ICE BBS v1.1

Interactive Communications Exchange
Electronic Bulletin Board System

For the Commodore 64

Bayou
Telecommunications

ICE BBS Version 1.1

Interactive Communications Exchange
Electronic Bulletin Board System

Copyright 1987 by Scott and Eric Green
All Rights Reserved.

Another fine product of "Dem Boys" at:

Bayou Telecommunications
P.O. Box 92191 Lafayette, LA 70509
BBS: 318/984-3854 Home: 318/984-1820 (noon to 10pm)

NOTICE: This software is Copyright 1987 by Scott and Eric Green. Under the Copyright Act of 1978, you are allowed to make a backup copy of this software, FOR YOUR USE ONLY, which, by terms of the Act, CANNOT be sold, given away, or otherwise distributed under penalty of $10,000 fine for each occurance of distribution. In addition, we hereby grant the right to create derivative works (i.e. modify the program), HOWEVER, all derivative works are property of the copyright owners, by terms of the Copyright Act.

WARRANTY, and LIMITATION OF LIABILITY:

If your original disk fails to work upon receipt, please contact us within 30 days. We take great pride in the reliability and craftsmanship of our product, and do not wish for such nuisances to impair your enjoyment. We will replace it immediately.

This program is designed as a forum for exchanging messages and programs. No fitness for any other use is implied or intended.

We are not responsible for how you use this program. You are legally responsible for all acts you perform with your copy of this program.

We only promise to be good, not perfect. Therefore we cannot, in good conscience, guarantee this product to be totally bug-free. If you discover a bug in the program, please contact us immediately. A bug unreported is a bug unfixed. We shall attempt to discover the cause of the bug, and report any fixes back to you. Under no circumstances are we legally obligated to do more than that, nor are we responsible for any damage done by any possible bug, although, of course, we'll try to be as helpful as possible.

Note that we are not, of course, responsible for bugs which you introduced yourself in your attempts at modifying the program.

Under no circumstances are we responsible for anything other than what we just mentioned, except for where legislated by state and national statutes.

TABLE OF CONTENTS:

I. Introduction 1

II. The Sysop Manual

A. System requirements & limits 4
B. The Access System
1) Concepts: Binary Math 7
2) Setting up your user groups 10
3) Setting up your board accesses 12
4) Setting up your U/D accesses 15
5) Sample systems 17
C. Setting it up.
1) Modem configuration 20
2) System Drives and disks 22
3) Step by Step: the Config program 24
4) Copying System files 28
D. Running your board.
1) Logging in the first time, and writing files. 30
2) System News, Gfiles, and Pfiles 31
3) The Maintenance section 35
4) Sysop screen, and All them Sysop keys 38
5) Board and u/d maintenance 43
6) Pfiles included with your system 44
E. Converting from version 1.0 to 1.1 46

III. The User Reference

A. Miscellaneous 49
B. Main menu 50
C. Editor 54
D. MCE commands 59
E. Bulletin menu 64
F. EMAIL menu 68
G. U/D section:
1) Areas and libraries 70
2) Protocols 70
3) The library-style subsystem 72
4) The exchange-style subsystem 79
H. User list subsystem 82
I. Pfiles, Gfiles, and News files 83

IV. The Technical Manual

A. File formats 87
B. Variables 90
C. BASIC subroutines 91
D. The ICE ML 94

Appendix A: Glossary 98
Appendix B: Complete menu listings 103
Appendix C: INDEX 108

INTRODUCTION

Recommended steps for setting up your system:

If you already run an ICE system, you can skip most of the manual. Read section E in the Sysop Manual for information on how to convert from version 1.Ø to version 1.1, and differences between the two. You'll find that everything, including the u/d libraries if you so desire, will carry over to the new version of the software.

Else, if you are a new ICE sysop:

1) Scan through the user manual, to get an idea of how the board operates.
2) Read A. System Requirements in the sysop manual to make sure that you have enough equipment.
3) Read sections D.3 and D.4 of the sysop manual, to learn the basic maintenance commands necessary to set up the system.
4) Connect all equipment to your BBS computer. Start at section A of the sysop manual, and follow all instructions on setting up the system. **IMPORTANT:** You are VERY unlikely to successfully set up the system unless you read section A! Sorry about that, but it's necessary, both because of the complexity of the software, and the pirating problem.

Customer support:

The authors can be reached at the following locations:

Quantum Link: send EMAIL to user "BAYOU RAT"
USENET: (ihnp4,cbosgd)!killer!elg (elg@killer.UUCP for Internetters)
BBS: The Silicon Bayou, 318-984-3854
Voice: Our home number, 318-984-182Ø (only between noon and 1Øpm, please).

The authors can also be reached at (killer,ut-sally)!usl!elg, elg@usl.CSNET, or elg%usl@csnet-relay.ARPA, but it is illegal to use State equipment for personal for-profit usage. So make any mail to those addresses requests of a non-financial, technical nature.

Copyright 1987 Scott & Eric Green

About this manual:

This manual was written by Eric Green on a Commodore 128 using Steve Punter's WordPro-128, and printed using a Star NX-10 printer and Xetec Supergraphix interface. The daisywheel printer used in previous renditions has been discarded because of its extremely slow speed, and its problems with dealing with ASCII characters (due to its Brother typewriter heritage).

The general style of the manual is to use headings offset into the margins to denote each major topic. Areas that denote interaction with the BBS are offset to the right, with user-entered text underlined to differentiate it from things printed by the BBS.

In our opinion, WordPro-128 is the most versatile word processor for the C-128. However, it has one rather pesky problem when dealing with illustrations: It does not contain commands for a "floating" illustration area, thus our charts and tables are rather clumsily inserted. When we can afford to purchase a computer capable of supporting a real typesetting program such as Donald Knuth's TeX program (which he used to typeset his masterpiece The Art of Computer Programming), such as an Amiga, that technical problem shall be resolved.

About the authors:

Eric Green is a 23 year old student at the University of SW Louisiana, where he is majoring in Computer Science. He also terrorizes the USENET Unix network, and sometimes some commercial networks, besides calling a large number of BBS systems in many cities across the nation via PC Pursuit. He wrote the ML, this document, and parts of the BASIC.

Scott Green is a 20 year old sometime student at USL, when he's not partying or hacking away at the keyboard. He wrote the great majority of the BASIC, and also parts of the ML.

About the program:

The machine language for this program was originally written in Merlin assembly language. However, it was translated to CASM under the C-Power compiler in order to take advantage of the separate-compilation features of the C-Power linker.

Copyright 1987 Scott & Eric Green

A brief history:

It all started when a local sysop bought one of the first copies of C-NET in Lafayette. Many midnight sessions and tons of modifications later, the sysop said "you two seem pretty bright, why don't you write your OWN BBS program?". At first we didn't, thinking that the next version of C-NET would probably satisfy us. It didn't. So we started designing. Several languages were examined to write the BBS program in: FORTH, Promal, and assembly language. The problem with FORTH was the inability to easily do overlays, not to mention that a typical large FORTH program is unreadable, and that most people don't know FORTH so they wouldn't be able to modify the program. PROMAL was discarded because it is a compiled language, and the sysop would not be able to modify the program. The same criticism applies to assembly language (not to mention a certain reluctance to take on such a masochistic project). BASIC with ML support remains the best route on a Commodore 64 to provide a totally customizable BBS program.

While searching for a language, the basic algorithms for the windowing and a lightbar (later discarded in favor of command keys) were devised, in FORTH. The character translation tables were ripped bodily out of a replacement ML that we'd totally re-written (from scratch) for another sysop's C-NET 9.Ø (which had a very rudimentary ML, to which we added output-a-line and input-a-line capabilities to reduce the garbage collection delays).

By November of 1985, serious twiddling resulted in them being translated into machine language. All further work was deferred by finals, until the middle of December, when all code was transferred from Commodore assembler to Merlin assembler, and the entire machine language was re-structured to make room for a fastloader at $c8ØØ. In the meantime, the BASIC editor was being written. By the end of February, about 75% of the machine language was written. The other 25% took six months to finish, as is usual in large-scale projects. The BASIC was written in a flurry during the summer of 1986, and ICE first met the light of day in August of 1986, when the sub-board, maintenance, and EMAIL subsystems were completed enough to let users log in.

During the semester, all work stopped. In December, exchange mode and Xmodem were added. During spring break 1987, word wrap, most editing control codes, the mail editor, and several other things of that nature were added. Finally, during the summer of 1987, version 1.1 of ICE was finished, with MCE, improved up/download sections with credit system, and many many other enhancements that, in our humble opinion, make it the best BBS program available for the Commodore 64.

Copyright 1987 Scott & Eric Green

Section II: The Sysop Manual

II.A System Requirements

Minimum Equipment needed:

The following is a list of the minimum equipment needed to run a BBS using ICE version 1.1:

1. Commodore 64 or 128 computer (of course!)
2. Minimum of 2 1541-type disk drives in order to run a bulletin board, 3 or more to have uploading and downloading. Of course, one can use higher capacity disk drives. For example, a 1541 and a single SFD-1001 is sufficient to support a bulletin board with some up/downloads, or a single Xetec hard drive is sufficient for awesome capacity.
3. Auto-modem, either 300 or 1200 baud. Note that older 1660 modems will need modification in order to be used with the BBS

Equipment tested:

The following equipment has been tested for operation with the BBS:

Modems:

1) Commodore 1650 and all compatibles (Westridge, Total Telecommunications)
2) Commodore 1660
3) Commodore 1670 (and thus, indirectly, all Hayes compatibles)
4) Avatex 1200

Hesmodems and compatibles should run, also, however, we haven't found one to test.

Modems which specifically WON'T run:

Anchor Mark XII -- no carrier detect

Anchor Volksmodem 1200 -- sends back responses at wrong baud rate

Anchor 6470/6480 -- totally non-standard. But, trying for 6480 compatibility. Real Soon Now.

Disk Drives:

1) Commodore 1541,1571,1581,SFD-1001,8050,8250 disk drives
2) Xetec hard drives

All other known hard drives, such as the ICT Datachief, Progressive Peripherals Device-9, etc., ought to run also. The only event that would make any device not run with the program would be if it took up RAM space either in the BASIC area, under the BASIC ROM, or in the ML expansion area from $c000-$d000.

Copyright 1987 Scott & Eric Green

Fastloaders:

Mike J. Henry's fastboot v2, available from your Fast Hack'em disk, or purchasable from Basement Boy's Software. Note that support for this module is being phased out, in favor of fastloader cartridges, which free the available space in order to add more up/download protocols (specifically, one faster over PC Pursuit, such as WXMODEM).

Most fastloader cartridges such as the Epyx Fastload, Mach 5, and Spinnaker Turbo Load/Save should work, also.

The main limitation of fastloaders is that your user file must not be on the same device as the program files being loaded, because these programs all take up a buffer inside the drive,thus corrupting the BAM whenever relative file operations take place.

IEEE interfaces:

All modern IEEE interfaces, such as the Skyles IEEE-FLASH, the Batteries Included Buscard-][, etc., should work with ICE. The IEEE-FLASH, Buscard-][, C-LINK][, and Handic Superbox have all been tested for compatibility, and run with ICE.

1660 modem:

I mentioned that older 1660 modems needed a hardware modification. Well, the problem is that they didn't have a carrier detect line. Without a carrier detect, the BBS has no way of knowing that a user is on-line. The solution is to run a carrier detect line. Commodore published the instructions for doing that in their Input/Output newsletter, which was sent to all dealers and user groups. I do not currently have the instructions available to me, however, I do have a modified 1660, and it does work. If you really are intent upon using your older 1660 with the BBS, I'd appreciate it if you sent me a copy of those instructions, for inclusion in a later version of the manual.

How do you know if your 1660 is an older one, or one of the new ones? I asked Commodore. They said "look for a shiny solder joint on the bottom of where it plugs into the computer". Huh? Did you understand that? I didn't. I asked if they had any other way of telling, and they said "no". You would think that they at least maintain records of the serial numbers of which are the new ones, and which are the old. That would be the professional thing for Commodore to have done. Oops. Error. "professional" and "Commodore" in the same sentence?

Copyright 1987 Scott & Eric Green

System Limits:
Number of users: Unlimited
Number of U/D libraries, total: 900
Number of U/D areas: 30
Number of U/D libraries per area: 30
Number of programs per library: 30
(for exchange-style libraries, drive-limited).
Number of bulletin sub-boards: 30
Number of bulletins per sub-board: 30
Number of general files libraries: 999
Number of program files libraries: 999
Number of files per library: 30
Size of editor: 50 lines
Access groups: 15

Protocols available: Punter, Xmodem, CRC Xmodem, Sizing CRC Xmodem, ASCII (text downloads only).

Copyright 1987 Scott & Eric Green

II.B The Access System

Groups

The access system of ICE uses some unusual characteristics of binary (computer) math in order to allow one person to belong to more than one access groups. There are 15 access groups. Access group #0 is always the Sysop's access group. The other 14 can be distributed as you like.

In order to show you how to use the ICE access system, I'm going to have to lead you on a long chase to answer the questions "what is counting?" and "what is a number?", in order to answer the question "What access do I give my users?". I'm sorry to have to do this to you, but it's necessary, if you really do want to run ICE the way it deserves to be run. And besides, it's interesting. It's about the REAL mathematics, not all that boring stupid arithmetic that teachers bored you with in grade school. Did you ever wonder why 10+10=20? Well, now you're about to learn that 10+10=100!

II.B.1 Binary Math

Binary math

Nowadays, we humans use something called "decimal math". "decimal" means that we have ten digits, which we assign to the first ten numbers that we can count. These digits really aren't anything special. They're just arbitrary symbols that we devised out of thin air to assign to the first ten numbers that we can count. When we count, we start at 0, and count like this:

0 1 2 3 4 5 6 7 8 9 oops!

Ooops, we ran out of digits. What do we do now? I know you're saying "duh, what a dunce", but bear with me. We have only ten digits, 0 through 9. We have a number, 9. We want to count up to the next number. But we only have ten digits, and we already used them all! So, what we do, is start back over with the first digit, 0, and add one to the digit that's to the left of it. Since there's no digit to the left of the '0', we treat it like adding one to zero, and thus the digit to the left of the '0' is a '1'. And the result is "10".

In general, the following rules for counting are true, no matter whether you're in decimal (ten digits), trinary (three digits), hexadecimal (sixteen digits), or Saganary (BILLyuns and BILLyuns of digits):

Copyright 1987 Scott & Eric Green

1) Increment the current digit. If the current digit is already the highest digit in your number system, then
2) Make the current digit a zero, and then make the current digit the digit to the left of the current digit (that is, do step 1 on the next digit in the number).

So let's count in binary.

In binary, we only have two digits: 0, and 1.

So let's count:

0 1 ooops!

What now? We're out of digits!

You guessed it. We apply rule #2 above. We make the current digit a zero, and then increment the next digit. Which, since it ain't there, we call "zero" and therefore make it a one.

So here we are, counting away....

0 1 10 11 100 101 110 111 ....

And all is happy in Countingville.

And in case you haven't guessed, 0, 1, 10, 11, 100, etc. are all just arbitrary symbols, or groups thereof that follow our two rules of counting above, and they just stand for the same numbers that we would count if we were counting in decimal. So:

0	stands for the same thing as 0
1	stands for the same thing as 1
10	stands for the same thing as 2
11	stands for the same thing as 3
100	stands for the same thing as 4

and so forth, unto infinity.

And oh, by the way, you can see why 10+10=100 (2+2=4, remember??). Tell THAT one to your math teacher!

We can do the same thing in trinary, saying 0 1 2 10 11 12 20 ...

Or in any other number base, from 2 to infinity, for that matter (even in Saganary -- just LOVE the way he says BILLyuns and BILLyuns all through his Cosmos series!). Lucky you, I'm not going to put you through that torture, instead, we're going to go on and cover the last thing you need to know how to do, in order to set up the access groups:

Copyright 1987 Scott & Eric Green

Converting binary numbers into decimal

In decimal, a number might be 234. What is 234? Well, it is 200+30+4. That is, it is $2*10^2 + 3*10^1 + 4*10^0$. Note that any number $n^0=1$, so the result is 2*100 + 3*10 + 4 which, as you'll notice, is the same as 234 (just add it up with your calculator if you don't believe me!).

So where did the 10 come from? Well, 10 is because it's a decimal number. If it was a binary number, what would we do?

Well that's easy. If it was a binary number, we'd use 2, instead. Get it? There's ten digits in decimal, and only two digits in binary. So, can you figure out how to convert the following binary number?

11010

Well, I don't know if you've done it or not, but here's what I get:

$1*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 0*2^0$

Only problem is, I'd wear my calculator out calculating all that 2^x garbage! So instead, we use the following simple table:

Table of 2^x

Bit #	Decimal value
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192
14	16384

Bit #0 is the digit furthest to the right, and bit#14 is the digit furthest to the left (when we have 15 digits, that is). In other words, digits in a binary number are numbered in the opposite way from the way we read, as are digits in decimal numbers, for that matter (we get this from the Arabs, which is where we got decimal numbers from -- they read from right to left, instead of from left to right like we do).

Copyright 1987 Scott & Eric Green

So what's the use of that horribly long and boring table up there?

Well, it's to make it easier to convert from binary numbers to decimal numbers. Instead of going through all that horrible trouble of 2^x and 2^y, we can do this with the binary number 11010:

Since 0*anything=0, we don't have to worry about zero digits. So we just have to worry about the $1*2^x$ terms. Bits 4,3, and 1 are set in the binary number 11010. So we just fetch the values for bits 4,3, and 1 out of that silly little table above, and add them up, and presto, we have the decimal value of the binary number 11010. Which is 16+8+2=26. Easy.

So now you're grumbling and saying "What the hell is this dude saying all this sh*t for?". After all, nobody ever lost money by underestimating the attention span of the American people. Well, I'll tell you -- in a sec. That's all explained in the next section....

Setting up user groups

Now if you were paying attention through all that long, looong passage above about binary math and all that stuff, you'll understand what I'm about to say. If not, well, best go back and re-read it (or even read it the first time!). Now we're about to explain about how the access system has 15 access groups, and we're going to explain the rabbit trick that allows you, the sysop, to give one of your users membership in more than one access group.

In short, each access group is represented by a bit in a binary number, and you give each user an access number, which is the decimal conversion of that binary number.

In order to more easily set up your access groups, there's some charts in the back of your folder. Get them out right now. Look at the one called "Access group worksheet". The way you use it, is to write in the name of the access groups that you want, starting at the first line (bit#0), going on to however many catagories of users you think you'll need. Here's mine:

Copyright 1987 Scott & Eric Green

ACCESS GROUP WORKSHEET

BIT#	GROUP NAME	
0	Sysop (always)	1
1	Goob (lousy user)	2
2	Regular User	4
3	C-64 Down/uploads	8
4	C-128 Down/uploads	16
5	Amiga Down/uploads	32
6	Club Member	64
7	Adult	128
8	RPG	256
9		512
10		1024
11		2048
12		4096
13		8192
14		16384

You'll notice that I put the access numbers off to the right in mine. I guess I should have done that for yours, too, but, 20/20 hindsight... anyhow, note that bit 0 is always Sysop. Always. In addition, I defined a bunch of other groups, like people I wanted to be able to get into the C-64 up/download sections, people playing AD&D (a Role Playing Game, often played through the U.S. mail, but just as easily played via a sub-board), club members (since I run a BBS for a computer club), and so forth.

So when a user logs in, you add up together all those numbers off to the right, for only the groups that you want to give to the user, and presto, you now have the access number to give to that user.

For example, to give him Goob, Club, Regular, and C64 Up/downloads, you'd give him 2+64+4+8=78, which would give him access to everything that one of those groups can do. Needless to say, it can get a bit tedious to have to do this manually, I suggest that you get a calculator just for your computer... mine is a Casio FX-115, which is solar-powered (never need batteries at an inopportune time), and does binary, octal, and hexadecimal, too.

