used with CDROM (or other read only) devices to allow access control to be added. Since it is new, it has not been extensively tested yet so be careful.

1.180 icebbs_demo.guide/FileList

FileList

The filelist command is used to make listings of the files that are available for download. This command has a number of command line parameters to control how the listings are built, and it will only scan the directories to which you let it have access. Type filelist by itself to get the help screen.

1.181 icebbs_demo.guide/Store

Store

The store command is useful for debugging script files that produce temporary files that need to be saved for examination. By using a combination of store and delold one can create a rolling log of the last few days of activity, which might be useful when tracking down errors in FidoNet or USENET export trips.

1.182 icebbs_demo.guide/Listing new files

Listing New Uploads

=====

One result of IceBBS using the AmigaDOS directory system for its file storage is that it becomes difficult to provide a list of the recently uploaded files to users. The work around I have adopted for the moment is to set up an assign to a directory called NewFiles:. Add this assigned name to your S:IceBBS.DIRS file so that it is visible to users as soon as they click on the Get Files tool in IceTERM.

Then, whenever you validate a newly uploaded file, place that file in it's proper place in your directory trees and at the same time place a

copy of the file into the NewFiles: directory. Now a user can easily find out what's new on the system by just opening the NewFiles: directory and showing the files sorted by date.

1.183 icebbs_demo.guide/Bulk programs

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.

BulkSplit	Splitting IceBBS files into DOS files and descriptions
BulkJoin	Joining descriptions and DOS files into IceBBS files
BulkTest	Testing your file areas for bad files
Bulk example	An example of the bulk programs

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).

1.184 icebbs_demo.guide/BulkSplit

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).

1.185 icebbs_demo.guide/BulkJoin

BulkJoin

Since it would be a lot of work to reverse the effects of a bulksplit command the bulkjoin program was created. This command combines the separate description and file directory trees into one IceBBS file type tree.

Note that if you have any files in the source files tree (1) which do not have descriptions corresponding to them in the desc tree these files will not be found and hence will be left behind.

If you have any descriptions for which the files cannot be located then these descriptions will be left behind and nothing will get written to the combined files destination.

If you have put a IceBBS combined file into the description tree it will be detected and just copied across to the combined file tree.

Finally, if you have put regular AmigaDOS files into the description tree, these will be detected and they will be written to the combined file tree with a default IceBBS header attached.

These last two allow you to take a description tree for a CDROM (or perhaps several!) and mix into it regular IceBBS and AmigaDOS files as you want, then to make a single combined image.

----- Footnotes -----

(1) A file tree probably made by a bulksplit command.

1.186 icebbs_demo.guide/BulkTest

BulkTest

The bulktest program can be used to check files of the DMS, ZOM,

LHA, LZH, ARC, ZIP, and ZOO types en-masse and to print a report of any suspect files it finds.

1.187 icebbs_demo.guide/Bulk example

Example

Ok, so lets follow the process through step by step. Say you have a set of IceBBS type files (ie. the combined description and file in one) in a directory tree whose root is:

```
DH2:bbs/files/
```

and you want to process these so that you get the separate description files and AmigaDOS files. These will be placed in separate directory trees, we'll place the descriptions in:

```
DH2:bbs/desc/
```

and the actual files in:

```
DH3:rawfiles/
```

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 file (1) 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...

----- Footnotes -----

(1) S:IceBBS.DIRS

1.188 icebbs_demo.guide/Message bases

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 IceTERM (1) 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.

Feedback messages	The simple one SIG message system
SIGs messages	The multi-SIGs system

----- Footnotes -----

(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.

1.189 icebbs_demo.guide/Feedback messages

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 file (1) are the following structure:

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

The messagenumber is the reference number for a particular message. These numbers must be sequential (there cannot be gaps or duplicate message numbers), although the first message in the file can start with any number from zero on up. The message server software looks at the first and last record in the header file to determine where in the file is the first new message that should be sent to the user. The messagelen field is the length of the text part of the message in bytes. The fileposn is the location in the data file at which the text of the message resides. The message server loads each message to be sent to a user by seeking to the fileposn and then reading messagelen bytes from the file. A crude and simple approach.

The time field is the time that the message was posted by the user, this is stored as the number of seconds since the start of the Amiga's clock (which is January 1, 1978). This is the same integer that you would obtain from the C linker-library ctime() function.

The replyto field contains the number of a message that this message was a reply to. If this message is a new posting then this number is -1 which should be considered to be an illegal message number.

The username and topic fields are null character terminated ASCII strings. The username is the name of the user who posted the message, and the topic field is the subject heading for the message. On new postings the system obtains the subject heading by extracting the first line of the text file on the user's machine. On replies the the topic is just a copy of the original subject line.

To set up an old style message system all that needs to be done is to specify the names for the index and message files in the S:IceBBS.BINIT file (see Line 6 and 7). As soon as the second message is posted (either from a user's terminal session or via the sysop's mail utility) the two files will be created. The complete set of steps that must be fulfilled are:

- * edit the sixth and seventh lines of S:IceBBS.BINIT file to have the correct file names for the index (header) file and the data (text) file for your message base and restart IceBBS,
- * copy a TERM.info file which has your user name and password in its tooltypes array and name it MAIL.info, place a copy of this either in the directory you start the mail command from or in the S: or C: directories if you prefer to be able to run mail from anywhere,
- * run the mail command and post one (1) new message. Run the mail command a second time and post one more message.
- * check the user accounts directory it should now contain a user account 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.

----- Footnotes -----

(1) This is subject to change without notice at the moment, so don't write any utilities for it.

1.190 icebbs_demo.guide/SIGs messages

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 system (1), 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 FidoNet) and UUCP (see UUCP) 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

SIG file format	Format of the .lnk and .dat files
FirstLinkItem struct	Structure definition
LinkItem struct	Structure definition
DataItem struct	Structure definition
VMSG struct	Structure definition
Traversing links	Moving about the messages within a SIG
The message reader	The sysop's message reader
Support utilities	The extra commands for the SIGs system

----- Footnotes -----

(1) There were several user utilities for this system: read, age, VBBSReader, and GPReader, all of which are now obsolete.

1.191 icebbs_demo.guide/SIG file format

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...
```

The linkages between messages in the SIG are stored in a file named: signame.lnk where signame is the name of the SIG. The linkage file contains:

```
a single struct FirstLinkItem
a LinkItem
a LinkItem
a LinkItem
and so on...
```

1.192 icebbs_demo.guide/FirstLinkItem struct

The FirstLinkItem structure

The FirstLinkItem structure is a special case, it will always contain some padding to make it the same size as the LinkItem structure. There is only one FirstLinkItem structure, and it is always located at the begining of the SIG's data file. This structure serves as the root node of the message tree for a particular SIG.

```

struct FirstLinkItem
{
    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 */
};

```

1.193 icebbs_demo.guide/LinkItem struct

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
{
    ULONG MsgNo;    /* the number of this message
                     (actually a date of posting) */
    ULONG Parent;   /* number of this message's
                     parent */
    ULONG Child;    /* number of the first reply to
                     this message */
    ULONG Next;     /* next message at this level in
                     the tree */
};

```

```

    ULONG Prev;    /* previous message at this level
                    in the tree */
    ULONG Offset; /* location of the message in the
                    data file, this is the offset
                    from the beginning of the file
                    to the DataItem structure for
                    this message, the data item will
                    be read to determine the length
                    of the message in the file, and
                    its type (compressed or
                    uncompressed)
                    */
    ULONG Subject; /* crc of the subject string */
    ULONG Author;  /* crc of the author string, except
                    for the EMAIL SIG where this is
                    the CRC of the user the message
                    is addressed to, if they are not
                    on this BBS then it is set to
                    zero.
                    */
    ULONG Flags;   /* special flags pertaining to this
                    message, these are shared by the
                    BBS and the TERM systems, so if
                    you change one of these flags be
                    certain to preserve the other 31
                    bits. Note these are the bit
                    values.
                    */
#define LI_READ    (1L << 0L) /* off line reader
                                sets this when the user
                                has read a message */
#define LI_DELETED (1L << 1L) /* message is
                                marked as deleted */
#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
                                permanent 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.

1.194 icebbs_demo.guide/DataItem struct

The DataItem structure

Each VMSG in the data file is preceded 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)
                    */
};
```

1.195 icebbs_demo.guide/VMSG struct

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:

```
struct MessageHeader
{
    ULONG Type; /* a 4 character identifier for
                 the file type == VMSG */
#define VMSG (((ULONG) 'V' << 24L) \
             | ((ULONG) 'M' << 16L) \
             | ((ULONG) 'S' << 8L)  \
             | ((ULONG) 'G'))

    ULONG BBSDate;
    ULONG CreationDate;
    UBYTE Author[64];
    ULONG DelBy;
    ULONG ReadCount;
    ULONG Archive;
    ULONG Expansion[5];

    /* the following six length fields are used
       to indicate the number of bytes of data in
       each field.
```

Any or all of these can be zero bytes long.

