TBBS

The Bread Board System

Version 1.3

User Manual

ISBN 0-928402-00-2 {IBM PC}
ISBN 0-928402-01-0 {CP/M 2.2}
ISBN 0-928402-02-9 {TRS-80}

Copyright (c) 1984 by Philip L. Becker

TBBS is manufactured by eSoft, Inc. We have made every effort to provide the user with error-free data diskettes. Should there be difficulty in transferring the data from these diskettes to your operating system diskette, contact us at (303) 699-6565 (voice).

TBBS is distributed on an "AS IS" basis only, without warranty. Neither eSoft, Inc. nor its authorized dealers of TBBS shall have liability or responsibility to any person or entity with respect to liability, loss, or damage caused or alleged to be caused by this software. This includes, but is not limited to, any interruption of service, loss of business or anticipatory profits, or consequential damage resulting from the use of this software.

eSoft SOFTWARE SUPPORT

eSoft, Inc., provides a 24-hour-a-day data support service to registered TBBS system owners at (303) 699-8222 (eSoft Software Support BBS -- 300/1200 baud modem). We strongly urge that this telecommunications system be used for error reporting and troubleshooting assistance. Because all system corrections will be distributed first via the eSoft Software Support BBS, we recommend that all registered TBBS system operators log on to the Support BBS at least once a month to receive latest system update information. (At your first logon, you will need to register the serial number of your TBBS master disk(s) and supply a password, which will be approved for full support access during the next working day. Other information is given online during the registration process.)

The Bread Board System software is distributed without copy-prevention protection, since we feel that such protection drastically reduces the utility of a program and punishes most the honest purchaser. We do, however, expect purchasers of TBBS to honor their registration agreement. This software is copyrighted, and sold to the purchaser for your individual and exclusive use. Any reproduction for use by other persons is a violation of our copyright.

WE URGE THAT FREQUENT BACKUP COPIES OF
TBBS FILES BE MADE TO PREVENT LOSS OF DATA.

Table of Contents

CHAPTER 1: Introductions

How To Use This Manual.....1-1
The Bread Board System Programs and Files.....1-2

CHAPTER 2: System Overview

Construction of TBBS.....2-1
Description of Sample System.....2-2

CHAPTER 3: Using CEDIT

The Functions of CEDIT.....3-1
A Review of CEDIT Operations.....3-3
Setting Up Your TBBS Options.....3-7
Defining Your Message Boards (BBS's).....3-19
Defining User Terminal Characteristics (defaults).....3-22

CHAPTER 4: Using MEDIT

How Command Menus Are Constructed.....4-1
Reviewing MENU0000.CTL.....4-3
MEDIT Operations.....4-6
TBBS Menu Control Command Types.....4-10
Menu Branching File Commands.....4-19
Message Board Commands.....4-21
Single File Display Commands.....4-29
Selection List File Display Commands.....4-31
Partitioned File Display Commands.....4-34
Upload File Commands.....4-36
General System Function Commands.....4-39

CHAPTER 5: Using QAEDIT

How Question and Answer (.QAF) Files Function.....5-1
How a Survey File is Constructed.....5-2
How a Voting File is Constructed.....5-4
How to use VOTESUM.....5-5
How to use QAEDIT.....5-6

CHAPTER 6: Advanced Techniques

Remote SYSOPing.....6-1
Advanced Menu Formatting.....6-4
Advanced QAEDIT file uses.....6-11
Disk Management for efficiency.....6-18

CHAPTER 7: Using MFSQZ

How to use the Message File Editor.....7-1

Table of Contents

CHAPTER 8: Using ULEDIT

How to use the User Log Editor.....8-1

CHAPTER 9: Troubleshooting

System Error Messages.....9-1
Troubleshooting Chart.....9-2
Bug Reporting Procedures.....9-4

CHAPTER 10: System Specifications

Data on RS232 Standards.....10-1
TBBS RS232 Driver.....10-4
How TBBS uses a modem.....10-7
CONFIG.CTL File Format.....10-10
MENU????.CTL File Format.....10-15
USERLOG.BBS File Format.....10-18
MSGHDR.BBS and MSG.BBS File Format.....10-19
????????.QAF file Format.....10-20
Word Wrap Algorithm.....10-21

CHAPTER 11: System Startup

Getting Started FAST!.....11-1
Updating From TBBS Version 1.2 to Version 1.3.....11-2

INDEX.....Index-1

Preface

Welcome to the TBBS family. As you experience the power and flexibility of the most advanced personal computer communications system available, I am sure you will be glad you purchased it. Version 1.3 is the culmination of over 100 man years of testing and use evaluation. It is my hope that it will provide the vehicle for the creative people in the microcomputing community to build new methods of information exchange without being bogged down by learning programming. I also hope this system will allow an entire new class of business and personal communications tasks to be performed which were not feasible before.