Copyright 1987 Scott & Eric Green

II.B.3 Setting up board accesses

OK, you know what your user accesses are going to be, now let's look at setting up your sub-boards. Sift through all those blank worksheets in the back of the folder, until you find one that says "BOARD WORKSHEET". Then glance at page 14, and look at my example board worksheet. Let's see what all those headings mean:

NAME

This is the name of the sub-board

BITS

This is what groups are necessary to get into the sub-board. Look at your user groups worksheet to see what group you want to require a person to have, before he can access this sub-board. Note that this is an OR-type affair --- if both bits 7 and 8 are set, it means that either group 7 OR group 8 can get into the sub-board. But remember, users are generally in multiple groups. That is, you may have a person who has GOOB, Regular User, and C-64 Downloads access. Thus the general procedure is to have only ONE bit set in the board's access, which is required to be set in the person's access. Of course, you can do anything you darn well please (it's called "flexibility").

ACCS

Look up at the table for converting binary to decimal. Add together the numbers corresponding to each bit in the BITS slot. For example, for bits 7 and 8, you add together 128 and 256, for a total of 384. If bit 1, the access is simply 2 -- there's nothing else to add to it.

DV

This is the device number of what disk drive you want the sub-board to be on. Generally, you can put about 5 sub-boards on a 1541, 10 on a SFD or one drive of an 8050 or 8250, and the sky's the limit for hard drives. For example, most people have a 1541 set as device 8, and a second drive is usually set up as device 9.

DR

This is the drive number. This is for dual drives such as the Commodore 8050, or Commodore 8250. The Xetec hard drive also uses this as its "unit" number, that is, a Xetec hard drive can be partitioned into up to 9 "units". Otherwise, for 1541, SFDs, and all other normal single disk drives, this must be a 0.

AN

Put a check mark here if you wish people to be able to post anonymously in this particular sub-board.

Copyright 1987 Scott & Eric Green

MCE

Put a check mark here if you wish people to be able to post using MCE (Macro Command Evaluator) on this sub-board. Note that many children have this urge to make the cursor do all sorts of stupid tricks with MCE, thus irritating beyond tolerance point any person who has passed puberty.

PST

Posting access. Should usually put a check mark here. This is a "freebee", resulting because the u/d section uses the same subroutine to pull in u/d directories. In the u/d section, this check mark means "can I upload?". It's not quite as useful in the bulletin section :-).

SUBSYSOP

Don't worry about this for now. When you have lots of users, you'll get tired of cleaning up the board after them, and you'll probably have several people whose responsibility is to clean up particular sub-boards. Or you may have some sort of on-line RPG which requires the game master to have sysop access only to his sub-board, without having sysop access on all of the board. For example, Allanon on my example board worksheet. The subsysop of a sub-board can do everything that a sysop can do -- but ONLY upon his particular sub-board.

Just fill in all the blanks on your worksheet. Then later, when we're doing the step-by-step configuration, all you'll have to do is pull all your names and numbers off of the worksheet, instead of having to think it all up at once.

Copyright 1987 Scott & Eric Green

BOARD WORKSHEET

##	NAME	BITS	ACCS	DV	DR	AN	MCE	PST	SUBSYSOP
1	General Board	1	2	8	0	✓		✓	
2	BBS Ads	1	2	8	0		✓	✓	
3	Politics	2	4	8	0			✓	
4	Mind Meanderings	2	4	8	0	✓	✓	✓	
5	UD reviews	2	4	8	0		✓	✓	
6	Flamefest	1	2	8	0	✓	✓	✓	
7	Club discussions	6	64	8	0			✓	
8	Adult Board	7	128	8	0	✓	✓	✓	
9	AD&D Discussion	8	256	8	0	✓	✓	✓	Allanon
10	AD&D moves	8	256	8	0	✓	✓	✓	Allanon
11	ExxTasy RPG Discussion	8,7	384	8	0	✓	✓	✓	Billybob
12	ExxTasy Moves	7,8	384	8	0	✓	✓	✓	Billybob

Copyright 1987 Scott & Eric Green

II.B.4 Setting up U/D accesses

The up/download subsystem is set up as up to 30 areas, with each area having up to 30 transfer libraries, for a maximum of 900 transfer libraries. This scheme was originally instituted upon the request of hard-drive sysops, who quickly filled up all of the original 30 transfer libraries available.

This is nice, but gives you a lot more things to keep track of.

So, let's first look at what up/download areas you want. Get the u/d area worksheet out from the back, and, if you want, look at my sample system on page 17. Note that my u/d worksheets really aren't very realistic -- first, I assume a Xetec hard drive, but then, I allocated far fewer u/d libraries than a typical hard drive BBS sysop would allocate! In defense, I wanted to allocate enough libraries to give you a good idea of how to handle it, but not so many that it took 15 pages to hold the listing of them all!

So now it's time for you to decide what u/d areas you want. These should be general, broad catagories of software, such as, perhaps, C-64 software, or, if you're on a hard drive, you can get a bit narrower, like having an area for C-64 games, an area for C-64 utilities, and so forth.

So what's all the stuff on the worksheet mean?

NAME, BITS, and ACCS are identical to what they were on the board configuration worksheet.

DV and DR are unused, except for one thing: Each u/d area can have its own individual entry file, that's printed whenever the u/d area is entered. This entry file is stored on device DV (8-15), drive DR (usually 0, for a single disk drive, but can be 0-9 for Xetec hard drives, 0-1 for Commodore dual drives).

The SUBSYSOP of a u/d area has no power or responsibility beyond being able to edit the entry file of that u/d area. He CANNOT exercise subsysop powers in the u/d libraries contained in that area, unless, of course, he is also the subsysop in each of those u/d libraries!

You've had plenty of practice by now. So, you ought to have that area worksheet filled out faster than I can say "BOO!". Now, for the fun part -- defining exactly what u/d libraries you want in each of those areas. Get the u/d worksheets out of the back of your folder. Look at my sample system on page 18, if you like, for an example.

Copyright 1987 Scott & Eric Green

What the headings mean:

NAME: Note that there are two different type of u/d libraries (see page 70 -- section III.G, U/D section). The two types are EXCHANGE-style, and LIBRARY-style. The exchange-style libraries are similar to 6485, 64 Exchange, or other such simple, crude programs that just allow you to log in, and directly access the disk drive. Library-style libraries store more information about each file -- who uploaded it, date, a short description, etc.

If the name starts with "E-", it is assumed to be an exchange-style library. Otherwise, it is assumed to be a library-style library (sounds redundant, doesn't it? But, better than the alternative of thinking up a new moniker for it!). Note that exchange-style libraries allow one to access each and every file on the disk drive, so you generally don't want system files on the same disks as exchange-style libraries.

A good example of a good application for an exchange-style library is if you have a spare 1541 hanging off your BBS. Then you can swap disks in and out at will, and the BBS will access any disk that happens to be in that drive.

BITS, ACCS, DV, DR -- same as for sub-boards.

UP -- normal people can upload to this library (note that sysops and the subsysop of that library can ALWAYS upload to it)

DN -- normal people can download from this library (again, sysops and the subsysop can always download)

ICE -- Whenever a person uploads a file, it is "ICE'd". That is, normal people cannot download it, until the sysop checks out the file, and un-ICE's it with the "I" command in the u/d subsystem. Nice if you don't want copyrighted software on your BBS, or if you want to check out the software first to make sure it doesn play Beethoven's 5th with the disk drive head.

The ICE selection is inoperative in exchange-style libraries, as there is no easy way to keep track of what files have been uploaded, and which were there all along.

SUBSYSOP -- same as for sub-boards.

We are now finished with the tedious business of figuring out exactly how we want our access system set up. So now, after a brief intermission to look through my example u/d subsystem, we go on to the REAL meat of the board -- actually configuring your BBS.

Copyright 1987 Scott & Eric Green

UD AREA WORKSHEET

##	NAME	BITS	ACCS	DV	DR	SUBSYSOP
1	C-64 Software	3	8			
2	C-128 Software	4	16			
3	Amiga Software	5	32			
4	Club Software	6	64			
5	Gfiles	1	2			

Copyright 1987 Scott & Eric Green

UD WORKSHEET

UD Worksheet #1 of 2

##	NAME	BITS	ACCS	DV	DR	UP	DN	ICE	SUBSYSOP
	Area #1: C64 Software								
1	C-64 adventure games	3	8	8	1	✓	✓	✓	
2	Misc C-64 Games	3	8	8	1	✓	✓	✓	
3	Disk utilities	3	8	8	1	✓	✓	✓	
4	Archiving Utilities	3	8	8	1	✓	✓	✓	
5	Terminal Programs	3	8	8	1	✓	✓	✓	
6	Misc Utilities	3	8	8	1	✓	✓	✓	
7	E-C64 Exchange	3	8	8	1		✓		
8	E-Drive 9	3	8	9	0		✓		
	Area #2: C128 Software								
1	Games	4	16	8	2	✓	✓	✓	
2	Terminal Programs	4	16	8	2	✓	✓	✓	
3	Misc. Utilities	4	16	8	2	✓	✓	✓	
4	E-C128 Exchange	4	16	8	2		✓		
	Area #3: Amiga	5	32						
1	Games	5	32	8	3	✓	✓	✓	
2	Terminal Programs	5	32	8	3	✓	✓	✓	
3	Misc Utilities	5	32	8	3	✓	✓	✓	

Copyright 1987 Scott & Eric Green

UD WORKSHEET

UD Worksheet #2 of 2

##	NAME	BITS	ACCS	DV	DR	UP	DN	ICE	SUBSYSOP
	Area #4: Club software								
1	C-64 Games	6	64	8	4	✓	✓		
2	C-128 Games	6	64	8	4	✓	✓		
3	C-64 utilities	6	64	8	4	✓	✓		
4	C-128 Utilities	6	64	8	4	✓	✓		
5	newsletter	6	64	8	4	✓			Lady Luck (#3)
6	E-Club Exchange	6	64	8	4	✓	✓		
7	E-Drive 10	6	64	10	0		✓		
	Area #5: GFiles	1	2						
1	BBS Lists	1	2	8	5	✓	✓	✓	
2	General Text Files	1	2	8	5	✓	✓	✓	
3	Other Realms SF Mag	1	2	8	5	✓	✓	✓	
4	Stupid People's Court	1	2	8	5	✓	✓	✓	

Copyright 1987 Scott & Eric Green

II.C System Setup

Modem configuration

Currently, we support 6 general modem families. NOTE: If you have an Anchor Automaton modem, stop here, throw it into the trash can, and continue only after buying a useable modem. The 6470, 6480, 6420, Mark XII, and Volksmodem 1200 are all useless as a BBS modem.

Hesmodem 2 --

Includes modems such as the Mitey Mo, Hesmodem 2. These haven't been tested. We ripped the numbers out of a terminal program. If you have a Mitey Mo, pray. Or better yet, let us know what happens, so that we can fix these numbers, if necessary.

1650 --

Includes the 1650, Westridge, Total Telecommunications, and many others.

1660 --

The Commodore Modem/300, noted for its brain damage. Note that only 1660s with carrier detects will work (early 1660s will need a hardware modification).

1670 --

The Commodore Modem/1200. This will also work with many Hayes compatibles, if they support registers (esp. S0) and produce signals properly (most modems use positive logic, the 64 uses negative logic), but do not have the DTR signal necessary to hang up the modem via hardware.

Hayes-type with Handic interface --

Hayes-clones, using the Handic RS-232 interface. The modem must support the DTR signal. If it does not, then set Poke/Hayes to Hayes instead of to Poke. A Hayes-compatible modem must also support registers 0, 2, and 7. Note that the old CBM RS-232 interface did not pass the DTR signal through, so you would need to do a hardware modification to that interface or else use the 1670 configuration info. Another RS-232 interface is the Omnitronix. However, I cannot recommend that you buy it. We had one that seemed somewhat shabbily constructed, and it failed.

Copyright 1987 Scott & Eric Green

Avatex 1200:

This is the Avatex 1200, not the Avatex 1200HC (which is a Hayes compatible, see above). Use the numbers for Hayes/Handic here. The primary difference is that the 1200 does not have registers or most Hayes commands, and thus we had to write a different device driver to use it (I.C.E. uses the device driver concept of having all modem-specific coding in a loadable module, so that to change modems, all you have to do is switch modules on your disk and use the configuration editor to change these configuration values. Old hat elsewhere, but pretty new to the C-64 world).

The following is a table of setup information for your modem. It is reproduced here only so that you see what parameters are used by the modem setup code, in case you need to modify a parameter. This particular table will be reproduced in the actual step-by-step description of how to configure the system. However, the long description I'm about to give about what each number means, will NOT be reproduced in the description of how to configure the system.

Modem:	Hes2	1650	1660	1670	Hayes/Handic
Onhook 56577	56	1	32	1	32
Onhook 56579	32	32	36	32	36
Offhook 56577	0	32	1	32	1
Offhook 56579	38	36	32	36	32
Poke/Hayes	Poke	Poke	Poke	Hayes	Poke*
Carrier mask	0	0	0	0	16

* If no DTR signal available, set this to Hayes

NOTE: The Hes2 numbers have not been tested by us, use at your own risk.

For the Avatex, enter the numbers for Hayes/Handic.

Meaning of those numbers:

Onhook and Offhook are what you poke your modem with to make it go onhook or offhook. 56579 is the Data Direction Register of the CIA that handles the RS232 port, and 56577 is the actual CIA register that handles the RS232 control lines. See the Commodore 64 Programmer's Reference Guide for more information about these. The 1650, 1660, and Hesmodem selections poke off the on/offhook bit (which is undocumented in the Programmer's Reference Guide). The Hayes numbers poke off the DTR signal, if the DTR signal is available. If that doesn't hang up your Hayes-compatible modem, you've probably got one of three problems:

Copyright 1987 Scott & Eric Green

a) All the signals are not being properly inverted by your RS232 interface. In this case, adjust your interface, if possible. If not, then invert these numbers. If that doesn't work, then

b) if your modem has a DIP switch in order to enable and disable DTR, then set it to enable DTR. IF that doesn't work,

c) Your modem probably doesn't support DTR. See the Poke/Hayes selection below.

Poke/Hayes:

If you can poke your modem off (as you can do with the 1650, 1660, or the Hayes/compatible with the Handic RS-232 interface), set this to Poke. If you have to print out "+++ ATH" to your modem to make it hang up, like the 1670, set this to Hayes.

Carrier mask:

This allows you to use modems with inverted carriers, such as the Hayes modem with the Handic interface. Generally, 0 means the usual "negative-true" type of carrier detect, and 16 means that it's a "positive-true" type of carrier detect (that is, your RS232 interface didn't invert it, or your modem put it out inverted in the first place!).

This should be enough information for you to configure the program to handle most modems not mentioned above. To write an actual modem driver (p.wait program), it would be best to study the sources and have much experience with modems.

II.C.2 Step by Step

In the following section, where I take you step by step through the configuration process, I'll mention several different disks

Definitions:

BBS disks: These are the disks that sit in your disk drives while you're actually running the BBS program. I call them by name, e.g. pfiles disk, user file disk, etc., and while I say "pfiles disk", I don't mean that it takes up the whole disk -- I just mean the disk that the pfiles are on. The user file, etc., may be on that disk, too.

Program disk: The disk that you bought from us, which contains the program. This disk is not a BBS disk, the only time it is placed in a BBS drive, is when you wish to boot the program. Afterwards, it is removed from the BBS drive and replaced with the appropriate BBS disk.

Pfiles disk: This holds all the p.xxx overlay files from side 2 of our distribution disk.

Copyright 1987 Scott & Eric Green

User file disk: This holds the user file, which contains a list of all users and their passwords.

Gfiles disk: Holds general text files.

System disk: This disk holds a variety of information:
1) Configuration info -- what disk drives are 1541, etc.
2) U/D and board info -- what boards and u/d libraries exist?
3) System text files -- the file that prints when you first connect to the system, the file that prints when you log out of the system, and all files inbetween, such as the menus, the help file, and news files.

And now we're ready for the biggy: Now we're going to actually set up your system. After this, you're going to actually going to be able to log in. I bet you didn't think we'd ever get to this point, after all those endless worksheets, that interminable discussion of number systems, and, especially, that insipid presentation of modem types and modem handling. And now you're thinking "goddarnit, isn't this dude ever going to finish this paragraph and get on with it?", and yes, you're right, I really ought to delete this paragraph and go straight into....

Copyright 1987 Scott & Eric Green

II.C.3 How to configure your system:

First thing to do:

MAKE A COPY OF YOUR DISK!!!!

Make a copy of BOTH sides of your disk -- the front side (system stuff and the main program), and the back side (program overlays).

Do not run the original, unless you are very brave and don't care about the possibility of accidentally erasing some important file. NEVER take the write protect tab off the original disk. NEVER put a write protect tab onto the copy of the disk.

Next, connect up every single bit of equipment you are going to use with the BBS. Note: Needed: one 1541, one or more other drives, and a modem. Be sure to change the device numbers on the other drives (instructions come with the drive, or, see your dealer). Also be sure that at least one of your drives is device 8, drive Ø (it doesn't have to be the 1541).

1) First things first: Connect all the disk drives you are going to use, insert blank disks, and format the disks (using the N command, as documented in the disk drive manual).

2) Insert the program disk (copy of front side of the disk we sent you) into one of the drives (must be 1541, 1571, or close compatible), and load "bbs config". It doesn't have to be drive 8. Of course, you'll have to take the blank formatted disk out of the drive in order to insert the program disk...

Type "run" (nice touch).

3) The program will then say, "Insert system disks into their drives". Remove the program disk, shove blank formatted disk back into the drive, and hit a key.

It will then look on the disk for configuration info, and when it doesn't find it, put you into the main menu of the configuration program. Select option 1) Configure System. This places you in the Configuration Menu.

4) Select option 1) System setup. This places you at the System setup menu.

Copyright 1987 Scott & Eric Green

5) Select option 1) Userfile device/drive. The userfile is best placed upon a 1541, because in general a 1541 is more reliable than more-obscure drives, at least insofar as relative file access goes. In a typical system, you might say device 8, drive Ø (note that unless you have a dual drive such as an 8050, MSD SD-2, etc., the drive number will always be Ø).

6) Select option 2), Prg. file device/drive. This is the device you load overlays in from. This is best placed upon a fast IEEE drive, such as the SFD-1001, 8050, etc., due to the slow speed of the 1541. However, if you must use the 1541, use a fastloader. You can either use "fastboot v2", off of the Fast Hack'em disk, or one of many fastloader cartridges. If you are using a fastloader, be sure to put the Prg. files on a different disk drive from the user file, due to a nasty habit that many fastloaders have (munching disks when relative files are used).

7) Select option 3), General file device/drive. In general, putting this on a high capacity drive such as the SFD, 8050, or perhaps a 1571 in double sided mode, is a Big Win. It is also convenient to put this on the same device/drive as one of your upload libraries, so that users can upload new general files and you can place them into gfile libraries without needing to copy them to another disk.

8) Select option 4), System files dev/drive. This is where all the menus are stored, along with many other miscellaneous files necessary to run the board.

9) select option 5), Electronic mail dv/dr. Electronic mail typically does not take up much space, but DOES take up many directory entries, so you might wish to place it on a 1541.

10) Select option 6), 1541 drive config. It will initially list all drives from 8 to 15 (even if they aren't there), as "non-1541". To tell it that a drive is a 1541, select the drive number from the menu. It will then ask you "Is it a 1541?" to which you answer "Y", if it's so. Then it will ask you "Use fastloader?". If you are using the Fastboot V2 program to load pfiles (overlays) off this drive, in order to speed up loading, type "Y". Otherwise, type "N". Note that if you are using a fastload cartridge, and not "fastboot v2", you type "N" to this question.

11) Select option 7), Prefix string. Set this to a 1-5 character string that you can live with, such as the initials of your BBS. These characters will be the prefix typed in by all users when they enter their user number in order to log in, e.g. ICE2.