The actual data appears in the file in the order:

```
        Subject, Address, Destination,  
        OldSubject, Keywords, Network.  
    */  
    ULONG Keywords;  
    ULONG Network;  
    ULONG Subject;  
    ULONG Address;  
    ULONG ReplyTo; /* this used to be: Destination */  
    ULONG OldSubject;  
};
```

where:

VMSG

a 4 character identifier for the file type, if the SIG server encounters non-IceBBS files in its wanderings it will treat them as a straight ASCII message,

BBSDate

the date at which this message was posted to the system. The date is in conventional C-library (ctime) format,

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 NewsIn, to reconstruct the thread linkages between messages as they arrive), the same functionality will be added to the FidoNet import utility too, see Import FidoNet. For messages that originate locally this field is used to store a unique message ID number generated from the S:IceBBS.MSGID file, see IceBBS.MSGID file.

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.

These fields are each the indicated number of bytes long, there are no null termination bytes, but there could be. Any or all of them could be zero.

Subject

The subject line that the author has provided - in original postings this is the subject, but in replys this becomes a secondary line. This is filled in by the NewsIn and Import tools when USENET and FidoNet messages are received.

Address

this is the name of the user the message is addressed to. The FidoNet utilities will copy this into the To: field of a FidoNet message header, if it is not supplied then the default of All will be used. On USENET messages this field is ignored if present,

ReplyTo

this is the full Internet RFC-822 Reply-To: line, if it is present this will be used as the address to which replies are sent rather than the author,

OldSubject

the length of the previous subject in bytes.

Keywords

the size of the user defined keyword space, some reader programs allow the user to enter a number of keyword topics, these are stored in the message header for future searching,

Network

the size of a network information block. This is primarily intended for FidoNet support, the import program will strip the FidoNet control lines from the message and place it in this data area, then on export it will move it back into the body of the message so that it is Fido compatible. Note that the reader should not copy the contents of this block into replies, instead it should create a zero length field for any fresh postings.

The complete USENET header is also removed from the body of the

message and placed here by newsin, in this way we can be assured that the user does not tamper with it (unless he writes a special tool).

Note that several useful features are provided in the header. The first is that the author also has an email address, this is to allow for networking. The DelBy field allows for automatic message purging on a message by message basis. Several possible uses for this exist, a moderator might read a bunch of garbage messages (or superfluous responses) and this allows him to immediately (or at any later date) have these messages deleted so that the quality of the discussion is maintained. Note that setting the DelBy field causes a soft delete of the message, the SIG server just no longer sends the message to unprivileged users, 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 replies and new postings extracted for transmission.

1.196 icebbs_demo.guide/Traversing links

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.

```
Find message by number
Find the next message
Find the previous message
Find end of file
```

1.197 icebbs_demo.guide/Find message by number

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 deleted (1) 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;

    if (MsgNo == 1L)
    {
        Seek(file, 0L, OFFSET_BEGINING);
        Read(file, link, SIZE(LinkItem));
        return(TRUE);
    }

    if (!fli)
    {
        Seek(file, 0L, OFFSET_BEGINING);
        Read(file, &localfli, SIZE(LinkItem));
        fli = &localfli;
    }

    lower = 1L; /* lower limit of the search */
    upper = fli->Number;

    /* before searching we should examine
       the lower and upper bounds to see if
       we are sitting right there...
    */

    Seek(file, lower
          * SIZE(LinkItem), OFFSET_BEGINING);
    if (SIZE(LinkItem)
        != Read(file, link, SIZE(LinkItem)))
        return(FALSE);

    if (link->MsgNo == MsgNo)
        return(TRUE); /* found it */
    if (link->MsgNo > MsgNo)
    {
        /* seek to the right spot... */

        Seek(file, lower
              * SIZE(LinkItem), OFFSET_BEGINING);
```

```
    /* requested message too old */
    return(FALSE);
}

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
   */

while (lower < upper)
{
    target = (upper + lower) / 2;
    Seek(file, target
          * SIZE(LinkItem), OFFSET_BEGINING);
    if (SIZE(LinkItem)
        != Read(file, link, SIZE(LinkItem)))
    {
        return(FALSE);
    }
    if (link->MsgNo == MsgNo)
    {
        /* got it! next read will read
           the following message
           */
        return(TRUE);
    }
    else if (link->MsgNo < MsgNo)
    {
        /* because of integer math we can end up
           with the target equal to the lower bound,
           this is the time to stop since further
           iterations will produce the same result
           (as lower and upper are adjacent and we
           have already inspected both the seek
           position is correct too)
           */
        if (lower == target) return(FALSE);

        /* 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);
}

```

----- Footnotes -----

(1) 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.

1.198 icebbs_demo.guide/Find the next message

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 then to find each subsequent message after that (although its not really necessary since they are in fact sequentially stored).

To find the first real message in the file, just ask to find message number 1. This works because message number 1 is permanently reserved by the FirstLinkItem; and hence, the message that follows number 1 will be the first real message in the file.

```
int FindMessageAfter(BPTR file,
```

```

        struct LinkItem *link,
        ULONG MsgNo,
        struct FirstLinkItem *fli)
{
    FindMessageNo(file, link, MsgNo, fli);
    if (SIZE(LinkItem)
        != Read(file, link, SIZE(LinkItem)))
        return(FALSE);

    /* check to see if the message that was
       returned is number 0, if it is we have
       reached the logical end of file, so
       again return false
    */
    if (link->MsgNo == 0) return(FALSE);
    return(TRUE);
}

```

1.199 icebbs_demo.guide/Find the previous message

Finding the previous message

.....

This function is used to find the message that occurs before a particular message. This would be used in a sequential reverse read.

```

int FindMessageBefore(BPTR file,
        struct LinkItem *link,
        ULONG MsgNo,
        struct FirstLinkItem *fli)
{
    FindMessageNo(file, link, MsgNo, fli);
    Seek(file, -(2*SIZE(LinkItem)), OFFSET_CURRENT);
    if (SIZE(LinkItem)
        != Read(file, link, SIZE(LinkItem)))
        return(FALSE);
    return(TRUE);
}

```

1.200 icebbs_demo.guide/Find end of file

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_BEGINNING);
    Read(dat, &data, SIZE(DataItem));
    Seek(dat, data.Length, OFFSET_CURRENT);
    return(local.Offset + data.Length
           + SIZE(DataItem));
}
```

1.201 icebbs_demo.guide/The message reader

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.

1.202 icebbs_demo.guide/Support utilities

Support Utilities

Currently there are a number of support utilities for the IceBBS SIG system. These are trim, newsin, export, import, uuexport, inmail, outmail and updatesigs. some of which are described in the USENET (see UUCP) and FidoNet (see FidoNet) chapters. At the user's end there is the IceOLR off-line reader.

Triming SIGs	Deleting old messages to free disk space
Adding SIGs	Adding extra SIGs to your system

1.203 icebbs_demo.guide/Triming SIGs

Trim
....

trim is a tool to selectively delete old files from the IceBBS message tree on your system. This is commonly referred to as an aging process.

This program first builds a list of the SIGs to be processed and then trims each sig in the list. The list is built by:

1. first loading the S:IceBBS.NEWSIGS file,
2. if the names of specific SIGs are included on the command line then it will mark only those (out of all the SIGs in the SIG list) for processing.

1.204 icebbs_demo.guide/Adding SIGs

Updatesigs
.....

This program is used by the sysop to create the necessary files for all the SIGs in the system. Typically he runs this once when he has changed the S:IceBBS.NEWSIGS (see IceBBS.NEWSIGS file) file to include and new sigs. Old sigs should just be deleted when removed from the S:IceBBS.NEWSIGS file. The command takes no parameters.

1.205 icebbs_demo.guide/Mail

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 icon (1)

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() function (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:

-uUSERNAME

Can be used to specify the user name,

-eEDITOR

Can be used to override the editor name that was set by the icon,

-pPASSWORD

Can be used to specify the password,

-?

Displays a help screen.

The mail command can also be used to create new accounts as needed (3). One use for this is to create the IceBBSNEWUSER account. Another is to setup accounts for users (under a pre-arranged password) before they call. To do this just issue a command like:

```
mail "-uNew User Name" "-pHis password"
```

note that if there are spaces in the user's name or password then the command line switch and parameter must be enclosed in quotation marks as shown above.

----- Footnotes -----

(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.

(3) This functionality should really be part of the edituser program.

1.206 icebbs_demo.guide/Electronic mail

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.

EMail directories	Where and how email is stored
Mail between users	Sending mail within the BBS
Mail to FidoNet	Sending mail between BBS systems via FidoNet
Internet mail	Sending mail via the Internet
EMail support commands	The IceBBS software for mail

1.207 icebbs_demo.guide/EMail directories

Email directories

=====

The message and file exchange between users on the single IceBBS system is fully self-contained(1), 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 file (2) and are all stored in the system email directory (see IceBBS.BINIT file). When a user logs into the BBS the server will check his mail directory and send him a message (see Error messages), if there is waiting mail.

Similarly, when a user receives a file via email the system will place that file into a private directory (3), 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... directory (4). 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 type files and is really the vestigial remnant of the email subsystem's original file transfer method, so it may be deleted in future releases.

The system email directory is also used to store the users' email profile files, these are files that the individual users create and then place online for other users to view. These files are typically used to place a description of the user, his interests and skills in a readily accessible place. They are also useful places to publish a public key for an encryption program such as PGP. These profile files are formed from the account's file number by appending a letter P. These files are plain ASCII, no special care needs to be taken with regard to their contents.

----- Footnotes -----

(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.