TBBS is distributed with no copy protection. I feel personally that copy protection only punishes those who pay for a product. However, I point out that TBBS is a copyrighted product and that pirating it is illegal. We have pursued legal action in instances of wide spread piracy in the past and will continue to do so in the future. Since TBBS serves a limited market (compared to say, a video game), the earnings potential is severely impacted by piracy. If you wish to see TBBS stay alive and growing, it is in your own best interests not to give away what you have bought. You program broadcasts its serial number easily and may be tracked if you do pass it on. Enough said.

As the author of TBBS I would like you to know a little about me. My involvement in computer based communications extends back over 10 years. I have designed and implemented communications front ends for large mainframe computer systems and written operating systems for time sharing and real time applications over that period on mainframes, minis, and micros. One of my favorite feelings comes from making micros do jobs no one believes possible. My first microprocessor experience came in 1976 when I built a communications processor using microprocessors which cost 1/10th as much as the then available equipment and was much more flexible. I have been a proponent and active user of microprocessors ever since.

In early 1982 I called my first micro-based Bulletin Board. I was amazed by what I saw. As has become the norm, the most innovative, "people oriented" usage of computers was coming up from the bottom. I felt, however, that while the systems I saw showed tremendous human design and were very innovative, none had a professional integrated systems design or approach. I therefore was prompted to write TBBS. User reaction to versions 1.1 and 1.2 has showed that it fills a need and I hope version 1.3 will again push the state of the art forward another notch. In addition, Version 1.3 brings the power of TBBS to the IBM PC and MS-DOS compatibles, and also the world of CP/M 2.2. So now,

almost anyone who wishes to can use TBBS.

As a final note please read your manual thoroughly. The key to using TBBS to its fullest is to understand the system structure and philosophy. I guarantee that you have never seen a system like this in the micro world before. It is a revolutionary advance in personal communications and cannot be learned through osmosis. We will provide the support you need to get your system up and keep it running, but please do us a favor and read the directions! The introduction will get you oriented to the TBBS program structure, and Chapter 2 will explain the sample system provided in detail. When you feel you understand that system, read the advanced techniques in Chapter 9. This will give you some feeling for the truly awesome power which TBBS Version 1.3 provides to you.

Oh yes, be sure to send in your registration so you will have access to the SYSOP support section of the TBBS HQ board. Those who have used it say it provides the very best and most timely support they have ever had with any software product. Also, if you register you will receive the TBBS newsletter which has features and operating tips you'll not want to miss.

I think you will enjoy being part of the TBBS owner's community. It is a friendly, fun, place to enjoy pushing the state of the art in computer communications forward. It is populated with really nice fun folks. I hope to be typing to you soon.

Philip L. Becker September 18, 1983

- THE BREAD BOARD SYSTEM -
Version 1.3
(c) 1984 Philip L. Becker

HOW TO USE THIS MANUAL

This manual is organized into four distinct sections. These sections are:

- o The introduction
- o Building Your Custom TBBS system
- o Maintaining Your TBBS system
- o Technical Information

The section on Building your TBBS system includes all instructions necessary to learn how to install and maintain a TBBS telecommunications system in a custom configuration. This includes the "how to" instructions on the use of the programs CEDIT, MEDIT, and QAEDIT. It is both a tutorial and reference manual on the TBBS control file system. This section should be studied in levels as you work "hands on" with the sample system provided. First, try to gain an understanding of the fundamental processes involved. Then, learn the advanced features and capabilities level by level to release the underlying power of TBBS. You will then be able to construct a custom system to fill your communications needs.

The Maintenance section discusses the utility programs MFSQZ and ULEDIT which allow you to handle the message base and userlog functions required to manage a running system. This section should be studied after you have built or installed at least the minimum sample system supplied on the release disk and have it running.

The Technical specifications of interest to programmers or to those who may be modifying TBBS for use in a non-typical configuration are included in the Specifications section. Included is a bit map and sector layout of all data structures which you might wish to interface with if you write specialized utility programs etc.

We suggest you first finish this Introduction to TBBS, for an explanation of the various files and programs on your master system disks and an overview of how they all fit together. Then, if you are interested in getting your TBBS online fast, turn to Chapter 11, "Getting Started FAST". It's at the back of the manual, because -- once you have used it -- it would simply be in your way at the front of the Manual. Chapter 11 is intended only to get you started, so that you may practice and develop

experience with the sample minimal TBBS system.

Once you have completed the instructions in Chapter 11, you should start at the front of the Operations portion (Chapter 2, System Overview), studying all of the Operations chapters. Your later use or study of the Specifications chapters is at your discretion -- but we wanted to be sure the technical information is available if you need or want it.