Copyright 1987 Scott & Eric Green

12) Select option 8), Modem configuration. This takes you to the Modem configuration menu. The current values of the modem configuration will be listed, and you'll be allowed to change them. Plug in the values for your modem from this table:

Modem:	Hes2	1650	1660	1670	Hayes/Handic
Onhook 56577	56	1	32	1	32
Onhook 56579	32	32	36	32	36
Offhook 56577	Ø	32	1	32	1
Offhook 56579	38	36	32	36	32
Poke/Hayes	Poke	Poke	Poke	Hayes	Poke*
Carrier mask	Ø	Ø	Ø	Ø	16

* If no DTR signal available, set this to Hayes

13) You are still in Modem configuration, right? Well, select 7) and exit to System setup menu. Select 9) at system setup menu to quit to the Configuration menu. We are now ready to do the next two big steps: Setting up the sub-boards, and the upload libraries.

Select 2), Board setup. Enter the values off of your board worksheet, using the board editor.

In order to enter a value, you use the cursor keys to move to where you want to go. Then, you hit the return key. It will then give you a blinking cursor to enter the new value.

The exception is when you are editing the flags at the top of the screen. These flags are: Entry files (ignore this for now), Anonymous posting allowed, MCE allowed, Posting allowed. When you move the cursor over a flag, hitting return toggles it between Y and N.

Note that we ran out of screen width when writing the full-screen board editor. Therefore, parts of what would be shown on the screen (if we had a wider screen) are shown at the top of the screen, for the current line, only. Thus the flags at the top of the screen. Therefore, when you move the cursor to the right when you're on line 2, it flashes over line 2, then over the values at the top of the screen, and only then to line 3 -- just as if those values at the top were actually to the right of the screen.

Also note that you cannot edit board 30 if you only have 5 boards defined. You can only edit the next undefined board. The TOTL number at the top of the screen tells you how many boards you have defined.

Control-I and Control-D allow one to insert and delete lines.

When you are finished, type "Q" to quit.

Copyright 1987 Scott & Eric Green

14) Select 3), UD Area setup. Enter the values off of your UD area worksheet. The UD Area editor works just like the board editor, except that there are fewer flags to edit. If you chose not to define any UD areas, for example, if you're using only two 1541s, type "Q" immediately upon entering the UD Area Editor, and type "Y" when it asks whether you wish to save changes, and skip to step #16. This makes certain that the BBS program later knows that there are no UD areas defined.

15) Select 4), Up/download setup. Select the area whose u/d libraries you wish to set up, by using the cursor to move up and down the list of areas and then pressing return. Then use the UD Library Editor to enter the values off of your UD library worksheet for that area.

Repeat step 15 for each area that was defined.

16) Select 5), Create User File, unless you have a user file from an older version of ICE that you wish to preserve (in which case, you shouldn't have been following these instructions, in the first place!). After a suitable wait while the user file disk churns, you'll now be ready for the last and final thing to do: Editing the bootup file.

17) Select 6), exit to main menu.

Option 3) on the main menu, Convert User File, should not concern you. It is for people who've ran an ICE 1.Ø system, and are updating to ICE 1.1. Selecting this option will destroy a version 1.1 user file.

Select 4), Edit bootup file.

It will tell you to put in the program disk. Put in the BACKUP of the disk you received, NOT the original. Be SURE it is not write-protected.

If you have "fastboot v2", and you wish to use it, select option 1) Use fastboot.

For device and drive: You must boot the BBS program off of 1541 drive. However, you may want to put the files "ml" and "bbs" on a faster drive, such as a Xetec hard drive, for the speed of loading. Enter the device and drive of whatever drive that is, or just enter the device and drive of the 1541 you are booting off of (the BBS boots just fine from drives which are not device 8).

Finally, exit back to main menu, and exit the configuration program. We are now finished with the initial configuration of the program. You can take a brief rest to relieve the mental strain right now. But don't go too far, because we

Copyright 1987 Scott & Eric Green

still have a little more to do before we can boot the program:

II.C.4 Copying system files

You must copy a number of files from your double-sided distribution disk (hopefully, the copies, that I urged you to make as step #1 in the step-by-step configuration instructions). You can use your favorite commercial copier, such as Fast Hack'em, or a number of public domain copiers. I include a couple of public domain copiers on this disk that might be useful, especially if you are using disk drives other than 1541s (which most commercial copiers will NOT work with).

So now let's specify which files must be copied:

All files starting with "m." must be copied from the program disk (our distribution disk) to your system disk (the disk drive and device that you entered in step 8 of the step-by-step configuration).

All files starting with "s." must be copied from the program disk to the system disk, also.

All files starting with "p." (from the back side of the distribution disk), must be copied to your pfiles disk (the disk you specified in step #6).

If you selected the "fastboot v2" option, you must copy the program "fastboot v2" onto the front side of your program (distribution) disk.

Now for some fancy stuff:

If you are using a 300 baud modem, re-name "p.wait/300" to "p.wait" on your pfiles disk, and delete "p.wait/1200" and "p.wait/avatex" off of your pfiles disk.

If you are using a 1670 or other Hayes-compatible 1200 baud modem, re-name "p.wait/1200" to "p.wait" on your pfiles disk, and delete "p.wait/300" and "p.wait/avatex" off of your pfiles disk.

If you are using an Avatex 1200 modem, rename "p.wait/avatex" to "p.wait" on your pfiles disk, and delete "p.wait/300" and "p.wait/1200" off your pfiles disk.

If you chose not to have any UD areas in step 14, you can now delete "p.ud areas", "p.udload", and "p.udload-e" off of the pfiles disk. This saves about 60 blocks of disk space.

Copyright 1987 Scott & Eric Green

If you are especially short on space, you can also delete p.term (a terminal program), p.b-edit (a board editor), and p.weed (remove users who haven't called within a specified period of time). However, they really aren't all that huge (except for p.term), so the savings may be dubious.

Congratulations. You have now configured your system. It was a long and arduous task, but that's what flexibility gets you. Now all that's left to do is boot it up, log on, and handle some other trivial business that needs to be done before you can open the board to the general public. Which is what the next section is about:

Copyright 1987 Scott & Eric Green

II.D How to run your system:

Boot'em up:

Now we're ready for the big challenge: Starting up the system.

First, put the program disk (that is, a copy of the one we sent you, customized) into the drive you wish to boot off of. You may have noticed that in prior paragraphs, I said it had to be a 1541. Not quite. You CAN file-copy it off to another drive. But it DOES have to be device 8-15, drive #0, if it's not a 1541. Put all other BBS disks into their drives. Type load "boot",8 (or whatever drive the boot disk is in), then type run. Then you're off and running.

After it churns for awhile, it'll say "Put all system disks in all drives". After you do that and press return, it'll churn for awhile later, then ask you for the date and time. And finally, you'll end up with the regular sysop screen -- 9 lines of information and flags, and 16 lines of window, used for seeing what the user is doing. In a second, the user is going to be YOU.

Waiting for a call:

While you're sitting here waiting for a call, there's a couple of things you can do. Typing a capital-T gets you into the terminal. Typing commodore-I logs you on in immediate mode. Unfortunately, that looks and sees if your password matches user #1, and user #1 (you) hasn't logged on yet! Finally, commodore-1: This is what you want to type.

Logging in

Type commodore-1. The disk will whir, and the cursor will scroll rapidly down the window, until you see a prompt: 'Enter "GO"'. Enter "Z" at this prompt to skip the introductory message (since you haven't written the introductory message yet!). This will get you at a login ID prompt. Enter NEW here.

After answering 50 questions, you'll be dumped into the editor. Type something, anything, else the BBS will log you out without giving you an ID (under the assumption that you're an unsocial brat who's too illiterate to say anything). After a line of whatever (maybe "Now is the time for all good men..."), type /s, and then you'll be in -- be sure to write your password down somewhere!

Now that you're logged in, you have to give yourself access, just like when any other user logs in. There's four variables that you must set for each user:

Copyright 1987 Scott & Eric Green

Access -- you know what this is -- what we discussed in section II.b. For the sysop, access is easy -- set every bit (32767), so that you can access every section of the BBS.

Calls per day -- each individual user has an individually settable calls-per-day.

Download Credit Ratio -- This is the ratio of how many blocks the person can download, for each block the person uploads. The person is given 100 uploaded blocks automatically when he logs in. Nice if you have leechy users, you can give them something like 5 downloads for each upload, and when an individual user uploads garbage, lower his credit ratio. Otherwise set it to 255 for infinite downloads. 0 means he can't download at all.

Time per call -- how many minutes a person can stay on, per call.

Now use Control-1 through Control-5 and Commodore-1 through Commodore-5 to give yourself access 32767. Use F1 and F2 to give yourself lots of calls per day (hint -- when a number is zero, subtracting one gives you 255). Use F4 to give you a 255 download block credit ratio (infinite downloads). And finally, use F5 and F6 to give yourself lots of time per call. Then use commodore-t to give you plenty of time for this particular call, and you're off. For the description of all them keys, see page 38.

Writing files

You'll be dumped to main prompt. Typing MANX gets you into maintenance. Type D, and select the device and drive number of the system disk, as you defined way up there in the configuration process. Then, type W to use the editor (see page 54) to write the following files:

s.start -- file displayed before the user logs on

s.welcome -- file displayed after user logs on

s.end -- file displayed after user logs out

s.sys info -- information about your system (can you please mention our software here???).

s.new user -- displayed to a new user, before he gets to the answer-50-questions part of the new user login. Usually says something like "Welcome to the Banana Republic, here's our rules, follow them, or get off, and oh, if you post something nasty, we'll delete both it and you, but we aren't responsible for what you do, you're a big boy now".

II.D.2 System News, Gfiles, and Pfiles

After you've written all those files, now it's time to write the system news. Whenever a user logs on, he sees any new news. A typical news item might be something like "Sorry, but because of abuse, we're not letting people upload to our exchange libraries anymore". Or it might just say "Hi, this

Copyright 1987 Scott & Eric Green

is our new BBS, it's got lots of empty upload libraries, and lots of empty sub-boards, please help us fill them up." Anything that you want the users to see at least once, but which they'll probably just space out of, if they have to see it every time they log in.

So, you're in MANX, right? So type NF.

Then type FM. That stands for File Maintenance. In this case, news files. It'll tell you the directory is empty. Enter A, to add a file. It'll ask you for the access. If you want everybody to see it, just press return. Otherwise, just enter the access number whose bits correspond to the access groups you wish to see this news item. Just like you did for a bulletin sub-board, or a u/d library.

Then you're dumped into the editor, to write your news item.

So it's written, you type /s, and it's re-displayed the list of news items (in this case, the one you just wrote). You can write more if you want. But bear in mind that someone must actually READ your dearthless prose! Most likely, you just want to type Q, and get back to the news file menu.

When, in the course of human events, the news becomes no longer news and merely old stuff cluttering up your disk, the FM command is how you remove news, also.

OK. So you've written the system files. And you've written the news. There's only two more things you need to do: Set up the pfiles, and set up the gfiles.

Pfiles, of course, are program overlays. There is a number of them floating about for ICE. Alas, we didn't write them. The only pfiles which we support are p.b-edit, which allows you to edit your sub-boards (for example, to add new ones), p.user edit, which allows you to edit a user, p.file copy, which allows you to copy files, and p.weed, which allows you to zap users who haven't logged in, in a period of time.

So now we have to make a place for them to live.

To go to a pfile library, you just type PFx, where the x stands for some number between 1 and 999 (inclusive). Just like in high-school algebra, where an x always stands for some number. Or, if you're already in the pfile menu, Jx takes you to another pfile library. Typing just plain PF takes you to the pfile library #1. Type that now.

It's empty, of course. First, let's decide what pfile libraries you really do want. It's unlikely you'll have more than one public pfile library, at least, at first. Let's make

Copyright 1987 Scott & Eric Green

it library #1, since it's the easiest to get to (just type PF). Then you'll want a sysop library, that only you can get into, for all the sysop stuff like p.user edit or p.file copy. Make it some weird number like 73. Not that it matters, but it saves having to explain to users why they can't get into pfile library #2, when you add more pfile libraries.

Then type "E". This dumps you into the editor. You can either give a list of libraries here, or just list library #1 with its name (figure one out -- "games" sounds like a good one), and then say "This is the only library". When people type "J", it prints this list of libraries.

Then type "FM". It asks you for the name of the library. If you press return, the name is unchanged. But I doubt you want your library to be called EMPTY, especially since people can go to all those non-existent libraries all the way up to 999, and they're named EMPTY, too. It then asks you for the access of the library. Press return if you want anybody to be able to get into it. Else, enter the access number (this is determined the same way as for the boards -- that is, determine which user groups you want in here, add their bits together, and voila).

Then you can add the names of your pfiles. These must already be on the program disk. Since you don't have any games pfiles yet (unless you've been busy and downloaded them from the "Q"), just type S to save the new name and access, and Q to quit back to pfile menu. Then type J48 or whatever weird number. Type FM, and when it asks for access, say 1, so that only sysops can get into it (remember, bit Ø is always the sysop bit, and the value for bit Ø, in that chart way back at the beginning, was "1"). Then use the A command to add all those sysop pfiles -- "p.user edit", "p.b-edit", "p.weed", and "p.file copy". Then S to save all the changes, and Q to quit.

Now we're finished with the pfiles. Now we need to set up your gfiles.

Gfiles

Gfiles are text files, in standard Commodore ASCII ("Half-ASCII") such as is produced by Easyscript or a variety of other text editors. They should consist of lines of characters of less than 80 characters length, and they should end with a single uparrow at the beginning of a blank line. If for some reason you have an uparrow at the beginning of a line, precede it with an exclamation mark, "!".

You probably have some text files just lying around the house. For example, I have files that explain the Xmodem, Ymodem, Zmodem, and Punter protocols, which I collected off of various computer networks.

Copyright 1987 Scott & Eric Green

So, type GF. Then you can do everything that you did in the pfiles. Edit the list of libraries that's printed when a user types J, FM to edit the name and access of the library, and possibly to Add text files (make sure that the text files have been moved to the gfiles disk that you selected in section II.C.3). Type Jx for each gfile library that you want, and type FM in each of those libraries to set its name and access (being sure to Save your changes before exiting FM).

I estimate you've been working for two hours now. So wasn't that fun? (please, no rotten cabbages, I apologize!). Congratulations. Your system is now up and running, and fully configured. All that's left to tell you now, is how to KEEP it running. Which is what the rest of section D of this manual is about.

Copyright 1987 Scott & Eric Green

II.D.3 The Maintenance section

Maintenance

This section is reached by typing "MANX" at any menu (the name of a "C" compiler). A person must have bit Ø set in his access (sysop access) in order to get into this section. It contains commands for use only by sysops, which can munge directly with disks. Obviously, you don't want your ordinary users munging directly with data on your disk!

Commands:

* -- print what the **currently-logged-in device** and drive are.

@ -- **DOS command.** Allows you to issue a DOS command to the currently-logged-in disk drive. Most useful for issuing the V command, so that co-sysops can regularly validated your disks from remote. The "*", ":", and "?" are not valid here, so that if someone wants to delete every file on your disk, they have to go through a lot of trouble. However, the "=" is passed through, so that re-naming and copying can take place.

$ -- print out a **directory** of the disk. Asks you for a search string... if you do not enter a search string, it prints out the entire directory. If you do enter a search string, it only prints out entries whose first characters match the search string. Notice that characters like "*" and "?" are automatically translated into harmless Commodore-key characters by the i/o routines, and thus do not work as you expect here. It automatically adds a "*" to the end of the search string you enter in order to create the filename to open the directory with.

For example, if your currently-logged-in disk is the system disk, you might say "$", and

```
Match string:s.

Device Ø  Drive 8

Ø "I.C.E. 1.Ø Drive 8" Øa Ød
85   "s.user.config"     rel
2    "s.udloads"         seq

etc. etc.
```

To see how this works, try this in BASIC:

```
load"$Ø:s.*",8
```

It will load the directory of all files starting with "s." on drive #Ø, device #8).

Copyright 1987 Scott & Eric Green

D -- **change the currently-logged-in device and drive.** Change the device that the "R", "W", "@", and "$" looks at. Will not allow you to switch to an illegal device or drive, but **warning**, there is a bug in the SFD-1001 -- if you try to change to drive 1 on the SFD, it will place the SFD into dual-drive mode, and forever after, every time you try to access the SFD it will complain that there is not a disk in drive 1 (the SFD was created by chopping an 8250 dual drive in half, and apparently this is something that Commodore's awesome engineers forgot to take out in the move to a single drive).

F -- **read feedback.** Read the feedback that the users sent you with the "F" command at the main menu. After each piece of feedback, it give you the option to change the sender's access, delete that user, send mail to that user, quit, and other esoteric things of that nature (you're smart, you'll be able to handle it).

FMW -- **Write forced mail.** This command asks for the user's user #. If you just remember the user's name, you can use the "W" command to write a file on the EMAIL disk called "fm.USER NAME", where, for example, if the forced mail is for the sysop of The Silicon Bayou, you would write a file named "fm.ERIC G".

Force mail is mail that is sent to the user that the user cannot space out of, that is seen immediately upon each login until the Sysop deems to delete either it or the user. It may say something like "Frodo Annoyance, if you use vulgarity one more time, I'll kick your #$%"$&% off my board!". If the last lines are set to certain values, the user can be deleted (for example, you might want to tell him WHY he's being deleted!), the forced mail can be erased (if you only want him to see it one time), or you can immediately log the person off. To do this

Last two lines:
ERASE
OFF

will erase the forced mail, and log the person off.

Last lines:
ERASE
DELETE

Will erase the mail and delete the user (logging him off too!).

Copyright 1987 Scott & Eric Green

You can use any one of those three as the last line, and it will perform the appropriate action:

DELETE -- delete the user
ERASE -- erase the forced mail
OFF -- log the reader off.

Note to the slow-of-thought: You only enter the CAPITALIZED portion of the above as the last sentence of the forced mail. The "-- blah blah" just tells you what each one does.

FMD -- delete forced mail. Asks for the user #. If you just know his name, you can use the "@" command while on the EMAIL drive, and scratch the file "fm.USER NAME" where USER NAME is whatever the name of the user.

I -- **edit a user's ID.** Allows you to change his login name, the number of calls he can make in a day (CPD), number of downloads he can make per call (DPC), and amount of time he gets per call (TPC). If you want to delete this user, change his name to "^" (up-arrow). The BBS system will then erase him.

L -- **read system log.** Prints a log of users that have logged on -- their name, when they logged on, when they logged off, the logoff status (carrier lost, no time, regular logout).

R -- **read a file on disk.** Asks for a file name, looks for that file on the current work disk (type "*" to see what that is), and prints that file to the screen. Recommended only for regular "seq" files with lines of less than 80 characters (we have no idea what the ML would do if you fed it a binary file -- the ML is a stoic type, so it won't complain, but it might refuse to work at all).

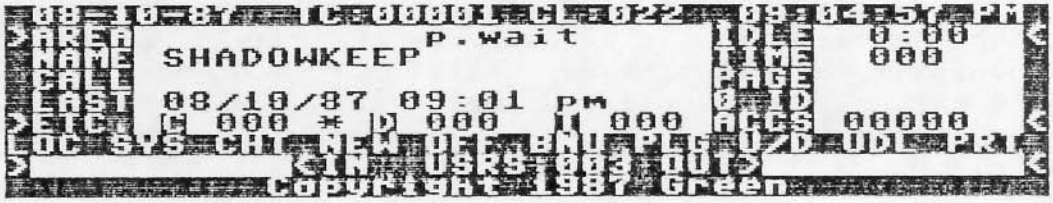
W -- **write a file.** Allows you to edit an ordinary text file on disk, as vs. binary files, system files such as s.boards (which generally will get very confused because of the "!" that the editor puts at the start of each line), or mail/bulletin files (which are compound files often over 50 lines long, which will not fit in the editor). Most useful for changing the "s.start", "s.new user", and "s.end" messages, which are the files printed out before login, to the new user, and after logout, respectively.