1.208 icebbs_demo.guide/Mail between users

Between users on the BBS

=====

When a user selects the email tool in the IceTERM program the BBS will start sending whatever mail is currently in the user's mail directory. The messages are sent one at a time to the terminal program and are displayed in a scrolling list window. Any files (of the junk mail type) that are waiting to be sent will get sent simultaneously via the regular file download server.

When the user attempts to send email or files to another user on the BBS he will be presented with a list of users to choose from. Initially this list is empty, so the user will have to request the current list (or a part of it) from the BBS before being able to pick a name.

Since the emailed files directory for each user is accessible via the regular get files tool of IceTERM it is possible for a sysop to just copy files directly into the directory. These files can be in any format that the get files tool understands, plain AmigaDOS, regular IceBBS format, or even the indirected IceBBS format. The easiest way of doing this is to use the email tool as this will take care of finding the correct directory name for you; however, with care the AmigaDOS copy command could be used to perform a mass mailing, for example if the system email directory is dhl:usermail then one might be tempted to try:

```
copy junk_mail.LZH dhl:usermail/#?F
```

to copy the file junk_mail.LZH into every user's email files directory. This, does not work because the copy command does not allow a wild card on the destination. However, a simple script file to do the same could be built and executed:

```
list >temp dhl:usermail/#?F dirs +
      lformat="copy junk_mail.LZH %p%n"
execute temp
```

Note that only those users who have received emailed files in the past will actually get this mailing.

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.

1.209 icebbs_demo.guide/Mail to FidoNet

Between BBSs via FidoNet

=====

The IceTERM program's email facility can also be used by a user to send, receive and reply(1) 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 Exporting netmail) 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 Fields 1 to 11) 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 program(2) is used to call other BBSs.

The second program this is needed is the import program (see Import FidoNet). 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.

In a system that uses the TrapDoor software an additional program called TrapPoll should be used to handle automatic delivery of mail to other BBS systems.

See FidoNet for further information.

----- Footnotes -----

(1) 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.

(2) TrapDoor or WPL.

1.210 icebbs_demo.guide/Internet mail

Internet mail
=====

The IceTERM program's email facility can also be used by a user to send, receive and reply(1) to Internet email messages. As far as the user is concerned these are much the same thing as mail between users on the same BBS, with the exceptions that he must type in the name and Internet address of the user he is sending mail to (for replies this is done for him), and he can optionally enter a subject. In order for the BBS to supply this function the inmail support program (see InMail) must be run at various times of the day and UUCP must be installed and the appropriate sendmail command must be specified in the S:IceBBS.BINIT file, see Line 16.

Inmail scans the UUMAIL: directory, which is where inbound Internet

mail ends up, and any mail it finds addressed to accounts on the BBS will be placed in 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.

An additional command: fwdemail is provided (see FwdEmail) so that email that arrives in an IceBBS mail box can be automatically passed on to another Internet mail box, anywhere on the Internet. This was added at the request of a number of users who preferred to have their mail delivered to work or to their machine at home.

----- Footnotes -----

(1) As was the case for FidoNet email a user can still send messages directly from IceTERM; however, any replies to these messages will be placed in the email SIG only.

1.211 icebbs_demo.guide/EMail support commands

Support Commands
=====

These are the additional IceBBS commands for supporting the email system.

CheckMail	To check if there is mail waiting for a user
EMail	The sysop's direct email command
Forward	To forward mail from one account to another
Send	Now obsolete

1.212 icebbs_demo.guide/CheckMail

CheckMail

This command is used to check if there is any email waiting on the system for a particular user. Syntax is:

```
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 it's icon (1). 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.

----- Footnotes -----

(1) which it searches for in the current directory first, then the S: and then the C: directories.

1.213 icebbs_demo.guide/EMail

Email

This utility to 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:

-a

Will cause the message (or file) to be sent to all users, any users who are named on the command will receive two copies (not implemented yet),

-f

The data file the command is to send is not a message and should be sent via file transfer, a copy of the file will be placed in each user's emailed files directory so they can download it at their leisure with the get files tool,

-F

The data file the command is to send is not a message and should be sent via file transfer, this sends the file as soon as the user accesses the email tool. This is useful for sending junk mail files. The user cannot resume the download of files sent in this way so its not a good idea to send something important this way,

-m

The data file the command is to send is a file containing a message so send it as an email message,

-r

Use this directory for the email (overrides S:IceBBS.BINIT). This is intended for testing (for example you might want to copy the entire email directory and then work with a copy of it so that you do not destroy any user's unread mail).

-s

Use this to set a different name for the UUCP sendmail command, if not set this will default to sendmail,

-u

Use this as the user name for reading and sending mail (it overrides the icon file's USER= field). This is useful if you want to test mail and you want to have several sysop accounts.

When the command is run without specifying one of the -f, -F or -m flags then it will display any waiting mail for the given user account. This is something to avoid doing if you are running the email tool through the remote sysop shell since this will open the editor which will be impossible to exit.

As a sysop you may actually need to receive email under several different names. The easiest way to do this is to run the email command from a script file similar to this:

```
email
email "-ustephen vermeulen-newsletter editor"
email "-uStephen Vermeulen"
email "-uthe sysop"
email -usysop
```

1.214 icebbs_demo.guide/Forward

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.

1.215 icebbs_demo.guide/Send

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.

1.216 icebbs_demo.guide/Monitor

Monitor

This program implements an activity monitor window (similar to the one the IceTERM program provides) that the sysop can run on his WorkBench. The program takes a number of default configuration values first from the icon (1) 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.

----- Footnotes -----

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

1.217 icebbs_demo.guide/Conferencing

Conferencing

IceBBS supports a simple multi-user conferencing feature.

Configuring conference	The few things that can be adjusted
Confer	The sysop's conference window
Problems	Problems you might encounter

1.218 icebbs_demo.guide/Configuring conference

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 Line 2). 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.

1.219 icebbs_demo.guide/Confer

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,

HAILSTRING=

sets the hailing string that will be printed to the conference at the begining of each of your messages,

TAILSTRING=

sets the tailing string that will be printed to the conference at

the end of each of your messages.

1.220 icebbs_demo.guide/Problems

Potential Problems

=====

There are several potential problems with the conference system. None of them are dangerous, but you should be aware of them:

- * If an ASCII mode user enters the conference but does not leave it and his modem is not set to hang up when he exits his terminal program, the result can sometimes be lots of garbage in the conference,
- * An ASCII conference user may experience some line noise and this may just happen to turn everyone's postings to greek letters. This can be fixed by someone typing a CTRL-O into one of their conference postings,
- * Regular IceTERM users may sometimes mess things up by setting up strange options (like making both the text colour and the background colour the same) in their hail and tail strings.

1.221 icebbs_demo.guide/News bulletins

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.

IceBBS.NEWS file

News files

Defining the news bulletin files

What's in a news file

1.222 icebbs_demo.guide/IceBBS.NEWS file

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.

1.223 icebbs_demo.guide/News files

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.
```

1.224 icebbs_demo.guide/DOS shell

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 copied (1) 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 DOS shell access), 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.

----- Footnotes -----

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

1.225 icebbs_demo.guide/Doors

The Doors System

Introduction

=====

IceBBS supports a unique and powerful system of expansion called the Doors System (1) (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 Line 14.

Door tool types	Tool types used in the door icons
Operation theory	How do doors work?
RemFile door	The remote sysop's file maintenance door
Preview door	The IFF picture previewing door
Edituser door	The remote sysop's user account editor
CircleMUD door	A multi-user dungeon door
ARexx doors	Writing doors in ARexx
IceFTP	Internet file transfer protocol
IceIRC	Internet relay chat

----- Footnotes -----

(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.

1.226 icebbs_demo.guide/Door tool types

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
AREXX

Door icons must have both the MENUNAME and SERVER tool types present for the door client to be displayed in the user's list for selection. The STACK will default to 8000 bytes if not present and the PRIORITY will default to 2 if not present.

STACK sets the number of bytes that the door server or client's stack will be set to when the door server process is spawned by the

IceBBS package. PRIORITY is the communications channel priority (not the AmigaDOS task priority) that the door's client and server will use to communicate. The priority is user selectable in case the user decides at doors are more or less important than another BBS task.

The MENUName is used by the user to specify the name of the door that will be displayed in the selection menu. You should set an informative default string here (eg. Multi User Dungeon, Abandon Hope All Ye Who Enter Here etc.), the user can change this to something else if he prefers.

The SERVER tooltip is used by the sysop and user to define the name of the server that is to be launched on the BBS when this client is started. This allows the user and sysop to change the actual file names (in case there is a name conflict) while still allowing the correct server to be launched. This tooltip is only required at the user's end of the connection.

The ACCESS tooltip is only needed in a server icon, the BBS checks for this tool type to determine the access control group and the minimum level required in that group before the user is allowed to use the door. Each ACCESS tooltip defines the access level required for one particular group, but as many ACCESS tooltips as are needed by the sysop may be included. Note that any groups that do not have access levels defined for them default to requiring level 255 (full sysop access) before access is granted.

The AREXX tool type is used to indicate that the server or client is an ARExx program rather than a compiled C door.

1.227 icebbs_demo.guide/Operation theory

Theory

=====

Once the user selects a door client from the list on his system his terminal program will LoadSeg() and CreateProc() launch that client. IceTERM sends the client a startup message that emulates the message that WorkBench sends any programs that it launches. This was done so that you do not have to do anything special to compile a IceBBS door program. Once the client is running it sends the BBS a message telling it what server to load and start.