Copyright 1987 Scott & Eric Green

II.D.4 Interrupt-driven function keys:

The Sysop Screen

I had hoped to present a neat little numbered picture of the sysop screen, and show what each little piece of it meant. Unfortunately, the sysop screen didn't come out well at all when I attempted a screen dump... it seems like black and white just doesn't do it justice. So instead, just sit there looking at your sysop screen, and we'll go over it bit by bit, line by line, left to right, starting at the top left corner.... In order to be more precise, please log in, using commodore-I, so that it displays all of your information at the top of the screen. Then you'll be able to truly see what it looks like. And oh, if you didn't believe me, here's the screen dump of the sysop screen:



Status: Waiting for a call!

In the top left corner is the date (I'm sure you recognized it). Going rightwards are the two cryptic numbers "TC:" and "CL:". TC is the Total Calls that have been made to the system. CL is the Calls Logged since you last cleared the login log by using the L command in maintenance. The first line finishes with duh time.

The second line starts with AREA. This generally prints out the prompt of where you are in the BBS. For example, while at board 1, the AREA part of the screen shows B1>. Right next to the AREA is the module currently loaded. If you're waiting for a call, this shows "p.wait". While a new user is logging in, it'll show "p.new user". And so forth. Finally, there's IDLE. The BBS has an idle timer built in, so that if a person doesn't do anything for four to five minutes, the BBS will log them out. It currently reads only in minutes, due to our attempts to keep the interrupts as fast as possible (to prevent bogging down your computer).

The third line starts with NAME. If a user is currently logged onto the board, his name will be here. Else, it will be the name of the last user who was on the board. TIME is the number of minutes that the user has left on the system. When this reaches zero, the user is logged out. Using Commodore-T will increment this number. Holding down the Commodore, Control, and T keys (Commodore-Control-T) will

Copyright 1987 Scott & Eric Green

decrement this number. Commodore-Z will zero this number, thereby logging the user out immediately. You may notice that these keys, and all other sysop keys, are interrupt-driven, and are processed totally transparent to the user who's on the system.

The fourth line starts with CALL. This gives various information about the user, when there is a user logged in. The first number is the number of times that he's called today. The second number is the total number of times that he has called the system. Then his telephone number is displayed. Finally, if the person has used the "C" function, asking for the Sysop to chat, the word CHAT flashes in kaleidoscopic colors in the blank spot to the right of PAGE.

The fifth line starts with LAST. If the system is waiting for a call, this is the time that the last user logged out. Else it's the last time that the current user logged in. Just to the left of ID, there is a reversed digit, usually a Ø. This is the number of items of sysop feedback that the user has left in this call. Finally, there's ID, which shows the ID of the current user, if there is a user logged in. Note that if you use a prefix string of greater than three characters, only the first three characters will be displayed here, along with the user number of the current user.

Now for some fun: The ETC line, line six.

First, a brief definition. When I say "current caller" in the next few paragraphs, I mean a person who is currently logged in and using the bulletin board system.

C is the number of calls that the current caller can make in a day. The F1 key increments this number. The F2 key decrements this number. If this number is zero and you hit F2, this number becomes 255.

The little asterisk immediately to the left of D is the "carrier detect light". It lights up and becomes yellow when there is carrier detected (or in local mode). Otherwise, it is grey.

D is the download credit ratio of the current user. If this is zero, the user cannot download. If this is 255, the user has unlimited downloads. Anywhere between means that for each block of uploaded material, the user can download that many blocks. Use F3 to increment this number, and F4 to decrement it. As before, if it is zero and you hit F4, it rolls over to 255.

T is the number of minutes that the current user can stay on per call. Use F5 and F6 to increment and decrement it.

Copyright 1987 Scott & Eric Green

Note that calls per day, download credit ratio, and minutes per call are saved for each individual user. That way, if Jim Butterfield calls your board, you can give him lots of time, without having to whip up a brand new access level on the spot.

Next comes ACCS. This is the person's Access word, as described in section II.B.2 of this manual. You can increment and decrement each digit in this five-digit number individually. Control-1 through Control-5 will increment the respective digit, while Commodore-1 through Commodore-5 will decrement the respective digit. For example, if the person's access is 00000, hitting Control-1 will make it 10000. Hitting Commodore-5 will then make it 10009 (note that decrementing a zero, rolls it back to 9, and incrementing a 9, rolls it over to zero). The system will let you display numbers up to 99999 up there, but note that it won't REALLY be 99999... the biggest access that there can be, is 32767. Anything bigger overflows, with essentially random results (machine language programmers, don't paste me for not describing 16-bit integer arithmetic, negative numbers, and overflows!).

Finally, we get to the biggy:

The flag bar:

Line seven of the screen is the flag bar. This displays a variety of flags. When a flag is on, it is lit in bright yellow. Otherwise it is a drab grey. So now, for a brief description of each one:

LOC -- Local mode. When this is lit, you are connected to the system via your keyboard, as if you'd called in via the modem. Commodore-L and Commodore-I both toggle this flag on/off. Commodore-L will log you in like a normal user. Commodore-I is "instant mode". Instant mode logs you on instantly, and when you log off, does not save your user statistics, therefore making your call basically "invisible" to the system. "instant mode" is for use when you need to log in and do something real quick, such as, perhaps, delete an obnoxious user who just logged off. For normal logins, you should use Commodore-L.

SYS -- If this is lit, then when a caller types C at any menu, the BBS will ring a loud klaxon, and set the CHAT flag flashing in the PAGE area of the screen. If this is not lit, then the system will just print "the sysop is not in right now", and just set the chat flag flashing without the klaxon. Commodore-S toggles this flag.

CHT -- when lit, this puts you into CHAT mode. This allows you and the caller to type at each other. When unlit, the caller is dropped back to whatever he was doing before chat mode. F7 toggles this flag

Copyright 1987 Scott & Eric Green

NEW -- when lit, no new users can log into the BBS. The BBS ignores it when someone tries typing "NEW" at the login prompt. Commodore-N toggles this flag.

OFF -- When lit, when the current user logs off, his user information is not updated. When the user logs off, this is unlit. The purpose of this flag is for situations where perhaps you advise the user that he needs to change terminal programs, or in other such situations where the user must log off, but really isn't finished. Commodore-F toggles this flag.

BNU -- When lit, new users can go through the new user process, but are not allowed to enter the BBS until you validate them. Commodore-U toggles this flag.

PLG -- enables printer log. When lit, it prints out a log telling which users have logged in, whether they've left feedback, and other such information. Commodore-P toggles this flag. Note that you must have a printer turned on, otherwise, it will crash with a "Device not present" error.

U/D -- When this is lit, nobody can enter the file transfer section of the BBS. Turn this on during peak hours, or if you have no file transfer sections (for example, if you're running on two 1541s). Commodore-o toggles this flag.

UDL -- maintains a log of all files uploaded and downloaded, and the user who did the dastardly deed. This log is kept in the file "s.udlog" on the system disk. You can read it using the R command in maintenance. Commodore-q toggles this flag.

PRT -- When lit, prints everything printed to screen, to the printer. This is for when you want to print out the feedback, a bulletin, or anything else that's printed to the screen by some BBS command. Simply hit F8 when the text is being printed, then hit F8 again when finished.

That's the end of the flagbar. The next line contains the interrupt-driven windows into the RS232 INput and OUTput buffers, and tells how many total users there are in your user file. The final line of the sysop part of the screen either displays a copyright notice, or displays the real name (as vs. pseudonym) of the person currently logged onto the system, along with his computer type. Finally, the rest of the screen is like a 16-line terminal hooked up to the BBS program, allowing you to see what the user is typing, and also allowing you to type, too, if you so desire.

Miscellaneous other sysop keys:

The following three keys don't directly control something displayed on the screen, and are thus somewhat

Copyright 1987 Scott & Eric Green

unusual. So I put there here, after everything else has been done:

Commodore-A -- abort file transfer in progress. If a person is downloading or uploading a file using Punter protocol or any of the Xmodem protocols, this stops the file transfer, and returns him to the U/D menu.

Commodore-control-d -- dump the user. Zero his time, poke the modem off (of course, trying to poke a 1670 Modem-1200 off is an exercise in futility...).

Commodore-b -- blank the screen. Uses the VIC-][’s blanking bit, and turns the screen completely black, in order to keep the phosphors on your monitor from being burned.

And that terminates our discussion of the sysop screen, and the sysop keys. The next section in the manual is board and u/d maintenance, where you learn how to write entry files, how to change various parameters while online, and various commands for killing and editing. First, a "wall chart" of the sysop command keys:

Sysop wallchart:
Control-1 through Control-5: Increment digits of access
Commodore-1 through Commodore-5: Decrement digits of access
F1/F2: Increment/Decrement Calls Per Day
F3/F4: Increment/Decrement Download Credit Ratio
F5/F6: Increment/Decrement Time per Call
F7: Toggle CHT flag
F8: Toggle PRT flag
Commodore-a: Abort U/D file transfers
Commodore-b: Blank screen
Commodore-Control-d: dump users
Commodore-f: toggle OFF flag
Commodore-g: toggle UDL flag
Commodore-i: enter INSTANT mode (when waiting for call)
Commodore-l: toggle LOCal mode
Commodore-n: toggle NEW flag
Commodore-o: toggle U/D flag
Commodore-p: toggle PLG flag
Commodore-s: toggle SYS flag
Commodore-t: increment time-remaining
Commodore-control-t: decrement time remaining
Commodore-u: toggle BNU
Commodore-z: zero time-remaining

Copyright 1987 Scott & Eric Green

II.D.5 Board and U/D maintenance:

Bulletin commands:

The board section contains a number of commands to make it easier for the sysop or sub-sysop to maintain the sub-boards. First, you must go to the sub-board where you wish to invoke the command, then you can invoke it.

The E command allows a sysop or subsysop to Edit a bulletin, or a response to a bulletin.

The K command allows a sysop to Kill bulletins. The general syntax is:

K: Asks, for each bulletin on the sub-board, whether you want to kill it or not.

Kx: Kill bulletin #x

Kx,: Asks, for each bulletin starting at bulletin #x, whether you want to kill it or not.

Kx,y: Asks, for each bulletin between x and y, inclusive, whether you want to kill it or not.

The Z command allows the sysop or subsysop to edit board parameters, such as the name of the sub-board, whether there's an entry file, etc. Entry files are files that are printed when you enter the sub-board. They are not printed unless you enable them. This is for people running their boards off of 1541s, who probably don't want entry files due to the slow speed of the 1541. You can also edit all other board options from within Z -- whether users can post on the sub-board, who the subsysop is, can people post anonymous messages on this sub-board, and whether people can use MCE (Message Command Evaluator) on this sub-board.

The I command allows you to "ICE" postings. That is, once this command is invoked, users can no longer add responses to it. This is useful for a bulletin which has 50 responses, when you're hinting that it's time to move the topic to another bulletin, or for when a conversation has gotten particularly obnoxious (e.g. Atari vs. Commodore), and you just want to put a stop to it.

U/D commands

The U/D library maintenance commands are somewhat similiar to the bulletin sub-board maintenance commands. However, due to the differences between managing sub-boards and managing u/d libraries, there are some differences.

The "K" command operates similiarly to the "K" command in the bulletin section. However, it asks you if you wish to

Copyright 1987 Scott & Eric Green

remove the file from the disk. For example, you may want to kill all of the files out of the u/d library's directory, and only later kill the files off of disk (after you perhaps back them up?). This only works in library-style libraries, not in exchange-style libraries.

In exchange-style libraries, you can use "@" (disk command) instead, in order to scratch or rename files.

When in local mode, the "U" (upload) command allows you to add an entry to an exchange-style library.

Finally, "Z" invokes U/D library maintenance, which allows the sysop and subsysop to edit all the information about that u/d library, just as "BM" does for sub-boards.

At the U/D area prompt, you can type "Zx" to invoke the U/D area maintenance. This allows you to write an entry file for each area, which is printed when the person first enters that area, before any u/d library entry file is printed.

Pfiles included with system

Several optional program overlay files (pfiles) are included with the system. This section of the manual describes each one of them, and its operation.

p.term:

This allows you to log onto other bulletin boards using the BBS modem, without taking down your BBS. This is useful for the case where you only have one modem, or just for convenience's sake ("Darnit, Louise, where'd you put my terminal programs disk???").

To enter the terminal program, you type "T" (capital-T) when the system is waiting for a call. The terminal program's commands are fairly self explanatory. You can upload and download in either Punter, Xmodem, CRC Xmodem, or Size/Xmodem. You can auto-dial either in 300 baud format, or Hayes format. When auto-dialing, a comma or "p" in the telephone number will pause. A space will abort the auto-dialing. The terminal mode itself is simply a "glass TTY", totally dumb, save the ability to clear the screen and backspace.

p.weed:

You must place this pfile into a pfile library in order to use it. p.weed's purpose is to weed out users who haven't called recently. When you run it, it asks for a date. People who haven't called since that date are deleted. There is a mode to have it ask you if you really want to delete that person, in the event that you don't want to run a totally impartial weeding out of deadbeats.

Copyright 1987 Scott & Eric Green

p.file copy:

Self-explanatory. Allows you to copy files from one disk drive to another disk drive, without taking down the BBS. Nice for backing up files in a u/d library to one of your disks, or vice versa. Does not handle relative files -- only SEQuential and PRoGram files.

There may also be other modules, but as of release date, they have not yet been finalized.

Copyright 1987 Scott & Eric Green

II.E Converting 1.0 to 1.1

Differences

The main differences between version 1.0 of ICE and version 1.1 of ICE are:

MCE:

See section III.D for more info.

s.welcome:

The login message that tells you your name, last call date, etc., has been replaced with a file on disk that takes advantage of the new MCE commands.

USER FILE:

The user file has been changed in order to support multiple protocols, the block-credit system, and hash pointers for a hashed user file (not yet implemented). In order to convert your old user file to the new user file, you must load the configuration program, and select option 3, Convert User File.

USER LIST:

SA (Search for Access) has been added to the UL section, for sysop use.

BOARDS:

All bulletin formats are backward-compatible. However, several new capabilities have been added: The ability to enable/disable MCE posting for each sub-board, the ability to have an entry file for each sub-board, and posting access on each sub-board. These are all initially turned off (especially distracting for posting access, since it means that nobody will be able to post on anay of your boards!). All that is required is for you to go to each sub-board and type "Z" to turn on the posting ability for the sub-board, MCE if desirable, and entry file if desirable (if you turn on entry file, be sure to write one!). The ability to "ICE" a posting (so that no more responses can be added to it) has also been added. "I7", for example, will "ICE" bulletin #7.

U/D areas:

The up/download system has been totally re-vamped. It now consists of up to 30 up/download areas, each consisting of up to 30 u/d libraries. If you wish to retain your current u/d libraries, you can rename the file "s.u/ds" to "s.u/d 1" on your system disk, and then re-name the files "d.u/d x" on all u/d disks to "d.u/d 1 x". After that, you can make a u/d area #1, named "old u/d libraries". You will then have to type "Z" at each library, just as you had to type "Z" for each sub-board.

However, it's suggested that you start from scratch. I'm sure you're tired of seeing the same old programs, anyhow,

Copyright 1987 Scott & Eric Green

right?

Whatever you do, you'll have to define some u/d areas, and libraries within those areas, even if it's just to define one area called "old u/d libraries". Read section II.B.4, Setting up your U/D accesses. Then load up the configuration program, and define the areas and libraries that you want, using the new sub-board/area/library editor as is described in section II.C.3, Step by Step: The Config Program.

Of especial note is the ability to turn off downloading and uploading for each library, and the ability to turn on ICE'ing of all uploaded files (they cannot be downloaded until you, or your subsysop, checks them out and invokes the "I" command).

Two new protocols, CRC Xmodem and Sizing CRC Xmodem, have been added.

The ability for the user to abort any file transfer, even Punter, by typing two control-X characters, has been added.

NA has been added, so that you can change to a new area from within a u/d library. NA by itself takes you to the area change prompt. NAx will take you to area #x, library #1. NAx,y will take you to area #x, library #y.

Similiarly, UD from any other menu, does the same as the NA command above, with identical syntax (UDx, UDx,y).

The U/D subsystem now operates on a block download-credit-ratio basis, instead of a downloads-per-call basis. Where it says D on the screen, is now the number of blocks that the person can download, for each block that the person uploads. Me, I suggest a block-credit ratio of 5-10 blocks, but that's because I'm a notorious leech. 0 means that the person can't download, and 255 means that he has infinite downloads, in case of something like a club BBS where the club members are paying good money for the ability to download every bit they darn well please.

Immediate Mode

Immediate Mode means that you can type Commodore-I to log on instantly, and when you log off, the BBS doesn't know that you ever logged in. Your call is not logged, your user statistics are not updated. This is nice for making a quick login to delete the user who just left you a nasty message in feedback, for un-ICE'ing a file that a user just uploaded, or any other time where you'd like to log in real quick, but don't want to go read the sub-boards.

The **OFF** flag works similiarly, but for normal users. Commodore-F toggles this. Note that it is turned back off

Copyright 1987 Scott & Eric Green

when the current user logs off, so you don't have to worry about hundreds of ghost users.

Copyright 1987 Scott & Eric Green

Section III: The User Reference

III.A Miscellaneous information

Inputting a line:

I.C.E. allows you to edit the current input line (when you're sitting at a prompt) in much the same manner as Unix. When you are at a prompt, the following control characters are available for your usage:

Control-W
(hold the control key, and the W key at the same time, just as Shift-w gets you a capital W):

Deletes the last word you typed. If your line looks like

MAIN>NOW IS THE TIME

and you type control-w, the line will now look like

MAIN>NOW IS THE

Control-U
deletes the entire current line.

Control-R
Re-draw the entire current line, in case line noise or some other mechanism garbages it.

Control-I
Do a "tab" (four characters).

Aborting printouts, and flow control:

While text is scrolling by, you have a variety of options. First, you can pause in your reading of it, by hitting a control-S. The "p" key works too. When you've caught up, finished copying the text you wished to save, etc., you can then hit a control-Q or any other character to continue reading the file.

If you are tired of reading the file, you can hit the space bar in order to abort reading it. The "/" key also works (actually the primary purpose of the "/" key only occurs within the bulletin section -- where the space bar only aborts the current response, and "/" aborts the bulletin). This does not work on "forced" text, such as "forced" mail, or the new user message.

Copyright 1987 Scott & Eric Green

III.B I.C.E. Main Menu

Introduction:

The main menu is where you start out from when you login. This section of the manual lists, in alphabetical order, the commands in the main menu, description of each, and sometimes, an example of usage. It is mostly a branching-out point to other submenus, but also contains some unique commands.

B -- bulletin section!
Bx -- bulletin section #x!

The bulletin section is the most important part of the bulletin board system. Which is obvious when you stop to think about why it is called a "bulletin board system." This section contains commands for reading and writing bulletins in a variety of fashions, and is modestly the best bulletin section in existence today. The bulletin section is divided into subsections, each of which is devoted to one or more topics. See the bulletin section documentation for more information.

Examples of usage:

B -- just enter bulletin sub-section number 1.
B9 -- enter bulletin sub-section number 9.
B5 -- enter bulletin sub-section number 5.

C -- yell for chat!

This will signal the sysop that you wish to chat online with him, in the event that you have a question you need to ask him or you need to recieve access to the board (that is, if you are a new user). The first time this command is invoked, it will do a "whoop" over the sysop's speaker, and set a flag flashing in the corner of the sysop's screen. Later attempts to use this command are meaningless, and the system will tell you "You have already called him!". Thus it is useless to repeatedly hit the "C" key over and over again, the sysop will eventually get around to talking to you if he is home (assuming that you are not a perpetual nuisance that he does not want to talk to).

E -- Edit your terminal parameters.!

This allows you to edit such things as what kind of computer you are using, line length, and upper/lowercase.

Copyright 1987 Scott & Eric Green

F -- leave feedback

This leaves a message in the sysop's private mailbox. Use this when you have a question to ask about the system, need your access level changed, or otherwise have something of interest to ALL the sysops, not just one. If you wish to send a personal message to a particular sysop, send him personal mail via the M section (since all the sysops can view feedback, and some sysops may not like reading personal messages directed to another sysop).

GF -- general files

These are text files of interest, containing educational information, BBS lists, and other files of that nature. Read the Files section of the manual for more information.

H -- help section.

I hope you know the purpose of the help section. If not, I hear that in Haiti they have a great brain-transplant operation...

I -- system info

This gives information about the software and hardware that the BBS is running on. Put how many disk drives you're using, I.C.E. ordering information, or other similiar information here. The file is stored on the "system" disk as "s.sys info". Use the "E" command in the maintenance section, or Easyscript (lines must be of under 80 characters, though). Just type I and it will print out the file for you

LD -- Last Date command

What in the world does this command do? Well, when you go to board level, the RA and RN commands print out bulletins posted since the last day that you logged in. But, what if you had 55 pieces of mail the last time you logged in and didn't get to read the bulletins? Well, you just use the LD command to set your Last Date to the last time you read the bulletins, and all is happy in BBSland. Heaven for sysops, who often must log in at odd times to do quick maintenance work.

M -- electronic mail section

This section contains commands which allow you to read your electronic mail, and send electronic mail to others. See the mail section of the manual for more information.

NF -- news files

These are text files containing the system news (those files typed out to you right after you logged on). Read the Files section of the manual for more information (GF, PF, and NF are all handled by the same subroutine, which is passed a variable to detirmine which type of file it happens to be handling).

Copyright 1987 Scott & Eric Green

OFF -- logoff
Leave the system.

PF -- program files
I.C.E., like some other BBS programs, allows "program files", files which are loaded off of disk into the overlay area for users to run. You may find some simple games or other entertainments in this area, but the primary purpose of this area is for system maintenance overlays such as a directory edit routine and a "weed" routine (to delete users who have not logged in for over N amount of time). See the "files" section of the manual for information about "pfiles" (as they are usually called).

PW -- change your password
Fairly self-explanatory. You type PW, it prompts you for your current password, then prompts you to enter your new password (which you must do twice, to make sure that you don't mis-type it the first time and end up not being able to log onto the system).

S -- tell you your status
Gives you information about yourself -- your access, how many times you have called the system, the number ofbulletins you have posted, etc.

T -- date/time information
[T] gives you this information: the current date, and the date of the last day you logged on.

U -- re-read the new user message.
Self-explanatory. Type "U", see new user file again. The new user file is stored as "s.new user" on the "system" disk.

UD -- up/downloads
UDx -- enter up/download area #x
UDx,y -- enter up/download library #y in u/d area #x
If the bulletin section is the heart of the BBS, this is the soul. This section allows you to upload (send) and download (recieve) public domain programs from other likeminded people. Some obtain so much software through this means that they never have to deal with arrogant publishers and their brain-damaged copy protection schemes again. See the up/download section of the manual for more information.

Example of usage:

UD -- enter U/D area prompt
UD3 -- enter up/download area #3, library #1
UD2,4 -- enter u/d area #2, library #4

Copyright 1987 Scott & Eric Green

UL -- user list section

The user list section contains commands to find and list users in a variety of fashions -- straight list, search by name, search by computer type, and others. See the user list section of the manual for more information.

Miscellaneous things to ponder!

The following commands are available at ANY menu, not only at the main menu:

B -- bulletin boards
M -- EMAIL
GF -- general files
NF -- news files
PF -- program files
OFF -- leave the system
T -- date/time/last
C -- yell for chat
F -- leave feedback
UL -- user list
H -- help section

Copyright 1987 Scott & Eric Green

III.C Editor:

General editor usage:

Whenever you need to edit text, you are dumped into the editor subsystem. To use the editor, you simply type text, and that text goes into the editor's "buffer". Editor commands are necessary in order to save the text to disk, edit lines, etc. Editor commands start with a / (slash), and must be at the beginning of a line. For example,

/s

would save the editor buffer to disk and quit from the editor.

I repeat once again: in order for a command to be understood by the editor, it MUST start with a "/" at the BEGINNING OF THE LINE, and the editor command MUST be the ONLY thing on that line.

The Buffer

The buffer consists of the lines of text you have typed. These lines are numbered sequentially, from the first line through the last line you've typed. For example, if you have typed 5 lines, there are 5 lines in the buffer, numbered 1,2,3,4,5. If the last line you typed was "Now is the time for all good men to eat frozen pizza", that is line number 5.

General command syntax:

Some commands, such as the /s command above, don't need you to tell them what lines you are operating on (it just saves all the lines in the editor). However, many other commands can be restricted to only work on one line, a range of lines, or the whole buffer.

/d1,3 -- deletes lines 1,2, and 3. In general, anything of the nature "x,y" operates only on the lines x through y (such as the lines 1 through 3 in /d1,3).

/d4, -- deletes every line after line 4. In general, when you omit the "y" in the "x,y", and just have the form "x,", it replaces the "y" with the line number of the last line in the buffer. Thus, if you had 5 lines in the buffer, /d4, would delete lines 4 and 5.

/c4 -- centers only line number 4. When you have a command of the form "/cx", it operates only on line #x.

Copyright 1987 Scott & Eric Green

Now for a real bone of contention: What does something like "/c" or "/d" do, just typed by itself? Answer: It follows no general rule. Some commands, like /d, default to the last line of text. Other commands, such as /c, default to ALL text in the buffer. You'll have to read the documentation of each command to see what to do.

Editor Commands

Following, is a list of editor commands, in alphabetical order.

/a -- Quit the editor without saving text

/c -- center text

Centers line(s) of text, that is, puts spaces in front of the line(s) until it/they appears to be in the center of the screen. Typing /c alone on a line by itself centers every line of text in the editor buffer, which is not recommended as a way to gain your sysop's favor (reading 30 lines of centered text is, let us say, tedious).

Examples:

/c1 -- center line 1 (very common, to center titles etc.).
/c1,2 -- center lines 1 and 2
/c30, -- center lines 30 through the last line (center your name?).
/c -- center all lines in buffer (nice if you just typed in line 1, just don't do it with 40 lines of text in the buffer!).

/d -- delete text

Delete text. All text on lines above the line(s) deleted moves down. A /d on a line by itself deletes the last line of text typed. General syntax is covered in section SYNTAX, above.

Examples of usage:

/d1,4 -- deletes lines 1,2,3, and 4.
/d3, -- deletes line 3 and all lines after it.
/d3 -- deletes only line 3.

Copyright 1987 Scott & Eric Green

If the editor contains

```
NOW IS THE
TIME FOR
ALL GOOD
```

and you type /d1, the editor now contains

```
TIME FOR
ALL GOOD
```

/E -- edit a single line

This allows you to re-type a line that you earlier mis-typed. A /e alone on a line will edit the last line in the buffer. The /e command prints out the old line, then allows you to type in a new line to replace the old line.

Typing a /x at the beginning of the line will exit edit mode, in the event that you decide that the old line was OK, after all.

These line entry control codes apply to /e:

Control-V (holding control down, and pressing V at the same time), will take the character corresponding to your cursor position out of the old line, and put it into your new line.

Control-P will replace whatever's on the line, with the contents of the old line.

Control-A and Control-D subtracts one from, and adds one to, control-V's pointer into the old line, effectively Adding a space for you to type a new character, and Deleting a character out of the old line.

Control-R redraws the line, Control-W deletes the last word typed, and Control-U deletes the whole line, while Control-I is a tab. These are identical to the normal line-entry control codes.

Examples:

/e3 -- edit line 3
/e -- edit the last line in the buffer
/e4, -- edit, starting at line 4, each line from line 4 through the last line in the buffer.
/e1,3 -- edit line 1, then edit line 2, then edit line 3.

Copyright 1987 Scott & Eric Green

/I -- Start inserting lines.!

The ONLY valid syntax is "/ixx", where the "xx" is a number. For example, /i23. If you have 30 lines in the buffer, and /i23 was the last command, and you then type the sentence "Now is the time", the sentence "Now is the time" will be the new line 23, and the old line 23 and all text above it will be moved up. The current line number will then be incremented to line 24, and when you type another line, that will become the new line 24, and the old text will be moved up again, and so forth. When you are finished inserting the necessary lines, type /x at the start of the line in order to exit insert mode and once again begin appending lines to the end of the editor buffer.

Example:

The editor contains

```
NOW
IS THE
TIME FOR
ALL GOOD
MEN
```

and you type

```
/i3
```

And then type

```
VERY BEST
```

The editor then contains

```
NOW
IS THE
VERY BEST
TIME FOR
ALL GOOD
MEN
```

/l -- list text

When you list text, it puts the line number, then the actual line, for each line of text selected (if just /l, then the whole text buffer). Valid syntaxes: /l4 lists only line 4, /l1,4 lists lines 1 through 4, /l4, lists line 4 through the last line in the buffer. See SYNTAX, above.

Copyright 1987 Scott & Eric Green

/N -- clear text.
Erases all text in the buffer. Prompts you with "Are you sure?" so that you won't erase text accidentally.

/O -- toggle line numbers
Typing /o alone at the start of a line will turn line numbers on or off. When line numbers are on, the BBS prompts you with the line number of the current line whenever you are at an input prompt. Most useful when you are in insert mode and uncertain of what line you are currently entering.

Example:

If you have just entered the editor (for example, just posting a bulletin), typing /o will turn line numbers on. If you then type /o again, line numbers will be turned back off.

/r -- read lines
Reads the lines -- that is, prints them as they will be seen by whoever reads the mail, feedback, or bulletin you are editing. /r alone, on a line by itself, prints out all lines in the buffer. General syntax applies, see SYNTAX above.

Examples of usage:

```
/r4     -- read line 4
/r3,    --  read line 3, thru the last line in the editor buffer.
/r1,4   -- read lines 1,2,3 and 4.
/r      -- read every line in the editor buffer.
```

/s -- Save your text to disk, and quit the editor.

/w -- Toggle wordwrap on/off.

Copyright 1987 Scott & Eric Green

III.D MCE Commands

What is MCE?

MCE stands for Message Command Evaluator. This allows you to put simple programming-like commands in your messages, if the sysop so allows. Currently, MCE is only accessible in messages that you post in bulletin sub-boards. The sysop can disable it for each sub-board, also. However, the sysop of the BBS can use it anywhere, as can sub-sysops in their own sub-board.

Command format:

MCE commands are typed into your text as part of the text. An MCE command can be put anywhere in your text, with some slight exceptions that will be explained as they occur. MCE commands start with the "english pound" character, that character right between the CLR/HOME key and the "-" key at the top right of your keyboard. The '£' is followed by two characters, which tell what command is being used. Some commands need a third character, an ARGUMENT character, which is used as an operand for the command. For example:

£if0This£elThat£ei

contains three MCE commands: "£if", which takes an argument, "0", "£el", which takes no argument, and "£ei", which also takes no argument. For another example, which uses just any arbitrary character as its argument:

£igyYes!£elNo!£ei

which contains the MCE command "ig", which takes any arbitrary character, in this case "y", as its argument.

Programming model:

MCE consists of several parts:

1) Substitution commands. When one of these is encountered, something else is substituted, for example, the person's name, several backspaces, or a slight pause.

2) Flow-control commands. MCE supports if-then-else style printing control. That is, IF <a condition is true> THEN print this/do these MCE commands ELSE print that/do those MCE commands ENDIF. For example:

You are £if0a sysop£ela luser£ei, GOODBYE!

will print "You are a sysop, GOODBYE!" if you are a sysop, and otherwise print "You are a luser, GOODBYE!".

Copyright 1987 Scott & Eric Green

3) Programming commands. The flow-control commands are actually inter-related with these. The MCE programming model has ten registers, which can hold logical truth values. This allows you to perform more sophisticated flow-control operations than simple testing of access bits. There is also another register, called the ACCUMULATOR, which must hold one of the operands for all two-operand logical instructions, such as the "AND" instruction.

4) Formatting commands. With all the substitutions going on up there, the output can look pretty messy. The formatting commands allow you to make it look nicer by putting as many words on each line as will fit, no less, no more. If the reader has a 40 column screen, the text appears as neat and clean 40 column text, while if the reader has an 80 column screen, the text appears as neat and clean 80 column text.

And now that I've suitably confused you, now for the actual commands:

Substitution commands:

These are the simplest to explain. Basically, when one of these occurs in text, some other character(s) are substituted in their place.

£?%n -- A MCE command with an operand, prints an integer value from the BBS program. Current values for "n" are:

```
0 -- ct%  - Reader's number of calls today
1 -- ut%  - Reader's total calls to the system
2 -- tc%  - Total calls to the system
3 -- ac%  - Reader's access
```

For example, "Your access is £?%3" would print "Your access is 32767" (if you happen to be the sysop, with infinite access).

£?$n -- MCE command, with operand, prints a string value from the BBS program. Current values for "n" are:

```
0 -- na$  - Reader's pseudonym
1 -- rn$  - Reader's real name
2 -- tn$  - Reader's telephone number
3 -- cm$  - Reader's computer type
4 -- ld$  - Last date that the reader called
5 -- b$   - Board number (in board/ud entry files)
             Last user (in welcome file)
6 -- c$   - Board name (in board/ud entry files)
             Logout time of last user (in welcome file)
7 -- in$  - Just a misc string
8 -- a$   - The file name currently being printed
9 -- ll$  - string of dashes, terminated with a
carriage-return
```

£DT -- print the current date.

Copyright 1987 Scott & Eric Green

£BSx -- print 'x' number of backspaces (where 'x' is a single character, between 1 and 9, inclusive).
£PSx -- pause for "x" half-seconds (allowed: 1-9).
£BLx -- print "x" number of bell characters (allowed values for "x": 1-9).
£CS -- clear the screen.

Programming Commands:

This describes flow control and logical operators. There are ten registers, numbered Ø through 9, along with an accumulator, numbered as 'A'. These "registers" can be operated on by a variety of logical operators, described below. In the commands below, you will see things like "CLx". Where you see the lower-case "x" in such commands, you substitute a register number from Ø-9 (or A) for the "x" character. The only exception is "IGx", which will be describe later. First, let's see how we can set and clear flags:

£SEx -- set flag #x
£CLx -- clear flag #x
£CF -- clear all flag registers
£TAx -- transfer flag 'x' to the accumulator.
£FAx -- transfer accumulator to flag 'x'.
£ORx -- do a logical "OR" operation between the accumulator and register number 'x', leaving the result in the accumulator. Description of a logical OR: IF flag 'x' is set OR accumulator is set, set accumulator. ELSE accumulator is clear.
£&&x do a logical AND operation between the accumulator and register 'x', leaving the- result in the accumulator. Description of a logical AND: IF both flag 'x' AND the accumulator are set, THEN set the accumulator. ELSE accumulator is clear.
£↑↑x -- do a logical NOT operation on register 'x'. If register 'x' is set, then clear register 'x'. If register 'x' is clear, then set register 'x'.
£CPx -- compare accumulator to register 'x'. If register 'x' contains the same value as the accumulator, then set the accumulator. Else clear the accumulator.
£GEx -- pause, get a single character from input, when you get that character, put it into register 'x'.
£LDx -- load character 'x' into the accumulator.
£PUx -- put register 'x' to output (assuming it's a printable character, inputted by £GEx).

Copyright 1987 Scott & Eric Green

Now, for the flow-control commands:

MCE allows simple decision making. All decision-making commands have this format:

IF-COMMAND stuff £EL other-stuff £EI

where IF-COMMAND is a condition, such as "£IGY", that can either evaluate to true or false. The text between the condition and the £EL is printed (or evaluated) if the condition is true. Else, the text between the £EL and £EI is printed or evaluated. example:

£IF0 sysop £EL not sysop £EI

which will print "sysop" if the person is a sysop, else print "not sysop". The £EL can be omitted if you are not interested in what happens when the IF-COMMAND evaluates to FALSE. For example,

£IF0£SE0£EI£IF1£SE0£EI

contains two decisions -- if either bit 0 or bit 1 in the person's access is set, it sets flag 0, which can be tested later on (thus allowing access-based decisions to be based upon more than one access bit).

Note that if-then-else style commands cannot be nested and expected to perform normally. For example,

£IF0 £IF1this£ELthat£EI £EL something else £EI

would do NOTHING like what you expect... any behavior that IS observed from nested IF-THEN's is totally implementation dependent, and can be expected to go away in future versions of ICE, or be altered, as the implementation is improved. Instead, use multiple IF statements which alter the states of registers, and then do a single operation based upon the state of a register.

But enough of the preliminaries, now for the actual commands:

£IFx -- IF bit 'x' is set in the person's access, THEN the condition is true, and do everything between the £IF and the £EL (or £EI if there is no £EL). 'x' can be between 0-9 for bit 0 to bit 9. Bit 10 is represented as an 'A', bit 11 as a 'B', and so on up to bit 14, which is a 'E' (ran out of numbers after getting to bit 9!).

£DOx -- DO stuff if register 'x' is set. 'x' is a digit between 0 and 9, or an 'A' if you are referring to the "accumulator". For example,

Welcome to the £DOAPublic Domain£ELSpecial£EI software libraries!

will print "Public Domain" if register "A" is set.

£IGx -- Wait for input. If input is the same as character 'x' (which is any valid character -- but both it and the input character is automatically uppercased) then do the stuff between £IG and £EL (or £EI). Else do stuff between £EL and £EI.

Copyright 1987 Scott & Eric Green

£EL -- invert the condition of the is-printing flag. If the system is currently printing (that is, if the IF, DO, or IG tested true), then turn off the printing. Otherwise, if the IF, DO, or IG tested false, turn the printing back on. If encountered in ordinary text, flips the is-printing flag anyhow, usually turning off printing (it has no way of knowing that there was no condition preceding it).

£EI -- turn the printing back on, period.

Justification:

With all of the above substitutions and if-thens going on, the output is likely to look pretty messy. Turning on justification makes it look a lot neater, because as many words as will fit, are put onto a line. A line is assumed to have as many columns as the user has, so the bulletin will appear in 40 columns to 40 column users, and 80 columns for 80 column users, no matter what width screen the original poster had. This is simuliar to the justification process performed by word processors, except MUCH simpler. Now for the commands:

£JU1 -- Turn on justification. Note that this command must be at the beginning of a line in order to work properly, because otherwise, how does the justifier know what column you're at in a line, if you don't start at column 0?

£JU0 -- turn off justification. Should occur either at beginning or end of a line, otherwise messy printing may result.

£PP -- New Paragraph command. Prints a carriage return, then four spaces, and starts printing words again.

£CR -- Prints a carriage return (starts on a new line).

End of section III.D

Copyright 1987 Scott & Eric Green

III.E Bulletin Commands

Basic concepts:

The bulletin section is where you post and read articles, or "bulletins". There are many different bulletin subsections about different topics, and each bulletin can be responded to. The responses to a bulletin are attached to that bulletin, unlike on cruder systems where a response is just a

Note that the library-oriented up/download subsystem maintains a file which contains information about each file uploaded or added to the sub-library. This file contains data such as the name of each file, the name of the person who uploaded it, the date it was uploaded, and a 1-line description of the file. Thus, you cannot blithly swap disks in and out of a drive if you are using a library-oriented up/download subsystem, you must add each file into the sub-library's information file. You can do this by using the "U" command from local mode, which "uploads" a file (meaningless in local mode, so the BBS just assumes that the file exists already on the disk, and just needs to be added to the information file).

Also note that this information file is maintained on the same disk as the up/download sub-library in question. This in general means that you have to copy files off the disk, and can't just swap disks. In public domain use, where you have to filter the programs anyhow based upon what programs are in your disk library or your club's library, this is no real problem. For other circumstances, an exchange oriented sub-library may be in order.

A good organization is to have library-oriented upload sections for the general public for public domain use (so that you know who uploaded each file, when, and can chastise people who dare upload copyrighted software to such a library). Use exchange-oriented download sections for swapping disks from your disk library in and out of the drive upon request, or for cases where you receive so many uploads that you must put a fresh disk in the drive every day.

Commands:
(in alphabetical order)

"+" -- go to next available up/download library.
"-" -- The opposite of "+"

For example, if you are in up/download library #4, and the next accessible library is library #6, typing "+" will get you to library #6.

x -- move to up/download library #x.

For example, if you just typed "UD", you are in up/download library #1. If you then type "5", assuming that there is a library 5 and you have access to it, you will be placed in up/download library #5.