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, GPChess 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.

1.228 icebbs_demo.guide/RemFile door

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 archiver does not stop and wait for a confirmation from the user (1) as this will cause a locked up process on the BBS machine. If you want to allow the co-sysops to use several archivers you just supply additional ARC= tooltypes. If you leave a particular option empty the user's door will not allow him to use that feature of the archiver.

Note that some archivers do not have a list option (like those that pack entire disks) and their extract option may not be useful either since they usually want to unpack to a floppy drive.

Here are some sample settings for the ARC= tooltypes:

```

ARC=lz      |t|l|-m x|
ARC=lharc   |t|l|-m x|
ARC=zoo     |-test|-list|x//|
ARC=arc     |t|l|x|
ARC=unzip   |-t|-l|-x -o|
ARC=dms     |test|||
ARC=zoom    |check|||

```

It appears that the server part of remfile can only run an archiver if the archiver is present in the C: directory. Under AmigaDOS 2.0 this may be different if the WorkBench was started after your path commands.

----- Footnotes -----

(1) As lharc does by default when it must create a subdirectory.

1.229 icebbs_demo.guide/Preview door

The Preview Door
=====

The preview door allows users to preview pictures before deciding to download them. Since full pictures take a long time to download the only feasible way of providing a preview capability is to reduce the size of the picture before sending it to the user. If a picture is compressed both in height and width by a factor of 4 this will result in a factor of 16 savings in transmission time.

As this image reduction process can take a lot of time (especially on an unaccelerated Amiga) and not all pictures can be compressed the same amount and still be recognizable it was decided that the images should be precompressed. A tool such as ASDG's AdPro, the shareware HamLabPlus or the freely distributable Wasp and FBM (1) can be used to prepare the minaturized pictures for previewing.

The only restrictions on the previewable pictures is that they should be smaller than 320 pixels wide and 200 pixels high and they cannot use HAM mode (they can use the 64 colour EHB mode). In order to keep transmission times low you should keep the sizes of the miniature pictures down to the 100x100 mark.

----- Footnotes -----

(1) The Fuzzy Bit Map image prcessing tool package that is available on the Fish Disks.

1.230 icebbs_demo.guide/Edituser door

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.

1.231 icebbs_demo.guide/CircleMUD door

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.

1.232 icebbs_demo.guide/ARexx doors

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 program(1).

----- Footnotes -----

(1) The simulator allows one to test doors without even running an IceBBS or IceTERM.

1.233 icebbs_demo.guide/IceFTP

Internet FTP
=====

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.

Installation for sysops

Drag the FTPserver icon into your BBS doors directory. And that's about it. You may want to adjust the access levels to the server if you only want certain groups to have access to it. This is done using the ACCESS= tool type that is standard to all IceBBS doors.

Of course, you will also need to have AmiTCP 4.0 (or higher) installed and running. And you will need some sort of connection to the Internet running (perhaps through PPP), see Internet tools for more information.

1.234 icebbs_demo.guide/IceIRC

Internet relay chat

=====

What is IRC?

IRC - Internet Relay Chat - is a real-time, text-based, conferencing system for the Internet. It allows a large number of people to chat at once about a large number of topics. It is real-time because within a few seconds (usually) of you typing a new message into a channel it is visible to all the other people who are viewing that channel.

Typically there might be five to ten thousand people connected into the IRC at any one time. There might be several thousand different topics being discussed, some might only have one or two people, some might have fifty.

IRC has been extended to provide lots of other services (like file transfer) over time. The current version of IceIRC only does the basics, but it's enough to start.

Installation for IceBBS sysops

Drag the IRCserver icon into your BBS doors directory. And that's about it. You may want to adjust the access levels to the server if you only want certain groups to have access to it. This is done by editing the ACCESS= tool types in the icon.

Of course, you will also need to have AmiTCP 4.0 (or higher) installed and running, and some sort of connection to the Internet running (perhaps through PPP). Internet tools has more information about this.

1.235 icebbs_demo.guide/Icon files

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.

1.236 icebbs_demo.guide/Account files

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 Line 4. 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.

1.237 icebbs_demo.guide/CDROM drives

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 IceBBS.DIRS file) 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 them (1). 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 BulkSplit) tool can be used to help with this,
- * The regular access command cannot be used on CDROM drives (because it cannot store its IceBBS.ACCESS files on them); however, there is a replacement command: hashaccess (see HashAccess) that stores the access control files on a writable disk some where else.

Deferred assigns

CDTV

CD32

CDROM-FS

Xetec

AsimCDFs

AmiCDROM

CDROM sources

Applications of this CLI feature

Using the CDTV as a CDROM drive

Using the CD32 as a CDROM drive

A commercial CDROM file system

A commercial CDROM file system

A commercial CDROM file system

A freely distributable CDROM file system

Where to get CDROMs

----- Footnotes -----

- (1) The AMUC CDROMs do have IceBBS format description files.

1.238 icebbs_demo.guide/Deferred assigns

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_Galorel:gifs defer
assign _Audio: AUDIO: defer
assign _Graphics: KM_GRAPH_01: defer
assign _AMINET: AMINET_0693: defer
assign _FishAndMore1: Fish_and_More:fish defer
assign _FishAndMore2: Fish_and_MoreII:fish defer
assign _AB20: AB20: defer
```

where I setup assigns for 9 CDROMs (I only have one drive), then in my S:IceBBS.DIRS files I list all nine disks, thus:

```
_Minis:
_GIFS:
_FishOnCD:
_Audio:
_Library:
_Graphics:
_AMINET:
_FishAndMore1:
_FishAndMore2:
_AB20:
```

and the BBS software takes care of only showing the one assignment for the one disk that is actually in the drive. At any time (apart from when someone is downloading from the disk that is currently in the drive) I can change CDROMs and the BBS software will pick this up and only show the assignment for the new disk.

The S:IceBBS.BINIT file contains additional configuration controls that can be used to reduce access conflicts when multiple users are accessing the same CDROM at the same time. Line 15, see Line 15, allows one to specify the number of blocks that will be read from the disk at a time when a file is being downloaded. Also there is an arbitration system that is very useful on juke box style CDROM drives, see IceBBS.ARBITRATE file for more information.

Currently there are six handlers that make CDROMs accessible to AmigaDOS. These are described in the following sections.

1.239 icebbs_demo.guide/CDTV

CDTV
=====

Commodore's CDTV product (and the A590 add on for the A500) can read CDROMs. One might think that this is not a viable solution for a BBS since it would be difficult to run a BBS on a CDTV machine (1); 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.

----- Footnotes -----

(1) There is a company that makes a product called SCSI-TV that adds a SCSI port to a CDTV, hence it might actually be possible to run an entire BBS on CDTV.

1.240 icebbs_demo.guide/CD32

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.

1.241 icebbs_demo.guide/CDROM-FS

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 used (1) this product quite extensively and has found it to work quite well apart 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.

----- Footnotes -----

(1) This was in 1992, so things may have improved.

1.242 icebbs_demo.guide/Xetec

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 (bundled with two CDs of freely distributable software sorted by subject). I have tested this on GVP Series II controllers with the 3.07, 4.0 and 4.5 FaaastROMs and two versions of the Sony drives and the Pioneer DRM-600 and DRM-602 and have found it to work flawlessly.

1.243 icebbs_demo.guide/AsimCDFS

AsimCDFS

=====

This is into its second major release, Ian Gunn has been running (and beta testing) this software for about a year now on his Great White North IceBBS. Currently he is running 7 CDROM disks off a GVP Series II controller, one disk is in a Chinon drive and the other 6 are contained in a Pioneer DRM 604 drive (this is a 6 disk changer type drive).

1.244 icebbs_demo.guide/AmiCDROM

AmiCDROM

=====

This is a freely distributable CDROM file system that is available

through most good BBS systems. It comes with source code in C.

AmiCDROM is a CDROM disk filing system for the Commodore Amiga. It supports the ISO-9660 standard, the Rock Ridge Interchange Protocol and the Macintosh HFS format.

The author can be reached at the Internet address:

ln_fmu@pki-nbg.philips.de

This package has been used on the AMUC Express BBS to drive a DRM-600 and a DRM-602(1) with great success. In fact it seems to do the same job that Xetec did while using much less memory.

The commercial packages still have better support for some of the other uses of CDROM, such as for PhotoCD and for playing CD32 and CDTV games.

----- Footnotes -----

(1) Both are 6 disk jukebox drives from Pioneer.

1.245 icebbs_demo.guide/CDROM sources

CDROM Sources
=====

If you know of any good sources for CDROMs of public domain, freely distributable or other software that can be legally placed online please let me know and I'll add them to this section.

Walnut Creek	A publisher of freely distributable CDROMs
Fred Fish	The Amiga Freely Redistributable Software Library
AMUC	The AMiga Users of Calgary

1.246 icebbs_demo.guide/Walnut Creek

Walnut Creek

Walnut Creek specializes in publishing CDROMs that contain images of some of the large Internet FTP sites. Some of these are updated quarterly and reduced price subscriptions are available.

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

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

1.247 icebbs_demo.guide/Fred Fish

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

1.248 icebbs_demo.guide/AMUC

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

1.249 icebbs_demo.guide/Removable drives

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 CDROM drives.

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

1.250 icebbs_demo.guide/Priorities

Using ChangeTaskPri

IceBBS (1) 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.

----- Footnotes -----

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

1.251 icebbs_demo.guide/ANSI codes

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.

Escape sequences	ANSI codes for text manipulations
Amiga ANSI	ANSI codes specifics for the Amiga

1.252 icebbs_demo.guide/Escape sequences

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 first four correspond to the normal WorkBench colours (3). To set the background colour to white the user would send the sequence: CSI 41m.

Note that multiple effects can be selected with one escape sequence such as: CSI 1;33;40m which selects boldface with colour 3 as foreground and colour 0 as background. And that to reset everything to the default settings you need to send a: CSI 0;31;40m sequence.

----- Footnotes -----

(1) in hex: 1B 5B

(2) in hex 9B

(3) Under AmigaDOS 1.3 these are: blue, white, black, and orange

1.253 icebbs_demo.guide/Amiga ANSI

Amiga ANSI Codes

=====

Here is a little manual on Amiga ANSI codes that I got from a FidoNet message (1), it may

prove useful.

Cursor movement	Positioning the cursor
Erasing	Erasing the screen
Colour graphics	Various colour attributes
Text attributes	Various text attributes

----- Footnotes -----

(1) My thanks go to Walt Sullivan for posting this message from:
Why a duck? FidoNet node: (1:163/109.4).

1.254 icebbs_demo.guide/Cursor movement

Cursor Positioning or Movement

To move the cursor to a specified position: ESC [#;#h where the first # is the desired line number and the second the desired column. To move the cursor up without changing columns: ESC [#a where # specifies the number of lines moved. To move the cursor to a specified horizontal and vertical position: ESC [#;#f where # means first the line number and the second # the column number. To get a cursor position report: ESC [#;#r where the first # specifies the current line and the second # specifies the current column. To move the cursor down: ESC [#b where # specifies the number of lines moved down. To move the cursor forward: ESC [#C where # specifies the number of columns moved. To move the cursor backward: ESC [#d where # specifies the number of columns moved. To save the cursor position: ESC [s and to restore it: ESC [u.

1.255 icebbs_demo.guide/Erasing

Erasing the Screen and Erasing a Line

To do a CLS (erase screen move cursor to home position): ESC [2j To erase from cursor to end of line: ESC [k

1.256 icebbs_demo.guide/Colour graphics

ANSI Colour Graphics

To set the colour/graphics attributes, enter ESC [#;#m where the first # is the desired foreground colour and the second is the desired 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
44      blue background
        blue background
45      magenta background
        white background
46      cyan background
        black Background
47      white background
        orange background
48      unknown
        unknown
49      unknown
        unknown
```