Copyright 1987 Scott & Eric Green

A -- print information "About" files
Ax -- print "about" files, starting at file #x.

Acts much like the "S" command, except that because of the quantity of data it prints, it only prints information about four files at a time. Prints the name of the file, the person who uploaded it, how many blocks are in the file, the computer that the file is for, and a brief one-line description of the file's purpose.

AN -- about new
Prints the above information, but about "new" files (files uploaded since the last time you logged on -- see the "LD" command to change that last time).

BS -- change block size.

Changes the current block size, used when downloading (when uploading, whatever block size you've set on your end is the one in effect). This is useful when the lines are noisy and 255 byte blocks have trouble getting through... 127 byte blocks are especially effective.

Dx -- download file #x EX: D9

This allows you to receive ("download") a file from the BBS. The BBS will give you a chance to abort the transfer. If you choose to continue on and download the program, you must then go to your terminal program's main menu and select your "receive-a-file" or "download-a-file" selection.

If you are using Punter protocol, you may see the BBS printing GOOGOOGOOGOOGOO constantly on the screen. This is the Punter ML telling you to GOO to your terminal program's main menu and select your terminal program's "receive-a-file" option. If you wish to abort the transfer at this point, two control-X characters will abort.

Xmodem protocol gives you no such reminder, it just sits there waiting for you, and eventually gives up. Again, typing two control-X characters will abort.

The sysop can also abort any file transfer at any time by pressing Commodore-A on his keyboard.

IMPORTANT You can only select the "download" option if your terminal program supports the New Punter or Xmodem error-checking protocol. See the notes under "Summary" at the beginning of this document.

Copyright 1987 Scott & Eric Green

L -- list available up/download libraries.

Lists the name and number of all up/download libraries to which you have access.

P -- select a new up/download protocol.

Choose the default protocol you wish to use for file transfers. Current choices are Xmodem, CRC Xmodem, Sizing CRC Xmodem, and New Punter. A sizing Ymodem and Kermit may be added to later versions, if they'll fit into the limited available memory.

One of the enhancements to version 1.1 of the program is that whenever you select this option, your default up/download protocol is stored with your user record, to be used on all future calls until you again use P to change it to something else.

S -- scan titles of files in current library.
Sx -- scan, starting at file #x

This prints the filenames of all files in the current up/download library, starting with the last file that you scanned, read "about", or downloaded. It starts at file 1, and wraps around when the last file in the library has been "scanned".

The "Sx" option allows you to start scanning at file #x. For example, if there are 20 files and you type "S10", it will print the filenames of the last 10 files in the current up/download library.

SA -- scan new all

SA does a "scan new" (see below) on all accessible library-style sub-libraries (since dates aren't stored in exchange-mode sub-libraries, it skips those).

SN -- scan new

This prints the title of all "new" files on the current sub-library. A "new" file is one that has been uploaded since the last time you were on -- that is, you haven't seen it yet, so it's "new" to you. See the "LD" command in main menu to change the definition of what's "new" to you.

U -- upload a file

When invoked in local mode by the sysop, this allows the sysop to add a file to the u/d library. Note that the file must have already been copied to the disk.

When invoked the usual way, by a user, this allows you to transmit or "upload" a file to the BBS. *IMPORTANT* You must

Copyright 1987 Scott & Eric Green

have a terminal program which supports Xmodem or New Punter protocol in order to use this command. See the section "Introduction" at the beginning of this document.

The BBS will ask you many questions, and after you have answered them, it will tell you to select the "send" or "upload" selection of your terminal program. If you are using Punter, it will then start saying GOOGOOGOO as with the "download" command. If you are using Xmodem, it will sit there waiting for you, doing nothing. Sele the "send" or "upload" option of your terminal program. If you must abort (e.gt "file not found error"), transmitting two control-X characters to the BBS will abort Xmodem transfers (see "D"). Steve Punter included no remote abort option in his protocol, which is also brain damaged in several other subtle ways, such as idiotic timing requirements which don't work over Telenet or Tymnet (although it is still the best protocol for locally transferring files between two Commodore computers). However, since he published the source code to his protocol, I was able to modify it, too, so that it would abort upon receiving two control-X characters.

The sysop can abort any kind of file transfer at any time by typing Commodore-A on his keyboard.

This command will greatly enhance your sysop's opinion of you. Consistent use of the "D"ownload command without making use of the "U"pload command is likely to result in you losing access to the up/download system of your local BBS, or result in your being put onto a block-credit system, so that you MUST use the U command if you wish to use the D command. If you are at a loss for programs to upload to your local BBS, and you have downloaded many files from that BBS (saving much money in the process), use that money to call a long distance out of town BBS to help recompense the sysop and other users of the system for their time and effort in gathering together the files available for download.

V -- View sub-board sysop.

If the current up/download library has a sub-sysop (a person in charge of maintaining it), print his user number.

Commands documented in main menu documentation:

B - bulletin section
UL - user list section
T - date/time/last
F - leave feedback to the sysop
C - yell for chat
H - help file
PF - program files
GF - General text files
NF - System News files
MANX -- maintenance section

Copyright 1987 Scott & Eric Green

Up/download problems:

"It just prints out GOOGOOGO and nothing happens!"

Well, when it says GOOGOOGOO, it's expecting a New Punter file transfer. Select your terminal program's SEND or RECEIVE option. If you are UPLOADING, that means you are SENDing to the BBS. When you are DOWNLOADING, that means you are RECEIVING from the BBS. Think of the BBS as a huge mountain, and you and your puny C-64 at the bottom, and the up/download section at the very top...

"I did that. Still, nothing happens!"

Make sure your terminal program supports New Punter or Xmodem protocol. ***WARNING*** VIP term WILL NOT support New Punter protocol! The "Punter" that it supports is OLD Punter, which is totally incompatible with New Punter and this BBS. Most commercially-purchased programs, such as Vidtex or VIP-Term, will not support New Punter protocol. Some of the newer ones such as Bob's Term will, but you are much better off just going to your nearby Commodore club and getting one of their fine public domain programs. If you have no nearby Commodore club, TPUG (Toronto Pet User's Group) has a large disk library available for a semi-reasonable charge. See their advertisements in any large Commodore magazine (they advertise about twice a year in each of the magazines, so you might have to look in back issues).

"I try to download, and I get so many bad blocks that it takes an hour to download a 5 block program!"

Try changing your block size to transmit smaller blocks. When downloading, before you download, use the BS command on the BBS. When uploading, before you upload, select your terminal program's block-size command. This happens most often when you're on a noisy company PBX... some modems have settings to compensate for this problem (the 1670 even has an external DIP switch), check with your dealer or local Commodore user's group.

"When I try to download, it tells me that I have no block credits left!"

There's several possibilities here:

a) The sysop of the board gave you a block credit ratio of zero because he doesn't like you, in which case the solution is to talk to the sysop and detirmine the reason for his dislike,

b) The sysop of the board is a lamer and gave all the users of the board a ridiculously low block credit ratio, like 2 download blocks for every uploaded blocks, so that even though you've uploaded fairly and frequently, you still can't download as much as you've earned. The solution to this problem is to talk to the sysop, and remind him that with such a policy, people will either avoid his board, or start uploading garbage and making fake logins after being deleted for uploading garbage. He may say "Well tough sh*t lamer!".

Copyright 1987 Scott & Eric Green

Then tell him that the authors of ICE themselves suggest that he stop being such a horse's a$$. At this point, he will probably delete you. In which case, GOOD. A sysop like that doesn't deserve to have fine users calling his board.

c) The third possibility is that you really ARE a lamer, and haven't uploaded enough software to entitle you to downloading as much as you wish. There's many sources of public domain programs -- magazines, computer clubs, on-line services such as Quantum Link and Compuserve, and even other BBS's. If you're too cheap to contribute, and only want to take take take, then you DESERVE to be tarred, feathered, and ostracised.

Copyright 1987 Scott & Eric Green

III.G.4 Exchange-mode U/D Commands

Introduction:

You go to a up/download sub-library. It says "* Whole Disk Exchange Mode *". You type "S". Nothing happens. So what's going on?

You are using the EXCHANGE MODE up/download subsystem. Some sysops are not interested in the features of the library style up/download system, and want the freedom to be able to swap disks in and out of the download drive(s) upon request. Thus, the exchange mode, which implements an up/download subsystem similar to 64Exchange or 6485. Any library whose name starts with "E-" is an exchange-mode library.

This sort of subsystem is most useful on systems whose main purpose is mass file transfers, which is why it was not in the original ICE (which was used mostly for bulletins and public domain software). Unless you have a spare drive or two hanging around to put the exchange-mode library(s) on, you'll probably want to use the library-style subsystem, so that you don't have to dedicate a whole drive for up/downloads and can use features such as the one-line description of each file. This is especially true if you are using a hard disk, since you have to copy the files off the hard disk anyhow.

An additional benefit of exchange mode up/download libraries is that you can do multifile uploading and downloading using Commterm, Phoneman, Nezterm-128, or some other compatible terminal program. Note the mention of Commterm. ICE works perfectly with Commterm, unlike some other BBS programs which occasionally lock up. Multifile uploads, as well as downloads, work perfectly with ICE.

Using exchange mode:

@

allows sysops & library sysops to delete files, validate your disk, and issue other DOS commands.

$

allows you to see the directory of the disk. It asks you for a MATCH STRING. Pressing return means do a directory on everything. However, if you type some characters, a "*" is added to them and the directory is opened with that file name, instead of as just "$0:*" or "$1:*". For example, entering "m." means that every file whose name begins with "m." will be listed.

Copyright 1987 Scott & Eric Green

D and U
work as for regular up/downloads, except you have to enter a filename when downloading.

L
works as usual -- lists all accessible up/download libraries.

MD
is multi-download. It asks you for a match string (like the "$" command -- just press RETURN for all files), then takes you through all the filenames that match, asking if you want to (Y)es, download it, (N)o, don't, (B)egin the download, or (A)bort the transfer. After it's finished or you type (B)egin, it gives you 15 seconds to go to your multi-receive mode.

MU
is multi-upload. Self-explanatory.

MU and MD use the Commterm/Phoneman/Nezterm multifile protocol. They work in both Punter and Xmodem, however, note that most Xmodem terminal programs do not support the Commterm-type multifile protocol, but, rather, an older multifile protocol derived from CP/M conventions.

P
allows you to switch between Xmodem and Punter protocol.

Note that Commodore-A aborts ANY kind of transfer -- Xmodem, Punter, or multi-file. Also note that multi-file transfers are only valid for programs which implement the Commterm-style multi-file protocol. There is another protocol for multi-file Xmodem transfers (which we don't implement), but it is only valid for MS-DOS and CP/M, which have very restrictive file names (8 characters plus a 3 character extension). It cannot be adapted to Commodore DOS's wide range of possible file names.

R --
Read a sequential text file. If the file is in Commodore ASCII, it will look fine. If it is in Real Ascii, the case of all characters will be reversed. If it is in some other format, such as Paperclip screen-codes, it will look like total garbage. The R command will not let you read program files.

Copyright 1987 Scott & Eric Green

Commands documented in main menu document:

B - Enter bulletin section
UL - User list subsystem
UD - Enter up/download subsystem
F - Leave feedback to the sysop
C - Yell for chat
PF - Enter program file section
GF - Enter general file section
NF - Enter news file section
OFF -- Leave the system
MANX -- system maintenance

Copyright 1987 Scott & Eric Green

III.H User List Commands

Introduction:

The User List subsystem is really very simple. It allows you to list the users of the board. There's only four commands available for normal users:

A --
List all users. Allows you to pick a number at which to start.

S --

Search for a specific name. Asks for a name, then asks for a user number at which to start.

CT --

Search for people with a specified terminal type, starting at the specified user.

Q --

Quit to main menu.

In addition, the following commands are brought in from main menu:

[B] Board Level
[C] Yell for chat
[M] Mail section
[UD] Up/downloads
[OFF] Logoff
[T] Time/Date/Last
[F] Feedback
[H] Help section
[NF] News files
[PF] Program files
[GF] General files
[LD] Change last call date

Copyright 1987 Scott & Eric Green

III.I ICE File-handling subsystems

Introduction

There are three different types of files in I.C.E. Identical commands are used to manipulate each of these, due to the similiar nature of handling files (no matter what the actual nature of the file is). These files are organized into DIRECTORIES, generally according to fixed classifications. You can see these classifications by typing "J" (Jump to a directory) at the files menu. More information on that when we get into the commands summary section. First, a brief description of the various types of files:

The **PROGRAM** files

or "pfiles" consist of overlays which can be loaded in upon demand. These are mostly for the provision of optional sysop utilities, however, some small games may be provided here. This section is reached by typing "PF" at any menu prompt.

The **GENERAL** files

or "gfiles" consist of text files which are intended to be read by the user. This may be things such as a BBS list, description of the Punter protocol, or a list of operating system entries. This section is reached by typing "GF" at any menu prompt. Note that "gfiles" are regular 80-column (max) Commodore Ascii files, such as are produced by Easyscript or the ICE editor (yes, you can use the ICE editor in the maintenance menu to write a gfile containing, say, BBS rules, if you only have 50 lines of BBS rules). You do NOT have to format ICE GFILES in any special way, such as surrounding each sentence with quotes. Since a machine language routine prints out the file, it's not bothered by the limitations of BASIC.

The NEWS files

consist of the introductory messages which you see upon logging in. Obviously, there is not multiple news directories. However, the NF system allows you to re-read those introductory messages, which generally consist of things such as BBS rules and regulations, information about the club that owns the BBS, or other such things.

Copyright 1987 Scott & Eric Green

Commands
(in alphabetical order)

x -- read or run file #x!

If you are in the gfile section and there are 9 files, typing "8" at the menu reads the eighth file. Remember that you can use control-s to pause while reading a file, and the space bar to abort reading a file (see the "misc" section of the manual). If you were in the pfile section and there are 5 files, typing "4" would load the fourth program overlay, and run it.

E -- Edit list of file libraries

This allows the sysop to create and edit the file which is printed out when you type "J". This is just an ordinary text file. It is this fact that limits you to 50 GF or PF libraries (along with disk space, of course). In The Real World, though, that really wouldn't be practical, anyhow -- people would get real tired of typing "J" and seeing 50 lines of text.

FM -- Files maintenance

This is a sysop command only. It allows the sysop to add and delete files from the libraries, change library names and accesses, and other such goodies. Note that you have to copy the files by hand to the appropriate disk (which you set up at configuration time), before you can use this command to add them to the file directory. The exception is when you add a file to the system news files -- "FM" dumps you into the editor to write your text, in that case. Note that you must [S]ave the directory before you [Q]uit -- if you [Q]uit before [S]aving, you lose all your changes (except that files scratched off the disk stay scratched, alas).

J -- jump to a P/Gfile library

Typing "J" at a files prompt prints a list of libraries or directories available, and allows you to go to one of them in order to read/run its files.

Jx -- jump to P/Gfile library #x

If you already know which library you wish to go to, you can just type J2, J4, or whatever the library number is, without having to wait for the BBS to print a list of libraries such as the "J" command provides.

L -- list files

Lists all files in the current library/directory. To invoke, just type L at the files menu.

Copyright 1987 Scott & Eric Green

Commands documented in main menu section:

B - Enter bulletin section
UL - User list subsystem
UD - Enter up/download subsystem
F - Leave feedback to the sysop
C - Yell for chat
PF - Enter program file section
GF - Enter general file section
NF - Enter news file section
OFF -- Leave the system
MANX -- system maintenance

Copyright 1987 Scott & Eric Green

And the swiftly swirling sluffy slouched off into the sunset as the slithery sthnake snuck towards the eggs of the wombabbing zambidator, a common parasite in the tropics of Outer Orion, as you, gentle reader, wonder what kind of journey you've been taken on, in the phasmatastic conflagoration of fire and dazzle, and wonder if you have, indeed, reached.....

THE END

(which, as you know, generally terminates these sloppy grotesquities of cheap pulp sci-fi novels, but, never fear, we shall have at least 15 sequels, and make "Fairy Godmother of Dune" and "Great God-Nephew of Dune" and volume XXIV of "Invasion:Earth" look great by comparison!)

Copyright 1987 Scott & Eric Green

IV. THE TECHNICAL MANUAL

IV.A File Formats

This section of the manual describes various file formats. NOTE: Some of our internal documentation has not been updated for version 1.1, and some of our internal documentation dates back to even before the very first version 1.Ø of the program (August 1986), so there MAY be errors. However, we have made every attempt to be as accurate as possible.

s.config file -- master configuration file:

```
device  userfile -- dv%(Ø,1)
drive userfile -- dv%(Ø,2)
device prg files -- dv%(1,1)
drive prg files -- dv%(1,2)
device gfiles -- 2
drive gfiles
device system files -- 3
drive system files
device email -- 4
drive email
pf$ (ID prefix string)
sd%(8)-sd%(15)
o1%,o2% -- hang up modem
p1%, p2% -- pick up modem
cc% -- carrier mask
mt% -- modem type
COPYRIGHT 1986
      BY
ERIC AND SCOTT
     GREEN
```

The USER FILE:

File "s.user.config" on the system disk. The user file consists of one record in a relative file for each user. Each record is divided into three strings. User information is stored in a stringified form, and for partial reading of information, the relative-file byte-positioning offset is supplied. User information starts at record #11 in the file. Records 1-1Ø are reserved for system use only, for information such as the total number of users in the file, total calls to the system, calls logged, etc. If you absolutely MUST steal a record for your own uses, try to make it a high-numbered one, since those will, of course, be the last ones that we ourselves use.

Format of first record:

The first record is a single string.

```
mid$(in$,1,5):tc% (total calls to the system)
```

Copyright 1986 Scott & Eric Green

mid$(in$,6,5):cl% (calls logged since last log erasure)
mid$(in$,11):ur% (total number of users).

Format of 11th through nth records (each user):

String #1: Starts at relative offset #1 in the record:
pw$ -- the user's password.

String#2: Starts at relative offset #16 in the record:

ct% (calls today) - 3 characters -- 1-3
ut% (total calls) - 5 characters -- 4-8
ul% (upper/lowercase) - 1 character -- 9
ll% (line-length) - 2 characters -- 10-11
tu (total uploaded blocks) - 6 characters -- 12-17
td (total downloaded blocks) - 6 characters -- 18-23
tb% (total bulletins/responses) - 5 characters - 24-28
rn$ (real name) - remainder of string, starting at position #29, maximum size of 19.

String#3: Starts at position #65 in the record:

dp% (download credit ratio): 1-3
cp% (calls per day): 4-6
tp% (time per call): 7-9
ac% (access): 10-14
cm% (computer type): 15-16
ld$ (last call date): 17-33
tn$ (telephone number): 34-45
pt% (default u/d protocol): 46
lb% (backward hash-link): 47-50
lf% (forward hash-link): 51-54
na$ (user pseudonym): rest of string, starting at 55, maximum size of 19.

s.subs -- list of bulletin sub-boards:

ba%(0) # boards (1-30)

for each boards x:

bn$(x): board #x name
ba%(x): board #x access
board #x device - dv%(x+11,1)
board #x drive - dv%(x+11,2)
ss%(x) board #x subsysop ID
an%(x) board #x flags: bit 0=anonymous, bit 1=post, bit#2=entry file, bit#3 = mce-allowed.

s.u/d 0 -- list of u/d areas.

ba%(0) # boards (1-30)

for each U/D area x:

Copyright 1986 Scott & Eric Green

```
    bn$(x): area #x name
    ba%(x): area #x access
    area #x device - dv%(x+11,1)
    area #x drive - dv%(x+11,2)
    ss%(x) area #x subsysop ID
    an%(x) area #x flags: bit#2=entry file
```

s.u/d x -- list of u/d libraries (where "x" is the area number):

```
ba%(Ø): number of libraries.

    for each library x:
bn$(x): library #x name
ba%(x): library #x access
library #x device - dv%(x+11,1)
library #x drive - dv%(x+11,2)
ss%(x) library #x subsysop ID
an%(x) library #x flags: bit Ø=upload, bit 1=download,
    bit#2=entry file, bit#3 = freeze uploads
```

s.log -- log of callers to system.

```
    name id$
    login time
    logout time
    status (i.e: normal/carrier lot/fmail deleted)
    <uparrow>
```

Copyright 1986 Scott & Eric Green

IV.B Variables

List of variables:

```
     dv%(40,2): device = 1, drive=2, 0-10 for system, 11-40
for u/ds,boards
       0=userfile 1=prg files 2=general files 3=system
files/menus/news 4=email/forced mail
     bn$(30) - board names
     an%(30) board flags
     sd%(15): slow drive/fastload enable for each device
     ed$(51):editor
     ba%(30):sub-board or u/d access word
     ss%(30): sub-board or library sub-sysop ID #
     pt$(30): post title;ud file name
     pd$(30): pot date;date of uload
     pa$(30): author; uploader
     id%(30): id# of who posted or uploaded
     pr$(30): last response date, 30 char description
     nr%(30): # responses, # of blocks
     ba%: 1=board config loaded
     ud%: 1=u/d config loaded (what about areas, Scott? Do
they have their own???)
     af%: result of gosub to strout/fileout -- holds
interrupt character.
     pf$: 1-5 character prefix string
     tc%: total calls to system
     cl%: calls loged
     ur%: # of users on system
     cs%: upper/lowercase the input
     pt$: prompt string
     pg$: current prg file loaded
     cc%: carrier type (16 if neg/0 if pos)
     o1%,o2%: bytes to hang up modem
     p1%,p2%: bytes to pickup modem
     lc%: local mode 0=no 1=yes
     mt%: 0=can poke it off, 1=must do +++ ATH trick
```

Copyright 1986 Scott & Eric Green

IV.C: BASIC subroutines

Note on pfile programming: When a pfile is invoked, the pfile loader GOTO's line #10. When a pfile has a menu, and calls the global commands routine, the global commands routine does a goto100 when it's finished, so your pfile's menu must start at line 100.

```
40210 read just name (user file)
40000 convert user info into b$,c$,d$
40100 write user info
40200 read in b$,c$,d$
40300 convert into user info
40250 read tc%,cl%,ur%
40500 convert back to b$,c$,d$
40400 ???
???
 pw$ = pa$(5)
 ct% = nr%(0)
 ut% = nr%(1)
 ll% = nr%(2)
 tu% = nr%(3) (hmm, where's tu in 1.1???)
 td% = nr%(4) (hmm, wonder where td is in 1.1???)
 tb% = nr%(5)
 rn$ = pa$(0)
 dp% = nr%(6)
 cp% = nr%(7)
 tp% = nr%(8)
 ac% = nr%(9)
 ld$ = pa$(1)
 tn$ = pa$(2)
 cm% = nr%(10)
 na$ = pa$(13)

12000 B
12020 UD
12100 UL
12110 M
12120 MANX
14000 = global menu routine
12130 a$="main":goto20010 -- run main menu
16000 read news dir
16500 read gfile/pfile dir
  a$=name
  xx=lib #
  x= 1 for pfile 2 for gfile
  ul = 0 for news 1 for gfile 2 for pfile

2700 chat call
2800 time/date
2900 read board/area/ud config a$=which
3000 read sub-board/library dir
  xx = subboard #
  a$ = filename
  a% = -1 (on return?) if they don't have access
```

Copyright 1986 Scott & Eric Green

```
3100 list boards you can access
1355 set a and b to numbers lp must be max+1 a$ must be set
to phase
3200 print editor buffer out to file #2
3210 clear editor
2540 do both
3300 "v"iew sysop
30250 print commands to modem
1990 editor entry for when there is already something in it.
im%=.:ln%=.
13000 load protocol a$ = prot name

I/O subroutines:
500 print line -- no interrupt, no CR
501 print line -- no interrupt, CR
402 print line -- interrupt, CR
540 open,print file, interrupt,stop when space pressed, close
file
550 print in$(no interrupt)
560 print file, interrupt, read till "↑"
570 "skip" routine.
480 open,print,no interrupt,close file
4000 get in$ from disk?
4010 date compare ld$,a$ -- set a% 1=new
503 print in$(interrupt)
505 input in$ from modem, cs%=in what case.
590 print 1 CR
591 print 2 CR
507 get in$ (single-character)
510 relative file positioning
520 open a file dv%=device ou$=file name (with ,r or ,w or
whatever).
525 open error channel dv%=device
530 read error channel into a%,a$,b%,c%
540 print-a-file (cs%,sp%,mm%) a$=name x=device # of
dv%(40,2)
550 print in$(like print ou$)
560 print-a-menu (cs%,sp%)
900 system startup
1000 editor subsystem
21000 load overlay a$=name dv%=device
22000 chat mode
20000 load a$: goto10 (for loading pfiles)
20010 load a$: goto100 (for loading ML?)
20020 open "s.user config"
21010 print in rea a$ (module name)
21020 print in area pt$ (prompt)
21030 print[tc;cl;usrs)
20030 opena$ x=device (0-4) a$=name+read/write
30000 logout (ask'em)
30100 logout (gone!)
2100 check blocks free x=device 0-4
fb=blocks free returned
2200 check dir entries if a%=-1 then no free entries
2300 check both
```

Copyright 1986 Scott & Eric Green

Copyright 1986 Scott & Eric Green

IV.D ICE ML

I am using CASM under the C-Power environment as my assembler, due to its linker, which allows me to assemble only the module that I just changed, instead of me having to re-assemble 5,000 lines every time I make a minor change. The following file is taken from my assembly language source code, and is the list of labels which are globally defined (this list is included with a "#include" statement in all of my source files):

```
; file: labels.h

.nlst
; this file contains definitions of
; all the labels of interest, plus
; a quick overview of other various
; thingies:

mce$char = '\      ;char to delimit mce cmds

;
;  memory configuration:
;    9a00-9aff  ml jump table
;    9b00-9bff  global variables
;    9c00-9dbf  basic jump table
;    9dc0-9dff  irq target
;    a000-b200  my i/o routines
;    b200-bcff  ml routines
;    bd00-c000  translation tables
;      be00 ascii to c64
;      be70 ascii to printable
;      bf00 c64 to ascii
;    c000-c800  punter/xmodem/(editor)
;    c800-d000  fastloader
;
;  basic jump table:
;     preceded by "-" = scott's.
;
;   9c00,9c08,9c10,9c18,9c20,9c28,
;   9c30,9c38,9c40 -- obsolete,
;    not used (hopefully!).
;
; 40008 9c48 strin -- get in$ from modem
; 40016 9c50 strout -- print ou$ to modem
; 40024 9c58 cls -- clear screen/init stuff
; 40032 9c60 irqset -- setup irqs
; 40040 9c68 ovlay -- load overlay
;   9c70 -- obsolete
; 40056 9c78 prtfil -- print file to screen
; 40065 9c80 diskin -- input in$ from disk
; 40072 9c88 printinp -- print in$ to screen
; 40080 9c90 init -- init screen & irqs
; 40088 9c98 prtstf -- print ac% etc.
; 40096 9ca0 mkdt -- make dt$
```

Copyright 1987 Scott & Eric Green

```
; 40104 9ca8 cmpdt -- compare a$,ld$
; 40112 9cb0 dir -- do disk directory
; 40120 9cb8 ldstf -- load screens
; 40128  oucr -- string out with cr
; 40136 outnon -- string out, no cr, no irq
; 40152 passwd -- get a password
; -40160  screen all
; -40168  screen few
; 40176  chat mode
; 40184  skipit (skip a block in file)
; -40192  chat whoop noise
; 40200  basget (char into 254)
; 40208  carrier (called from punter)
; 40216  i$chkdv (check for device)
; 40224  i$term -- simple terminal
; -40232  clear window & color.
; 40240 initialize mce
; 40248 clear mce
; ml jump table:

bn2dec = $9a00  ; 16 bit to ascii
concat = $9a03  ; concat char to b2s buffer
window = $9a06  ; print .a to text window
cls = $9a09     ; clear text window
prttim = $9a0c  ; print time
prttr = $9a0f   ; print time-remaining
prtacs = $9a12  ; print access
prtcpd = $9a15  ; print calls-per-day
prtdpc = $9a18  ; print downloads-per-call
prtidl = $9a1b  ; print idle time
prtdt = $9a1e   ; print date
prttpc = $9a21  ; print time per call
; $9a24 fkey -- obsolete ...
getch = $9a27   ; getchar modem/keyboard
putch = $9a2a   ; putchar modem/screen
getstr = $9a2d  ; get in$
putou = $9a30   ; put ou$
carr = $9a33    ; check for carrier
irqin = $9a36   ; save bn2dec entries etc.
irqout = $9a39  ; restore them.
irqsbr = $9a3c  ; actual irq routine -- test point.
irqinit = $9a3f ; init irq vectors
setirq = $9a42  ; obsolete, replace someday
clrirq = $9a45  ; obsolete, replace someday
ovlay = $9a48   ; load an overlay
prtfil = $9a4b  ; print file to screen
dskin = $9a4e   ; input in$ from disk
prtinp = $9a51  ; print in$ to screen
init = $9a54    ; init screen & irqs
prtstf = $9a57  ; do all the prints up der
mkdt = $9a5a    ; make date/time string
cmpdt = $9a5d   ; compare a$ to ld$
dir = $9a60     ; do disk directory
ldstf = $9a63   ; load screens (obsolete?)
openrs = $9a66  ; open rs232 port?
```

Copyright 1987 Scott & Eric Green

```
outcr = $9a69
outnon = $9a6c
outirq = $9a6f
passwd = $9a72 ; get a password
fkey = $9a75   ; scott's function key code
scrall = $9a78 ; screen all
scrfew = $9a7b ; screen few
fason = $9a7e  ; fastload on
fasoff = $9a81 ; fastload off
skipit = $9a84 ; "skip" routine
carrier = $9a87 ; carrier detect routine


; ml variables:
inbuf = $9b00    ; input buffer
tr = $9b51       ; 39761 - time remaining
cpd = $9b52      ; 39762 - calls per day
dpc = $9b53      ; 39763 - downloads per call
tpc = $9b54      ; 39764 - time per call
idle = $9b55     ; idle time
day = $9b56
month = $9b57
year = $9b58
cr$eol = $9b59     ; 39769 - cr @ eol?
localmd = $9b5a    ; 39770 local mode - 1=true 0=false
putend = $9b5b     ; 39771 ending status of putou$
ll = $9b5c         ; 39772 line length
caps = $9b5d       ; 39773 capitalization (1=true)
caridl = $9b5e     ; 39774 carrier & idle
cartridl = $9b5f   ; 39775 carrier, time remaining, & idle
idleon = $9b50     ; 39775 idle on /off (1=on)
chaton = $9b61     ; chat flag (1=on)
chatflash = $9b62 ;39778 chat-flash flag 1=flash
editmd = $9b63     ; 1=editing a line in editor
filenum = $9b64    ; file # for all ml disk operations
;    ^^^^^ is also used as device for i$chkdv.

wrapmd = $9b65     ;39781 - 1=word wrap on.
mce$state = $9b66 ;state of mce interpreter
mce$out = $9b67   ;1=mce on output
mce$in = $9b68    ;1=mce on input
mce$ifthen = $9b69 ;0=ok,1='else',2='endif'
m$iflvl = $9b6a   ;0=ok, >0=in if-then-else.
m$format = $9b6b  ;0= no formatting, 1 = format.
output$ll = $9b6c ;info for formatter -- what line length?
nouparrow = $9b6d ;1=ignore uparrow on read-a-file
nospecial = $9b6e ;1=no special char substitution on input
p$gooblock = $9b6f ;transfer protocol goo/bad indicator.
p$size = $9b71     ;transfer protocol block size
; 9b72-9b73 -- obsolete.
diskst = $9b74     ;disk status - returned by dskin etc.
abortfil = $9b75   ; 39797 - 0 = abort reading file when
irq'd.
dtresult = $9b76   ; 39798 - a$/ld$ comparison --
0=a$<ld$,1=a$>=ld$)
```

Copyright 1987 Scott & Eric Green

```
menumd = $9b77      ; 39799 0=any character aborts output
carmask = $9b78     ; positive/negative carrier? 0=neg,
16=pos.
irqabl = $9b79      ; 39801 0 = can interrupt print string
with space or '/'
hangups = $9b7a     ; modem hangup bytes.
slowdrv = $9b7c     ; slow drive (for overlay loader)
dtbuf = $9b7d       ; -- $9b8f -- date/time buffer for mkdt.

; dddddatt's it fer now!

; miscellaneous other variables of interest:
vartab = $2d
 .list
```

Copyright 1987 Scott & Eric Green

Appendix A: Glossary

access: A word assigned to a user, whose bits detirmine what parts of the BBS system that a user can enter, via using the AND operation on each consecutive bit with the other input being the access word of the sub-board or library being entered. If two bits match (their logical AND is true), then the person can get into that part of the BBS.

and: One of the three fundamental operations of digital (computer) logic (the others are OR and NOT). You feed it two inputs, and if (input1 is TRUE) and (input2 is TRUE), then (output is TRUE). Used as the basis of the access system.

anonymous: name-unseen. When applied to postings, it means that the other users of the board cannot see your name (although the sysop, of course, can).

area: The up/download portion of the system is divided into up to 30 areas. An area is just something that can hold up to 30 up/download libraries. Up/download libraries are what keeps track of what files are online for you to download. Thus, there can be up to 900 up/download libraries in the system.

bit: the fundamental quantity of digital electronics: a 1 (TRUE) or 0 (FALSE), represented in your computer as 5 volts and ground, respectively. See AND, BOOLEAN ALGEBRA.

BBS: Bulletin Board System, a system whereby users can hold discussions, send mail to each other, and exchange public domain programs.

board: a) the entire bulletin board system. "I called the board yesterday, and there wasn't any new postings on it!". b) one of the sub-boards of the BBS. "I posted a message on the Programming Board about my problem." See SUB-BOARD.

boolean logic, boolean algebra: Traditional bi-natured logic ("If this is true and that is true, then the result is true") as expressed mathematically by George Boole when he invented Boolean algebra for mathematically manipulating logical quantities (thus making the computer possible). For more information, go to your nearby university bookstore, and buy a book on fundamental digital electronics.

bulletin: Text which is posted to a sub-board, which is in public view and can have responses (cf.) added to it. Sometimes called a MESSAGE.

buffer: A place where data is temporarily stored, such as the editor's text buffer (where it temporarily stores text while you're editing it), or the disk drive's sector buffers (where the disk drive saves text until it has enough to write to disk).

Copyright 1987 Scott & Eric Green

brain-damaged: generalization of HONEYWELL BRAIN DAMAGE, a mythical disease invented by Real Hackers at MIT to explain certain utter cretinisms in design of the Honeywell Multics computer (a large mainframe of the 60's and early 70's). Now generally used to describe any piece of equipment whose designer obviously only had "half a brain" (gee, can we call the 1541 brain-damaged?).

bug: an error, flaw, or defect. Also see FEATURE.

calls-per-day: the number of times that a person can call the BBS during a day.

co-sysop: a person who helps the SYSOP (cf.) run the BBS, by killing old bulletins, verifying names and telephone numbers of new users, etc.

directory: a) List of files on a diskette, maintained by the disk drive. b) list of files in an up/download library or other such library, maintained by the BBS program.

download: to recieve a file from the BBS (or any other source) via your modem.

downloads-credit-ratio: The number of blocks that a person can DOWNLOAD(cf.) for each block that he UPLOADS to the BBS.

email: electronic mail, sent by you to another user. It can only be read by that other user, and cannot be read by anyone else.

false: One of the two fundamental values of bi-natured Boolean logic (the other is TRUE), represented as a 0 numerically, and as 0 volts at the hardware level. See AND, BOOLEAN LOGIC.

Feature: A bug as described by the marketing department.

Feedback: messages sent to the sysop of the board via the F)eedback command.

Flow control: a) A method used by brain-damaged(cf.) terminals to compensate for their lack of speed. When the terminal is sent more characters than it can handle within a short period of time, it sends a control-S to the host, telling the host to pause. When it is finished, it sends a control-Q telling the host to resume. b) User emulation of the same, for when characters are spit at the user faster than the user can comprehend them (esp. at 1200 baud!).

Forced mail: Mail sent to a user that he must read (or at least see, since it is impossible to abort the printing of such mail without hanging up). Can only be sent by sysops.

Copyright 1987 Scott & Eric Green

General File: Text file, usually of an informative nature, such as the advantages and disadvantages of one BBS program over another, a list of local BBS's, or a description of the Punter protocol.

GFILE: See General File.

hacker: a) A person who likes to program, creates useful, well-written programs at every opportunity, and is extremely proficient at it. b) an appellation attached by ignorant newspeople to computer vandals who break into other people's computers and destroy data. This meaning has been adapted by the ignorant public, resulting in a slight naming problem amongst the Real Hackers, who now have no name to describe themselves with. Note that a Real Hacker has utmost respect for the computers and data that he hacks, and would never consider undertaking computer vandalism because that would be a breach of the Hacker Ethic.

Hacker Ethic: Computers are a way of expression, and a path to beauty. Therefore, you will explore and create with computers, and in no way do anything to harm a computer.

IRQ -- interrupt. Every 1/60th of a second, a timer "interrupts" the program, and tells it to do something else (such as see if you are pressing a key on the keyboard). If you are doing so, it then does whatever action is necessary (put the key in the keyboard buffer, raise someone's access, etc.). After all this is through, the computer resumes running the program (this all happens much faster than a blink of an eye).

library: The up/download system, PFILE system, and GFILE system are divided into numerically-designated sub-sections called LIBRARIES.

line: a collection of characters, terminated by a RETURN character. In the ICE implementation, a line must contain fewer than 80 characters.

mail: see EMAIL.

message: a) A bulletin or response (cf.), e.g. "I put a message on B1 about the meeting". b) a personal message in mail, e.g. "I left you a message on the Mystic Stone (another BBS)."

new: a) something that you have not encountered before. b) in BBS parlance, a "new" message is one posted since the last time you logged in (that is, you obviously haven't encountered it yet).

Copyright 1987 Scott & Eric Green

news: a) information about NEW (cf.) things. b) the files maintained by the BBS sysop containing this information. c) the sub-system of the BBS responsible for manipulating these files and allowing you to read them.

overlay: A program which is loaded off of disk to perform a certain task, and then is replaced by another overlay to perform another task. In ICE, overlays co-exist with the main program, loading into an 8K area at the start of BASIC memory.

PFILE -- see Program File

post: to add a bulletin to a sub-board, or add a response to a bulletin.

posting: the text of a bulletin or response.

Program File: an overlay (BASIC program) which can be loaded and run by the BBS. Generally contains a sysop utility or simple game.

Responses: Text which is added to a bulletin, usually representing information and discussion from other people about the topic of the bulletin (although you'll always have a joker who'll respond about floppy disks when the bulletin is about politics).

SFD-1001: A one-megabyte disk drive, manufactured by Commodore for their failed B-128 business machine. Interfaces to the Pet (IEEE) bus, not to the serial bus.

sub-board: The bulletin section is divided into numerically-designated sub-sections called SUB-BOARDS.

subsysop: He's like a sysop, but he only has power on his own sub-board.

sysop: Short for system operator, the person who is running the BBS. Pronounce "sis-op", as rhymes with "sister". Is NOT pronounced "Seye-sop", as rhymes with "eye".

system: A collection of sub-systems, which together makes up the entire system. For example, in a bulletin board system, the subsystems consist of the individual pieces of hardware (disk drives, etc.) and the individual pieces of the software (the up/download sub-system, the bulletin board sub-system, the mail sub-system, etc.).

text: a) any collection of sentences. b) a collection of LINES, such as the editor maintains during editing, or the BBS prints to you when you are reading bulletins.

Copyright 1987 Scott & Eric Green

time-per-call: The amount of time that a person can stay connected to the system when he calls the BBS.

true: Numerically represented as a 1 in digital logic, or 5v at the hardware lefvel. One of the two fundamental values of bi-natured Boolean algebra (see BOOLEAN LOGIC).