NOTE: Please note that for both the foreground and background sections, the colours repeat after each four codes; ie: blue text = 30 or 34; blue background= 40 or 44. (unless you use a WorkBench or screen with more than 4 colours) Also, as long as you know the escape code for the foreground colour, the same colour for background will be 10 numbers higher.

1.257 icebbs_demo.guide/Text attributes

Additional ANSI Attributes

To set additional attributes enter: ESC [#m where # is the number of the desired attribute. Select attributes from the list below: 0 all attributes off (defaults to original Preferences settings)

```
0
    all ANSI attributes off

    all ANSI attributes off
```

```
1      bold on

      bold on

2      unknown

      turns text black

3      Italics on

      italics on

4      underscore

      underscore

5      blink

      does not work on amiga

6      unknown

      unknown

7      reverse video

      reverse video

8      invisible

      invisible
```

1.258 icebbs_demo.guide/Future enhancements

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.

1.259 icebbs_demo.guide/Glossary

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 Import FidoNet 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.

CRC

Cyclic Redundancy Code, a form of checksum (or a finger print if you will) that is computed from the contents of a file. This is used to detect errors in transmission of the file, by recalculating the CRC once the file has been completely received and comparing it to the original CRC that was transmitted. CRCs can also be used to detect virus infections and other forms of file damage.

CSLIP

Compressed Serial Line Internet Protocol, a version of SLIP that compresses the headers of the packets that it carries between the two computers it connects.

DS

The Dual Standard modem from USR. This is a modem that supports both USR's proprietary HST protocol and the international standard V.32 and V.32bis protocols. Hence its "dual standard" name.

DTR

Data Terminal Ready, a line that indicates to the modem when the serial port it is connected to is live.

Email

Electronic Mail, messages and files that are sent electronically from one person to another, usually by use of some kind of a network or BBS machine.

FidoNet

This is a file transfer protocol which is often used to network a number of BBS machines so that they can transfer files and messages. It is also the name of the oldest and largest network that is running the FidoNet protocol.

FTP

File Transfer Protocol, after more than 20 years of use this is still one of the most popular ways of moving files between computers on the Internet. Sort of the X or Zmodem of the Internet.

Internet

A large meta-network of computers, mainly main frames and workstations, that are linked together 24 hours a day through dedicated lines. Personal computers are starting to appear on the fringes of this network, usually they are linked in by dial up lines running UUCP type software.

IRC

Internet Relay Chat, a real-time conferencing system that allows users all over the world to chat between themselves over the Internet.

ISP

Internet Service Provider, an entity who provides others with access to the Internet, usually for a fee.

LF

Line Feed, this is ASCII character 10, it is a character that a computer sends a printer to tell it to advance to the next line.

Its original meaning has become a bit garbled, and on 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, it's 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 IceBBS.MSGID file and VMSG struct 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 Import FidoNet and Viewing FidoNet packets.

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.

Sysop

System Operator, a person who runs a BBS or other computer system, generally someone who looks after the day to day maintenance of the computer hardware and software.

Tool type

The means by which the Amiga's WorkBench allows users to pass configuration related information into programs. The information is entered into the icon via the WorkBench's Information menu item and is available to the program.

TrapDoor

A FidoNet frontend mailer package for the Amiga, this is the software that handles the transmission of files between FidoNet systems.

USENET

A loosely connected network of computers running the UUCP Protocol software.

USR

US Robotics, manufacturers of high end modems for amature use, typically the modem manufacturer of choice on BBS systems since USR has a very agressive discount program for sysops.

UUCP

Unix to Unix CoPy, stricty this is the name of a program that is used to copy files from one UNIX machine to another, it is also the name of the protocol (as in "the UUCP Protocol") that is used to do this copying. This is the basis for USENET.

V.32

The international standard set by the CCITT for data transmission at 4800, 7200 and 9600 baud. This describes the means by which a modem encodes the digital data onto the analogue phone lines. V.32bis is an extension of this standard to higher speeds, typically to 14400 buad.

V.34

The international standard set by the CCITT for data transmission at baud rates between about 21000 and 28800. This describes the means by which a modem encodes the digital data onto the analogue phone lines. V.34bis(1) is an extension of this standard to higher speeds,

V.42

An international standard set by the CCITT for error detection and correction between modems (ARQ). This is commonly associated with V.42bis which adds a form of data compression to V.42 to allow faster transmission of previously uncompressed data. Note that V.42 and V.42bis can run at any transmission speed, so you will find them on 2400, 9600, 14400 and higher speed modems.

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.

----- Footnotes -----

(1) Which does not exist yet, but is expected sometime.

1.260 icebbs_demo.guide/Warranty

Matters Legal

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

Updates	Obtaining updates to the software
Copying	Distribution restrictions, what can be copied?
Support	How to contact the author
No warranty	What is warrantied
Disclaimer	What is disclaimed

1.261 icebbs_demo.guide/Updates

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.

1.262 icebbs_demo.guide/Copying

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.

1.263 icebbs_demo.guide/Support

Support =====

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

Stephen Vermeulen
45 Butler Cr. N.W.
Calgary, Alberta
CANADA, T2L 1Y3

Voice: (403) 282-7990
Email: svermeulen@ragnarok.mtroyal.ab.ca
FidoNet: stephen vermeulen at 1:134/92.0
WWW: <http://www.cuug.ab.ca:8001/~vermeuls/>

1.264 icebbs_demo.guide/No warranty

No Warranty =====

THERE IS NO WARRANTY FOR THE PROGRAMS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAMS "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE

PROGRAMS IS WITH YOU. SHOULD THE PROGRAMS
PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL
NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW
OR AGREED TO IN WRITING WILL ANY COPYRIGHT
HOLDER, OR ANY OTHER PARTY WHO MAY
REDISTRIBUTE THE PROGRAMS AS PERMITTED ABOVE,
BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL
DAMAGES ARISING OUT OF THE USE OR INABILITY
TO USE THE PROGRAMS (INCLUDING BUT NOT
LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE
PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN
ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

1.265 icebbs_demo.guide/Disclaimer

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.

1.266 icebbs_demo.guide/Thanks

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.

1.267 icebbs_demo.guide/Tool types index

Tool types and Buttons Index

.dat	SIG file format
.lnk	SIG file format
access	Door tool types
arexx	Door tool types
DataItem	DataItem struct
FindMessageAfter	Find the next message
FindMessageBefore	Find the previous message
FindMessageNo	Find message by number
FirstLinkItem	FirstLinkItem struct
LinkItem	LinkItem struct
menuname	Door tool types
MessageHeader	VMSG struct
MsgHeader	Feedback messages
na	UUExport
priority	Door tool types
SeekNextFree	Find end of file
server	Door tool types
stack	Door tool types
VMSG	VMSG struct
world	UUExport

1.268 icebbs_demo.guide/Command index

Command Index

Access	Access
account	Account
account	ASCII account
age	SIGs messages
alias	ASCII alias
amuc_coupon.rx	Coupon system

append	Append
assign	Deferred assigns
batchnews	UUExport
BBSJoin	BBSJoin
bbsnow	Front doors
BBSSplit	BBSSplit
BulkJoin	BulkJoin
BulkSplit	BulkSplit
BulkTest	BulkTest
bye	ASCII bye
cd	ASCII cd
ChangeTaskPri	Priorities
chat	ASCII chat
CheckMail	CheckMail
CircleMUD	CircleMUD door
confer	ASCII confer
confer	Confer
Coupons	Coupon system
ctrl	ASCII ctrl
defer	Deferred assigns
DelOld	DelOld
describe	ASCII describe
dir	ASCII dir
display	ASCII display
download	ASCII download
edituser	Edituser
Email	Email
email	ASCII email
email2cut	Exporting netmail
eu	EU
EUMUI	Edituser door
export	Export FidoNet
feedback	ASCII feedback
fido	Viewing FidoNet packets
fidopost	Posting to FidoNet
fidotime	Testing time of day
FileList	FileList
find	Find
findbuild	FindBuild
findscan	FindScan
findstat	FindStat
fmt	FMT
forward	Forward
fwdemail	FwdEmail
FwdEmail	Internet mail
getty	Configuring for UUCP
goto	ASCII goto
GPReader	SIGs messages
HashAccess	HashAccess
IceBBSNEWUSER	New users
IceOLR	The message reader
import	Import FidoNet
InMail	Internet mail
inmail	InMail
isnetup	IsNetUp
JamMail	FidoNet
lock	ASCII lock

login	ASCII login
Mail	Mail
makecoupon.rx	Coupon system
Makemotd	Makemotd
make_binit	Install script
monitor	Monitor
motdinfo	Motdinfo
MUIFind	MUIFind
mysort	MySort
news	ASCII news
newsin	NewsIn
newsince	NewSince
outmail	Outmail
pass	Account
pkt2email	Exporting netmail
preview	Preview door
prompt	ASCII prompt
protocol	ASCII protocol
read	SIGs messages
RemFile	RemFile door
safewait	Waiting in scripts
secho	Debugging scripts
send	Send
sendmail	FwdEmail
setaccount	Account
setuser	Account
sig	ASCII SIGs
Store	Store
TELNET	CircleMUD door
term	ASCII term
toolkit	Operation theory
top	ASCII top
TrapDoor	FidoNet
Trim	Triming SIGs
UnusedCoupons	Coupon system
Updatesigs	Adding SIGs
upgrade	ASCII upgrade
upload	ASCII upload
users	ASCII users
uucico	Configuring for UUCP
uucico	UUExport
uucp	FwdEmail
uuexport	UUExport
uusplit	FwdEmail
uuxqt	Configuring for UUCP
VAUX-Handler	DOS shell
VBBSReader	SIGs messages
who	ASCII who
WPL	FidoNet
xpr	ASCII protocol

1.269 icebbs_demo.guide/Concept index

Concept Index

00_FILE.LST, making and using	BulkSplit
00_INDEX.TXT, making and using	BulkSplit
A2232 serial card	A2232 card
A590	CDTV
Access control	Access
Access control	Access control
Access control, CDROM	HashAccess
Access control, directory for	Line 12
Access control, on CDROMs	CDROM drives
Access control, to DOS shell	DOS shell
Access levels	Access control
Access levels, number of	Number of groups
Access levels, setting	Setting levels
access, door tool type	Door tool types
Access, to DOS shell	IceBBS.SHELLACCESS file
Accounts, automatic upgrades and downgrades	Coupon system
Accounts, creating new ones	Mail
Accounts, editing remotely	Edituser door
Accounts, editing with EU	EU
Accounts, file format	Account files
Acknowledgements	Thanks
Adding SIGs	Adding SIGs
AmiCDROM	AmiCDROM
Amiga Library Services	Fred Fish
Amiga serial port	Internal serial
Amiga Users of Calgary	AMUC
AMUC	Thanks
AMUC	AMUC
anonymous	ASCII login
anonymous	Error messages
ANSI codes	ANSI codes
Appending files	Append
Applied Engineering, internal modem	AE modem
Arbitration, of CDROMs	IceBBS.ARBITRATE file
ARC	BulkTest
Arc mail	Import FidoNet
Areas, see SIGs	Message bases
arexx, door tool type	Door tool types
ARexx, doors	ARexx doors
ARP library	Required libraries
ASCII commands, hiding from users	ASCII lock
ASCII menus	ASCII menus
ASCII menus, chaining	ASCII goto
ASCII mode BBS, turning off	IceBBS.NOASCII file
ASCII mode, first menu file	Line 10
ASCII mode, hot keys	ASCII ctrl
ASCII mode, menu directory	Line 11
ASCII, defining new commands	ASCII alias
ASDG, serial card	ASDG DSB
AsimCDFS	AsimCDFS
AsimWare Innovations	AsimCDFS
Assign, defer option	Removable drives
Assign, use of defer option	Deferred assigns

Auto	Newsigs file format
Bad password	Error messages
Baldrick, a dog's body	EU
bankbroke	ASCII login
bankbroke	Error messages
batch directory	UUExport
BBS log file	Line 1
BBS name	Line 8
BBS, ASCII mode, turning off	IceBBS.NOASCII file
BBS, initialization file	IceBBS.BINIT file
BBS, text mode	ASCII menus
Bits, privilege control	Privileges
Black Adder, the	EU
Bold face	ANSI codes
Brackets, square	Conventions
Buffers, for CDROMs	IceBBS.ARBITRATE file
Building, find database	FindBuild
Bulk file operations	Bulk programs
BULKTEST.CFG	BulkTest
Bulletins, see news	News bulletins
Cables, for serial ports	Cables
Calls, too many	Error messages
Canadian Prototype Replicas	CDROM-FS
cantfindnewsitem	Error messages
Carrier detect	Cables
CD, carrier detect	Cables
CD32	CD32
CDROM access control	CDROM drives
CDROM drives, using with IceBBS	CDROM drives
CDROM, and download counters	CDROM drives
CDROM, arbitration	IceBBS.ARBITRATE file
CDROM, assign command	Deferred assigns
CDROM, file systems	AmiCDROM
CDROM, file systems	AsimCDFs
CDROM, file systems	Xetec
CDROM, file systems	CDROM-FS
CDROM, making one	Bulk programs
CDROM, security	HashAccess
CDROM, slow directories	IceBBS.DIRS file
CDROM, swapping	IceBBS.ARBITRATE file
CDROM, via CD32	CD32
CDROM, via CDTV	CDTV
CDROM-FS	CDROM-FS
CDROMs, and file format	Indirected storage
CDROMs, source for	CDROM sources
CDTV	CDTV
Chaining ASCII menus	ASCII goto
ChangeTaskPri	Priorities
Changing error messages	Error messages
Chat, multi-user	Conferencing
Checking mail	CheckMail
Client	Doors
Closed line	Error messages
Codes, escape	Escape sequences
Colours, in text	ANSI codes
Commands, defining new	ASCII alias
Comments	Conventions

ComPorts, 8 serial port card	ComPorts board
Conference, history depth	Line 2
Conference, multi-user	Conferencing
Conference, problems	Problems
Conference, sysop's window to it	Confer
Configuration file	IceBBS.BINIT file
Connect limits	Time limits
Control of user access to tools	Privileges
Conventions	Conventions
Cookies	Thanks
Copying	Warranty
Copyright	Warranty
Coupons, for account upgrades	Coupon system
Coupons, unused	Coupon system
CRC, testing files	BBSSplit
Cursor positioning, text	Cursor movement
Customizing, MOTD	Makemotd
Data files, for SIGs	SIG file format
Data set ready	Cables
Database for files, building	FindScan
Databases, building	FindBuild
Databases, list of	Find databases
Databases, multiple	Find databases
Databases, sizes	FindStat
Debugging, commands for	Store
Dedication	Thanks
Default account	New users
Default, FidoNet node number	IceBBS.FIDOCFG file
Defining new ASCII commands	ASCII alias
Defining, ASCII menus	ASCII prompt
Definitions	Glossary
Deleting file, message	Error messages
Deleting old files	DelOld
Description, none for file	Error messages
Descriptions, making lists of	FileList
Devices, null modem device	Null modem
Devices, required	Required libraries
Directories, access control	Access
Directories, for downloads	IceBBS.DIRS file
Directory in use, message	Error messages
Directory, for ASCII menus	Line 11
Directory, for doors	Line 14
Directory, for email	Line 5
Directory, for error messages	Line 3
Directory, for file descriptions	Line 13
Directory, for hashed access control	Line 12
Directory, for user accounts	Line 4
Disclaimer	Warranty
Disk reading	Line 15
Distribution	Warranty
Distribution archive	Distribution archive
DMS	BulkTest
Dog's body, a	EU
Door access not allowed, message	Error messages
Door server not installed, message	Error messages
Doors	Doors
Doors, ARexx	ARexx doors