UDs: see up/downloads

up/downloads: The sub-section of the BBS which allows you to send (upload) programs and recieve (download) programs to/from the BBS. Divided into numerically-designated LIBRARIES

upload: to send a program to the BBS (see up/downloads).

verify (user): Make sure that a person with the name given by the user lives at the telephone number given by the user.

word: a) a collection of BITS, the logical data size of the computer in question (e.g. 6502 -- 8 bits, 8086 -- 16 bits, 68020 -- 32 bits). b) Since 8 bits is useless, we'll call 16 bits a "word" on the 6502. Incidentally, integer variables take up 16 bits (which is why an integer is used to hold the person's access).

Copyright 1987 Scott & Eric Green

Appendix B: Menu listings.

```
2     "m.main"            seq
3     "m.boards"          seq
3     "m.mail"            seq
3     "m.editor"          seq
2     "m.user list"       seq
3     "m.udload"          seq
3     "m.udload-e"        seq
3     "m.maint"           seq
3     "m.files"           seq
```

Main commands:

```
---------------------------------------
[B] Bulletin section ![UL] User list
[UD] Up/downloads    ![OFF] Logoff
[T] Time/Date/Last   ![M] Mail section
[C] Yell for chat    ![H] Help section
[PW] Change password ![S] Your status
[F] Leave feedback   ![I] System info
[GF] General Files   ![NF] News files
[PF] Program Files   ![E] Edit term
[U] New user message !    Parameters
[LD] Change last call date
[MANX]  maintenance section
---------------------------------------
```

Bulletin commands:

```
---------------------------------------
[RN]Read new on board![+] board up
[RA] "    " all "   ![-] board down
[SN] Scan new message![Sx] Scan at #x
[AN] About new  "    ![Ax] About #x
[R] Read next message![Rx] Read #x
[P] Post a message   ![V] View Sysop
---------------------------------------
[#x]  Move to board #x EX: 2
[Kx] Kill file #x    ![Ex] Edit file #x
[Z]   Board Maintenance
[Ix]  ICE/unICE bulletin
---------------------------------------
[Q] Quit to main     ![D] Dump mode
[UD] Up/downloads    ![OFF] Logoff
[T] Time/Date/Last   ![M] Mail section
[C] Yell for chat    ![H] Help section
[F] Leave feedback   ![UL] User list
[PF] Program files   ![NF] News files
[GF] General files
[LD] Change last call date
---------------------------------------
```

Copyright 1987 Scott & Eric Green

Electronic mail commands:

[A] Read all messages	[x] Read #x EX:3
[S] Send a message	[D] Delete mail
[R]Reply to last read	[Rx] Reply to #x
[L] List messages	EX: R2
[Q] Quit to main	[UL] User list
[UD] Up/downloads	[OFF] Logoff
[T] Time/Date/Last	[F] Feedback
[C] Yell for chat	[H] Help section
[PF] Program files	[NF] News files
[GF] General files	[B] Board level
[LD] Change last call date	

Editor Commands:

[/S] Save	[/A] Abort
[/L] List	[/D] Delete
[/R] Read	[/C] Center
[/E] Edit	[/O] Toggle Line #'s
[/I] Insert Mode	[/N] Clear Text
[/X] Exit Insert/Edit	

Example of [/I]: /I1

General syntax for List/Read/Edit/ Delete/Center can be

x (line x only)
x,y (lines x through y)
x, (line x to end of editor)

EDIT and DELETE have a default of the last line of text.
LIST, READ, and CENTER have a default of all text entered.

Copyright 1987 Scott & Eric Green

User list commands:

```
[A]  List all users
[S]  Search for specific name
[CT] Search for computer type
[SA] Search for access
```

```
[Q] Quit to main      ![B] Board Level
[C] Yell for chat     ![M] Mail section
[UD] Up/downloads     ![OFF] Logoff
[T] Time/Date/Last    ![F] Feedback
[H] Help section      ![NF] News files
[PF] Program files
[GF] General files
[LD] Change last call date
```

Up/Download commands:

```
[L] List libraries    ![U] Upload file
[S] Scan titles       ![SN] Scan New
[SA] Scan new on all  ![Sx] Scan at #x
[A] About files       ![Ax] About at #x
[AN] About new files  ![V] View Sysop
[Z] Maintenance       ![Rx] Read file #x
```

```
[Dx] Download file number x EX: D5
[BS] Change block size.
[Kx] Kill file #x
[P]  Switch between Xmodem/Punter
[Ix] ICE/unICE a file
```

```
[Q] Quit to main      ![UL] User list
[C] Yell for chat     ![OFF] Logoff
[B] Board section     ![M] Mail section
[T] Time/Date/Last    ![H] Help section
[F] Leave Feedback    ![NF] News files
[PF] Program files
[GF] General files
[LD] Change last call date
```

Copyright 1987 Scott & Eric Green

```
    Exchange Up/Download commands
-----------------------------------------
[L] List libraries    ![U] Upload file
[D] Download a file   ![V] View Sysop
[$] Disk Directory    ![+] Library up
[BS] Change blk size  ![-] Library down
[MD] Multi-download   ![MU] Multi upload
[R]  Read a file
[@]  Sysop -- disk command.
[P]  Change U/D protocol
[Z]  Library maintenance
-----------------------------------------
[Q] Quit to main      ![UL] User list
[C] Yell for chat     ![OFF] Logoff
[B] Board section     ![M] Mail section
[T] Time/Date/Last    ![H] Help section
[F] Leave Feedback    ![NF] News files
[PF] Program files
[GF] General files
[LD] Change last call date
-----------------------------------------

       Maintenance commands:
-----------------------------------------
[L] Read system log   ![I] Edit id#
[F] Read feedback     ![*] print device
[R] Read a file       ![@] Dos command
[W] Write a file      ![$] Directory
[D] change device
-----------------------------------------
[FMW] Write forced mail
[FMD] Delete forced mail
-----------------------------------------
[Q] Quit to main      ![B] board level
[C] Yell for chat     ![UL] User list
[UD] Up/downloads     ![OFF] Logoff
[T] Time/Date/Last    ![M] Mail section
[PF] Program files    ![H] Help section
[GF] General files    ![NF] News files
[LD] Change last call date
-----------------------------------------
```

Copyright 1987 Scott & Eric Green

```
          Files commands:
---------------------------------------
[L]  List files in library
[x]  Read/run file #x. EX: 3
[J]  Jump to a P/Gfile library
[Jx] Jump to P/Gfile library #x. EX:J2
[E]  Edit list of libraries
[FM]  Files maintenance
---------------------------------------
[Q] Quit to main      ![B] Board level
[C] Yell for chat     ![UL] User list
[UD] Up/downloads     ![OFF] Logoff
[T] Time/Date/Last    ![M] Mail section
[F] Leave feedback    ![H] Help section
[PF] Program files    ![NF] News files
[GF] General files
[LD] Change last call date
---------------------------------------
```

Copyright 1987 Scott & Eric Green

INDEX:

$,command, 35,79
£&&x MCE command, 61
£?%$ MCE command, 60
£?%n MCE command, 60
£BLx MCE command, 61
£BSx MCE command, 61
£CFx MCE command, 61
£CLx MCE command, 61
£CPx MCE command, 61
£CR MCE command, 63
£CS MCE command, 61
£DOx MCE command, 62
£DT MCE command, 60
£EI MCE command, 63
£EL MCE command, 62
£FAx MCE command, 61
£GEx MCE command, 61
£IFx MCE command, 62
£IGx MCE command, 62
£JU MCE command, 63
£LDx MCE command, 61
£ORx MCE command, 61
£PP MCE command, 63
£PSx MCE command, 61
£PUx MCE command, 61
£SEx MCE command, 61
£TAx MCE command, 61
£↑↑x MCE command, 61
"bbs config", 24
*,command, 35
+,command, 64,73
-,command, 64,73
/a, 55
/c, 55
/d, 55
/e, 56
/i, 57
/l, 57
/n, 57
/o, 57
/r, 58
/s, 58
/w, 58
1541 4,12
1650, 4,20
1660, modem, 4, 5,20
1660, modification, 5
1670, 4,20
6420, 20
6470, 20
6480, 20
8050 4,12
8250 4
@,command, 35,44,79
A,command, 64,68,74,82
abort,file transfer, 41
aborting,text, 49
about,bulletin, 64
about,files, 74
access group, 10,12
access system, 10
access, 7,31
access,edit, 40
access,post, 13
access,sub-board, 12
ACCS, 12,15,40
add,u/d file, 75
algebra, 32
AN, 12
AN,command, 64,74
Anchor 6470/6480 4
Anchor Automaton, 20
Anchor Mark XII 4
Anchor Volksmodem 1200 4
anonymous,post, 12
Arabs, 9
area editor, 27
area setup, 27
AREA, 38
area,UD, 15,27,70
Area,worksheet, 17
arithmetic, 7
assembly language 3
Avatex 1200, 4,21
Avatex 1200HC, 21
B,command, 50
BASIC 3
BILLyuns, 7, 8
Binary math, 8
binary, 8
Bit, 9
BITS, 12,15
blank screen, 41
block size,Punter, 74
BNU, 41
board editor, 26
boot,system, 30
bootup file, 27
BS,command, 74
Buscard-][, 5
C,command, 50
C-Link][, 5
C-NET 3
C-Power 2
CALL, 39
calls per day, 31
calls per day,decrease, 39
calls per day,increase, 39

Copyright 1987 Scott & Eric Green

carrier,mask, 21
Carrier,mask, 22
CASM 2
CBterm, 70
center,text, 55
chat flag,toggle, 40
CHAT, 39
chat,command, 50
checksum, 70
CHT, 40
CHUG, 72
CL:, 38
clear,text, 57
commands,message, 63
Commodore, 5
commodore-1, 31
commodore-5, 31
commodore-a, 41,71,74,76
commodore-b, 41
commodore-control-d, 41
Commodore-Control-T, 38
commodore-f, 41,47
commodore-i, 30,47,40
commodore-l, 30,40
commodore-n, 41
commodore-o, 41
commodore-p, 41
commodore-q, 41
commodore-s, 40
commodore-t, 31,38
commodore-u, 41
Commodore-Z, 38
Commterm 3+, 70
Compu$erve, 70
computer type, 82
configuration,modem, 20
configuration,modem, 26
configure,bbs, 24
control-1, 31
control-5, 31
control-A, 56
control-D, 56
control-i, 49
control-P, 56
control-r, 49,56
control-s, 49
control-u, 49
control-V, 56
control-w, 49
control-x, 47,71,74,76
Converting, binary/decimal, 9
converting,1.0 to 1.1, 46
copying,system files, 28
Cosmos, 8
counting rules, 7
Countingville, 8
CRC Xmodem, 70
CT,command, 82
current device, 36
current drive, 35
D (command),68,31,36,65,74,80
date, 52
date,last, 51
decimal math, 8
decimal, 7
DELETE, 36,37
delete,line, 49
delete,mail, 68
delete,text, 55
delete,word, 49
device number, 12
digits, 8
directories,files, 83
directory, 79
directory,disk, 35
disk command, 35
disk command,u/d, 44,79
disk directory, 79
disk,gfiles, 23
disk,pfiles, 22
disk,pfiles, 22
disk,program, 22
disk,user file, 22
disks,bbs, 22
DN, 16
Donald Knuth 2
download credit ratio, 31
download credit ratio,
decrease, 39
increase, 39
download, 74
download,access, 16
DR, 12,15
drive number, 12
drive,current, 35
drive,default, 31
DTR, 20,22
dump mode,bulletin, 65
dump user, 41
DV, 12,15
E,command, 33,43,50,65,84
E-,library name, 16,70,79
Eagleterm, 70
Easyscript, 33
edit,bulletin, 43,65
edit,file, 31,37
edit,text, 56
edit,user, 37
editor, 54
editor,area, 27

Copyright 1987 Scott & Eric Green

editor,board, 26
editor,UD, 27
EMAIL, 51,68
Epyx Fastload 5
ERASE,forced mail, 36,37
erase,text, 57
example,access group, 12
example,board worksheet, 14
exchange mode, 79
exchange-style,libraries, 16
exit,bulletin reading, 65,66
F,command, 36,51
F1 (key), 31,39
F2 (key), 31,39
F3,key, 39
F4,key, 31,39
F5,key, 31,39
F6,key, 39
F7,key, 40
F8,key, 41
Fast Hack'em, 28
fastboot v2, 5,25,27
fastloader, 4, 5,25
feedback,command, 51
feedback,read, 36
file,read, 37
file,user, 25
file,write, 37
filenames,list, 75
files,list, 84
files,text, 34
files,writing, 31
FM(command), 32,33,34,84
FMD,command, 37
FMW,command, 36
force mail,write, 36
forced mail,delete, 37
FORTH 3
GF,command, 34
GF,command, 51
gfiles, 33
gfiles, 83
H,command, 51
Handic Superbox, 5
Handic, 20
Hayes, 20
help, 51
Hesmodem, 20
hexadecimal, 7
I,command, 37,43,51
ICE, 16
ICE,postings, 43
ICT, Hard drive 4
ID, 39
IDLE, 38
IEEE, interface, 5
immediate mode, 30,47
immediate mode,toggle, 40
infinity, 8
info,system, 51
Input/Output, 5
insert,text, 57
interface,RS-232, 20
J,command, 33,34,83,84
jump,directory, 83,84
justification, 63
K,command, 43
keys,sysop, 38
kill,bulletin, 43,65
kill,u/d file, 43
Kx,command, 65
L,command,37,38,65,68,75,80,84
last date, 51
LAST, 39
LD,command, 51
leave,system, 52
libraries, 15
libraries,list, 84
library, 70
library-style,libraries, 16
Limits, system, 6
line length, 50
line,delete, 49
line,redraw, 49
list,directories, 83
list,libraries, 75
list,mail, 68
list,sub-boards, 65
list,text, 57
list,users, 53,82
LOC, 40
local mode, 30
local mode,toggle, 40
log,printer, 41
log,system, 37
logoff, 52
M,command, 51
Mach 5 5
mail, 51,68
mail,delete, 68
mail,electronic, 68
mail,read, 68
maintenance,bulletin, 43
maintenance,command, 31
maintenance,file, 32
maintenance,files, 84
maintenance,section, 35
maintenance,u/d, 43
MANX(command), 31,32,35
Mark XII, 20

Copyright 1987 Scott & Eric Green

math,binary, 8
math,decimal, 8
mathematics, 7
MCE, 13
MD,command, 80
Merlin 2
message,new user, 52
Mike J. Henry 5
Mitey Mo, 20
ML 3
modem table, 26
modem, 4
move,to library, 73
MU,command, 80
multi-download, 79,80
multi-upload, 79,80
NA,command, 47
NAME, 12,16,34,38
name,sub-board, 12
new user message, 52
new users, 41
NEW, 30,41
new,files, 74,75
news files, 83
news, 31,51
next,library, 73
next,sub-board, 64
NF(command), 32,51
OFF, 41
OFF,command, 52
OFF,flag, 47
OFF,forced mail, 36,37
offhook, 21
Omnitronix, 20
onhook, 21
overlay, 22,25,32,52
P,command, 65,75,80
p.b-edit, 29
p.file copy, 45
p.term, 29,44
p.ud areas, 28
p.udload, 28
p.udload-e, 28
p.wait, 28
p.wait/1200, 28
p.wait/300, 28
p.wait/avatex, 28
p.weed, 29,44
parameters,terminal, 50
pause,text, 49
PF,command, 32,52
pfile, 22,32,52,83
Phoneman, 70
PLG 41
poke, 22
post,anonymous, 12
post,message, 65
PP&S, Hard drive 4
prefix,ID, 25
previous,library, 73
previous,sub-board, 64
printer log, 41
printer log,toggle, 41
printer,toggle, 41
problems,u/d, 77
professional, 5
program files, 83
Promal 3
protocol,change, 80
protocol,default, 75
protocol,up/download, 70
PRT, 41
PST, 13
Punter, 70,76
punter,protocol, 71
PW,command, 52
Q,command, 66
Quantum Link 1,72
quit,editor, 55
R,command, 37,65,80,68
RA,command, 66
re-draw,line, 49
read mail, 68
read,bulletin, 65
read,new, 66
read,text file, 80
read,text, 58
recieve,file, 74
reply, 68
RN,command, 66
Ron Meyer, 72
RS-232,interface, 20
rules,counting, 7
S,command, 52,66,69,75,82
s.end, 31
s.new user, 31
s.start, 31
s.sys info, 31
s.welcome, 31,46
SA,command, 46,75
Saganary, 7, 8
save,text, 58
scan,titles, 66,75
screen,blank, 41
screen,sysop, 38
search,user access, 46
search,user, 82
send,mail, 69
setup,area, 27
setup,system, 24

Copyright 1987 Scott & Eric Green

setup,UD, 27
SFD, 12
SFD-1001 4
Sizing Xmodem, 70
Skyles IEEE-Flash, 5
SN,command, 66,75
Spinnaker 5
Star NX10 2
status, 52
Steve Punter 2
sub-board subsysop, 13
SUBSYSOP, 13,15,16
subsysop, 67,76
subsysop,library, 15
subsysop,sub-board, 13
syntax,editor, 54
SYS, 40
sysop flag,toggle, 40
Sysop, 12
sysop,sub-board, 76
system info, 51
T,command, 52
tab, 49
table, 2x, 9
table,modem, 21,26
TC:, 38
Telenet, 76
term.c1, 71
terminal parameters, 50
terminal, 30
TeX 2
text files, 34,51
time per call, 31
time per call,decrease, 39
time per call,increase, 39
TIME, 38
time, 52
time,decrease, 38
time,increase, 38
time,zero, 38
title,bulletin, 64
titles,files, 74
titles,scan, 66
Total Telecommunications, 20
transmit,u/d file, 75
trinary, 7, 8
Turbo Load/Save 5
Tymnet, 76
U,command, 52,75,80
u/d log,toggle, 41
u/d maintenance, 43
u/d problems, 77
U/D, 41
u/d,worksheet, 18
UD editor, 27
UD setup, 27
UD,command, 52,70
UDL, 41
UL,command, 53
un-ICE, 16
UP, 16
up/download, 15
upload, 75
upload,access, 16
USENET 1
user file, 27
user groups, 10
user list, 53
V,command, 67,76
VIP-Term, 70
Volksmodem 1200, 20
W (command), 31
W,command, 37
wallchart,sysop commands, 42
Ward Christenson, 72
Westridge, 20
Wordpro-128 2
wordwrap, 58
Worksheet, access group, 12
Worksheet, board, 12
worksheet,area, 15,17
worksheet,example boards, 14
worksheet,u/d, 18
Xetec, hard drive, 4
Xetec, Supergraphics, 2
Xmodem, 47,70,72
Z, 30
Z,command, 43,44

Copyright 1987 Scott & Eric Green

ACCESS GROUP WORKSHEET

BIT#	GROUP NAME
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

BOARD WORKSHEET

##	NAME	BITS	ACCS	DV	DR	AN	MCE	PST	SUBSYSOP

UD AREA WORKSHEET

##	NAME	BITS	ACCS	DV	DR	SUBSYSOP

UD WORKSHEET

##	NAME	BITS	ACCS	DV	DR	UP	DN	ICE	SUBSYSOP

ICE BBS Registration

We maintain a mailing list of all ICE BBS sysops. We are also in the process of gathering a complete list of the phone number, name, and sysop of all currently running ICE boards. Please answer the following questions, and mail back this form. All information will remain confidential.

Name: _____

Organization:_____

Address:_____

City: _____

State: _____ Zip:_____

Area code:_____ Voice number :_____

BBS name :_____

B.B.S. number:_____

BBS sysop (handle): _____

The following information is in order for us to better judge what features to put in future versions of ICE:

Disk storage:

Hard drive (y/n, size,type): _____

_____

Floppy disk drives (size,type):

a) _____
b) _____
c) _____
d) _____

IEEE interface (if applicable): _____

Modem type and speed: _____

Do you wish to be listed on our official ICE BBS list? Listed information: BBS name, BBS sysop (handle), baud rate, disk storage, BBS telephone number.

BBS list (y/n):_____

Hours: _____