Doors, CircleMUD	CircleMUD door
Doors, picture previewer	Preview door
Doors, programming examples	Operation theory
Doors, remote account editor	Edituser door
Doors, remote sysop file maintainance	RemFile door
Doors, server directory	Doors
Doors, server directory	Line 14
Doors, theory of operation	Operation theory
Doors, tool types for	Door tool types
DOS Shell	DOS shell
DOS shell, access levels for	DOS shell access
DOS Shell, access to	IceBBS.SHELLACCESS file
DOS shell, no access	Error messages
Downgrading accounts	Coupon system
Download ratio exceeded message	Error messages
Downloading, file reads	Line 15
Downloads not allowed, message	Error messages
Dreams	Future enhancements
Drive not online message	Error messages
drivenotonline	Error messages
DRM-600, Pioneer drive	IceBBS.ARBITRATE file
Dropped connections	High speed protocol
DSB, ASDG dual serial board	ASDG DSB
DSR, data set ready	Cables
Dual standard modem	USR Dual Standard
Echo, replacement for	Debugging scripts
Edituser, GUI version	EU
Electronic mail, see email	Electronic mail
Ellipses	Conventions
Email	Electronic mail
Email files waiting message	Error messages
Email SIG, name of	Email SIG name
Email waiting message	Error messages
Email, access to	Email SIG name
Email, between users on the BBS	Mail between users
Email, cannot find user	Error messages
Email, checking for waiting mail	CheckMail
Email, configuring for Internet	Line 16
Email, directories for	EMail directories
Email, downloads	EMail directories
Email, FidoNet and Internet	Outmail
Email, files	EMail directories
Email, forwarding	Forward
Email, forwarding to the Internet	Internet mail
Email, forwarding to users	FwdEmail
Email, importing from USENET	InMail
Email, internet not available	Error messages
Email, on FidoNet	Posting to FidoNet
Email, on FidoNet	Exporting netmail
Email, sending files	Email
Email, sending junk mail	Email
Email, sending junk mail	Mail between users
Email, specifing directory	Line 5
Email, sysop command for	Email
Email, user profile	EMail directories
Email, via FidoNet	Mail to FidoNet
Email, via the Internet	Internet mail

emailfileswaiting	Error messages
emailwaiting	Error messages
Encryption, see also PGP	EMail directories
Enhancements	Future enhancements
Erasing, text	Erasing
Error messages, files for	Error messages
Error messages, specifying directory	Line 3
Errors, MNP	High speed protocol
Escape sequences	Escape sequences
EU	EU
Exceeded ratio	Error messages
exceededbratio	Error messages
exceededdaily	Error messages
exceedednratio	Error messages
exceededquota	Error messages
Expansion, see doors	Doors
Export, of messages from SIGs	Export FidoNet
Exporting, email	Exporting netmail
External protocol libraries	ASCII protocol
Features	Introduction
Feed, finding a	Finding a feed
Feedback, from ASCII mode	ASCII feedback
Feedback, old SIG system	Feedback messages
Feedback, specifying files for	Line 6 and 7
Feedback, sysop's reader	Mail
FidoNet node number, default	IceBBS.FIDOCFG file
FidoNet packets, viewing contents	Viewing FidoNet packets
FidoNet, arc mail	Import FidoNet
FidoNet, email utility	Posting to FidoNet
FidoNet, exporting messages to	Export FidoNet
FidoNet, importing to SIGs	Import FidoNet
FidoNet, net mail	Exporting netmail
FidoNet, PKT files	Import FidoNet
FidoNet, posting to	Posting to FidoNet
FidoNet, security on	FidoNet security
File areas	IceBBS.DIRS file
File descriptions, directory for	Line 13
File descriptions, in ASCII mode	ASCII describe
File does not have description	Error messages
File download, read size	Line 15
File format	File format
File format, indirected	Indirected storage
File not deleted message	Error messages
File security	IceBBS.ACCESS files
File system	File system
File systems, see also CDROM	CDROM-FS
filenotdeleted	Error messages
Files, access control	Access
Files, account file format	Account files
Files, appending	Append
Files, BBS log	Line 1
Files, bulk operations on	Bulk programs
Files, CDROM	HashAccess
Files, configuration	IceBBS.BINIT file
Files, CRC error test	BBSSplit
Files, for feedback	Line 6 and 7
Files, joining descriptions	BBSJoin

Files, joining descriptions and files	BulkJoin
Files, making listings of	FileList
Files, names of SIGs	Newsigs file format
Files, on removable drives	Removable drives
Files, PKT type	Viewing FidoNet packets
Files, remote sysop door	RemFile door
Files, saving temporary	Store
Files, sending in email	EMail directories
Files, splitting descriptions	BBSSplit
Files, splitting into parts	BulkSplit
Files, testing	BulkTest
FILES.BBS, making and using	BulkSplit
Find files, finds wrong thing	Find limits
Finding a news feed	Finding a feed
Finding files	Searching
Font, typewriter	Conventions
Fonts, meaning of	Conventions
Format, of IceBBS files	File format
Forwarding email	Forward
Fred Fish	Fred Fish
FreeNet	Finding a feed
Front door programs	Front doors
Games, multi-user dungeon	CircleMUD door
Games, see also doors	Doors
Get files not allowed, message	Error messages
Glossary	Glossary
GNU	Conventions
Goals	Introduction
Greg Pringle	Thanks
Groups, access	Access control
Groups, number of	Number of groups
Hacker	Error messages
Handler, for DOS shell	DOS shell
Handler, VAUX	Required libraries
Handlers, see also CDROM	CDROM-FS
Hayes modem	Old modems
Help files, see news	News bulletins
Hibbert, Ian	Nullmodem.device
High speed modems	Advanced modems
Hot keys, for ASCII mode	ASCII ctrl
Hourly time limits	Time limits
HST modem	HST modems
Ian Gunn	Thanks
Ice, meaning of	Introduction
IceBBS.FIND	Find databases
IceBBS.MSGID	IceBBS.MSGID file
IceBBS.NEWS	IceBBS.NEWS file
IceBBS.SHELLACCESS	DOS shell
IceBBSNEWUSER file	New users
IceOLR	The message reader
IceTERM, icon problems	Icon files
Icon library	Required libraries
Icons, problems with	Icon files
IDs, see also sequence numbers	IceBBS.MSGID file
IFF, previewing pictures	Preview door
imposter	ASCII login
imposter	Error messages

Indirected files	Indirected storage
Information highway	Internet tools
Install script	Install script
Installing the BBS	Quick start
Internal modem, Applied Engineering	AE modem
Internal serial port	Internal serial
Internet mail not available, message	Error messages
Internet, configuring Email for	Line 16
Internet, forwarding mail to	FwdEmail
Internet, testing status of	IsNetUp
Internet, tools for	Internet tools
Italics	ANSI codes
Join, or the append command	Append
Joining files and descriptions	BBSJoin
Jukebox, CDROM drive	IceBBS.ARBITRATE file
Junk mail, see email	Mail between users
Levels, access	Access control
LHA	BulkTest
Libraries, required	Required libraries
Limitations, of file finder	Find limits
Limits, default	New users
Line shutdown	Error messages
linenotopen	ASCII login
linenotopen	Error messages
Lines, see also serial ports	Line 17
Lines, see serial ports	Serial ports
Lines, sharing with other tasks	Sharing lines
Linkage files, for SIGs	SIG file format
Links, in SIGs	Traversing links
Listing new files	Listing new files
Listings, of files	FileList
Log file	Line 1
Login message	Error messages
Login message, see also MOTD	MOTD
Login text	IceBBS.HAIL file
Logout	ASCII alias
Lost or bad serial data	Internal serial
Low priority mail, LPM	Posting to FidoNet
LPM	Posting to FidoNet
LZH	BulkTest
Mail, see also email	Electronic mail
Mail, see also email	Outmail
Makemotd	Makemotd
Making IceBBS files	BBSJoin
Man	Newsigs file format
Marrieta	Thanks
Mastering, CDROMs	Bulk programs
Max access	IceBBS.MAXACCESS file
Memory needed	Requirements
Menu files, first for ASCII	Line 10
menuname, door tool type	Door tool types
Menus, ASCII mode	ASCII prompt
Menus, in ASCII mode	ASCII menus
Message bases, see also SIGs	Message bases
Message of the day, also MOTD	MOTD
Message, sent when modems connect	IceBBS.HAIL file
Messages, IDs for	IceBBS.MSGID file

Messages, see also error messages	Error messages
Messages, trimming	TRIM.LASTTRIM file
MNP errors	High speed protocol
MNP modems	MNP modems
Modems, advanced	Advanced modems
Modems, Hayes	Old modems
Modems, high speed	Advanced modems
Modems, internal	AE modem
Modems, MNP	MNP modems
Modems, null	Null modem
Modems, old style	Old modems
Modems, radio	Internet tools
Modems, see also serial ports	Line 17
Modems, supported	Requirements
Modems, SupraFAX	SupraFAX
Modems, USB Dual Standard	USB Dual Standard
Modems, USB HST	HST modems
Modems, V.32 and V.32bis	V32 modems
Modems, V.34	V34 modems
Modems, V.42 and V.42bis	V42 modems
Moderator access, to SIGs	Newsigs file format
Monitor, history depth	Line 9
Monitor, of user activity	Monitor
More, ASCII command	ASCII display
motd	Error messages
motd	ASCII login
MOTD, adding a file to it	Append
MOTD, adding account info. to	Motdinfo
MOTD, adding current network status	IsNetUp
MOTD, adding new files listing	NewSince
MOTD, does not appear	MOTD
Mountlist, for VAUX handler	DOS shell
Movement, of text cursor	Cursor movement
MSGID	IceBBS.MSGID file
MSGID, FidoNet	Export FidoNet
MUD	CircleMUD door
Multiface, serial card	Multiface II and III
Multitasking, slow	Internal serial
Multitasking, with other programs	Priorities
MySigs file	Newsigs file format
Narrator	Required libraries
Net mail, exporting	Exporting netmail
NetMail	Mail to FidoNet
Netmail, sysop utility for	Posting to FidoNet
Network, testing status of	IsNetUp
New file listings	Listing new files
New files listing	NewSince
New user, message for	Error messages
New users	New users
News bulletins	News bulletins
News files, format of	News files
News item missing message	Error messages
NewsGroups, see also SIGs	Message bases
newuser	Error messages
newuser	ASCII login
No file description	Error messages
No profile for user	Error messages

Node number, default	IceBBS.FIDOCFG file
nodlallowed	Error messages
nodooraccess	Error messages
nodoorserver	Error messages
nofiledesc	Error messages
nointernetmail	Error messages
noserver	Error messages
noshellaccess	Error messages
Not an IFF picture, message	Error messages
notaniffpic	Error messages
nouserprofile	Error messages
Null modem	Nullmodem.device
Null modems	Null modem
nullmodem.device	Nullmodem.device
Offline reader	The message reader
Old message base, from ASCII mode	ASCII feedback
Origin, FidoNet	Export FidoNet
Out of time message	Error messages
Password, bad	Error messages
PATH, FidoNet	Export FidoNet
Pausing a line	Pause line
Performance	Performance
PGP	Mail between users
PGP, public keys	EMail directories
PGP, uses of	FidoNet security
Picture not an IFF, message	Error messages
Pictures, previewing before downloading	Preview door
Pioneer DRM604	AsimCDFS
Pioneer, DRM-600, 602,604	IceBBS.ARBITRATE file
PKT files	Import FidoNet
PKT files, viewing contents	Viewing FidoNet packets
Plus character	Conventions
Polling script files	Polling script
PPP device	IsNetUp
Premastering, CDROMs	Bulk programs
Preview door, not an IFF	Error messages
priority, door tool type	Door tool types
Privileges	Privileges
Profile, none available	Error messages
Protecting against bad packets	FidoNet security
Protocol, high speed	High speed protocol
Pruning, databases	FindStat
Public access sites	Finding a feed
Public key, see PGP	EMail directories
Quit	ASCII alias
Quota exceeded message	Error messages
Radio modems	Internet tools
Ratio exceeded	Error messages
Real time chat	Conferencing
Register settings, for modems	Register settings
Remote sysop, file maintainance	RemFile door
Removable media	Removable drives
Restart, menu item	Pause line
Restricting file areas	IceBBS.ACCESS files
Restrictions	Warranty
Root directories	IceBBS.DIRS file
Root directory, moving to	ASCII top

RSA, see PGP	E-Mail directories
Script commands, time of day	Testing time of day
Script file, debugging	Debugging scripts
Script files, debugging	Store
Script files, running at certain times	Timed script files
Scripts, nstall	Install script
Searching, for files	Searching
Searching, multiple databases	Find databases
Security	FidoNet security
Security	Access control
Security groups, numbers of	IceBBS.MAXACCESS file
Security, files	IceBBS.ACCESS files
Security, of files	Access
SEEN-BY, FidoNet	Export FidoNet
Sendmail, specifying command for	Line 16
Sequence numbers, for messages	IceBBS.MSGID file
Serial devices	Required libraries
Serial port, internal	Internal serial
Serial port, sharing with TrapDoor	Use with TrapDoor
Serial ports	Serial ports
Serial ports, A2232	A2232 card
Serial ports, ASDG card	ASDG DSB
Serial ports, cables for	Cables
Serial ports, comports 8 port card	ComPorts board
Serial ports, configuring	Line 17
Serial ports, multiface card	Multiface II and III
Serial ports, null modems	Null modem
Serial ports, sharing with other programs	Sharing lines
Serial ports, supported	Requirements
Server	Doors
Server shutdown	Error messages
server, door tool type	Door tool types
Server, problem starting	Error messages
Server, see doors	Line 14
Session limits	Time limits
Session passwords	FidoNet security
Setting up	Quick start
Sharing serial lines	Sharing lines
Shell access	IceBBS.SHELLACCESS file
Shell, no access	Error messages
shutdown	Error messages
Shutdown line	Error messages
Shutdown server	Error messages
SIGs	SIGs messages
SIGs, access control	SIG access control
SIGs, adding new ones	Adding SIGs
SIGs, data items	DataItem struct
SIGs, exporting messages from	Export FidoNet
SIGs, exporting messages to USENET	UUExport
SIGs, feedback	Feedback messages
SIGs, file format	SIG file format
SIGs, finding a message by number	Find message by number
SIGs, finding free space	Find end of file
SIGs, finding the next message	Find the next message
SIGs, finding the previous message	Find the previous message
SIGs, first link	FirstLinkItem struct
SIGs, importing from FidoNet	Import FidoNet

SIGs, importing from USENET	NewsIn
SIGs, link item	LinkItem struct
SIGs, moderator access	Newsigs file format
SIGs, names of	Newsigs file format
SIGs, reader for	The message reader
SIGs, trimming	TRIM.LASTTRIM file
SIGs, trimming	Triming SIGs
SIGs, VMSG format	VMSG struct
SIGs: traversing links	Traversing links
Slow CDROM directories	IceBBS.DIRS file
Sorting, files	MySort
Spaces	Conventions
Special interest groups, see SIGs	Message bases
Specifications	Introduction
Splitting files and descriptions	BBSSplit
Square brackets	Conventions
stack, door tool type	Door tool types
Statistics, for find databases	FindStat
Status of internet	IsNetUp
Styles, in text	ANSI codes
Support	Warranty
SupraFAX modem	SupraFAX
Swapping, of CDROMs	IceBBS.ARBITRATE file
SyQuest drives	Removable drives
Sysop, message reader	The message reader
System name	Line 8
System privileges	Privileges
Talk	Required libraries
Task priorities	Priorities
Task quota limit	Error messages
tempdirinuse	Error messages
Templates, for account types	EU
Temporary directory in use, message	Error messages
Temporary files, saving	Store
Terms, explained	Glossary
Testing files, CRC	BBSSplit
Testing, integrity of files	BulkTest
Texinfo	Conventions
Text on first connect	IceBBS.HAIL file
Text only BBS	ASCII menus
Text, ANSI codes	ANSI codes
Text, bold, italics etc.	ANSI codes
Text, erasing	Erasing
Thanks	Thanks
Time bank	Time limits
Time limits	Time limits
Time of day, testing	Testing time of day
Time of last trim	TRIM.LASTTRIM file
Timed script files	Polling script
Too many calls, message	Error messages
Tool types, for doors	Door tool types
Tool types, problems with	Icon files
Tools, control of access	Privileges
toomanycalls	ASCII login
toomanycalls	Error messages
Translator	Required libraries
TrapDoor, sharing serial port	Use with TrapDoor

Traversing links, in SIGs	Traversing links
Trim, last trim file	TRIM.LASTTRIM file
Triming, old files	DelOld
Trimming, databases	FindStat
Trimming, messages	Triming SIGs
Type, ASCII command	ASCII display
Typewriter font	Conventions
Underlined	ANSI codes
Unique message IDs	IceBBS.MSGID file
Unknown user	Error messages
unknownuser	Error messages
Unused coupons, directory for	Coupon system
Updates	Warranty
Updating SIGs list	Adding SIGs
Upgrades	Warranty
Upgrading accounts	Coupon system
Uploads, listing new	Listing new files
US Robitics, see USR	HST modems
USENET	UUCP
USENET, connecting to	Finding a feed
USENET, importing email from	InMail
User account information, displaying	Motdinfo
User accounts, specifing directory	Line 4
User activity, monitoring	Monitor
User profile files	EMail directories
User unknown to email, message	Error messages
Users, new	New users
Users, see also accounts	EU
Users, see also accounts	Account files
USR Dual Standard modem	USR Dual Standard
USR HST modem	HST modems
UUCP, batch directory	UUExport
UUCP, configuring	Configuring for UUCP
UUNET	Finding a feed
UUNews:Export, directory	UUExport
UUSPOOL:batch, directory	UUExport
V32 modems	V32 modems
V34 modems	V34 modems
V42 modems	V42 modems
VAUX handler	Required libraries
VAUX-Handler	DOS shell
Wait, replacement for	Waiting in scripts
Walnut Creek	Walnut Creek
Warranty	Warranty
Welmat, sharing serial port	Use with TrapDoor
Wireless modems	Internet tools
Wish list	Future enhancements
WorkBench, icon bug	Icon files
WPL, and UUCP	Configuring for UUCP
WPL, sharing serial port	Use with TrapDoor
Xetec	Xetec
XModem	ASCII protocol
XPR libraries	ASCII protocol
YModem	ASCII protocol
ZIP	BulkTest
ZModem	ASCII protocol
ZOM	BulkTest

Zone routing
ZOO

Exporting netmail
BulkTest