We expect that you're anxious to tell the whole world about your TBBS...but we strongly urge you to use only a friend or two at home terminals as your initial users, until you become more familiar with the hundreds of various system options open to you, and how to use them.

Give this process time...a very busy weekend, at least. "Getting Started FAST" can get you operating in an evening -- but learning to use the full system will take dozens of hours of study, observation, careful planning and work. Rushing that process is likely to frustrate both you and your system users.

Every TBBS system now on-line is unique, and you're about to create your own unique system. Besides work, it's also fun -- so, enjoy it!

What are the parts of TBBS?

The TBBS distribution disk(s) contain several files. In this section we will try to give you an overview of what those files are used for. In addition, we will try to paint the "big picture" of TBBS, so that you will see where each piece fits as we discuss it in detail in the following chapters.

The distribution files may be grouped by their file types or extension names as follows:

* .COM - The actual TBBS and utility programs.
  .CTL - Configuration or Menu Control Files which define your system.
  .TXT - ASCII text file for display or program stored for file transfer.
  .QAF - Question and answer file for survey or voting use.
  .SVY - File containing answers from .QAF survey function.

* Note: On TRS-80 systems, all files have the . replaced with a / and in addition, the .COM files will be named /CMD.

We will now look at the members of each group in detail. First the .COM group.

The two main programs which comprise the TBBS run time system are:

```
      TBBS.COM     - main run time program
  *   TBBSDVR.COM  - the communications handler and remote host
                     program.
```

* Note: In the CP/M version, TBBSDVR is embedded within the TBBS.COM file and is not present as a separate file. However it is still a functionally separate program from TBBS.

Construction Editors

TBBS gains its flexibility and power from the fact that TBBS.COM does not have any hard coded structure as such. It takes its structure from the .CTL file group which defines the way your specific system looks. The .CTL files are discussed briefly below, and in detail in the CEDIT and MEDIT sections.

The remaining .COM files comprise the TBBS utility programs. These programs fall into two categories. Category one is the group of utility programs which you use to construct your system or to modify its appearance and operation. This group consists of:

```
CEDIT.COM   -  Creates/edits the Configuration File CONFIG.CTL
MEDIT.COM   -  Creates/edits TBBS Menu Control Files MENU????.CTL
QAEDIT.COM  -  Creates/edits Q & A files ????????.QAF
```

These three utility programs allow you to build your TBBS system. CEDIT and MEDIT define the overall system configuration and menu structure, while QAEDIT, another editor program unique to TBBS, allows the SYSOP to create special files for conducting system user surveys, ordering merchandise, accepting subscriptions, or conducting secret multiple-choice quizzes similar to elections ... for which the program VOTESUM.COM is used to tally results.

Maintenance Editors

The second category of utility programs are those which are used to maintain the TBBS system files during day to day usage of the system. These programs are:

```
MFSQZ.COM   - Maintains the TBBS message base. Allows editing,
              archiving, moving and rolloff of messages.
```

ULEDIT.COM - Maintains the registered User file. Allows viewing and editing of usage data, configuration data, and privilege and authorization for each user of your system.

VOTESUM.COM - A utility which will produce a summary report of a voting file which was created by QAEDIT as explained above.

Note: All TBBS editor programs are fully menu-driven, and can be used by the SYSOP either from the host TBBS terminal or (if online with the host system) remotely via telephone from any telecommunications terminal. In fact, using these programs remotely, you can re-configure your system, re-load TBBS and be online with a completely revised system without ever touching the local keyboard of your computer. Just another small example of the tremendous power of your TBBS system.

System Control Files

The next group of files are those ending with .CTL. These files are not ever actually used directly by you the SYSOP, but are the binary control files produced by CEDIT and MEDIT. These files in turn are used by TBBS to define the system structure of your TBBS. It is this control file structure which gives TBBS its tremendous power. There must always be one file named:

CONFIG.CTL - The Configuration file produced by CEDIT.

This file is loaded by TBBS when it first runs and contains all of the information on how many system options are to run in your system. It also contains the definitions for your message boards (up to 25 in all) and your pre-defined terminal types. This information becomes a part of TBBS for this run.

The remaining .CTL files will all have the name form of MENU????.CTL. In addition there must always be one file named MENU0000.CTL. This file is the top level menu of TBBS. It is the starting point. This menu may call out other menus to any level of complexity desired. Unlimited GOTO capacity and GOSUBS from any point up to 20 levels deep. These files are created and edited by the MEDIT.COM utility program. In the sample system the following MENU????.CTL files are supplied:

MENU0000.CTL - Sample Top level menu
MENUTERM.CTL - Sample Logoff or Goodbye menu
MENUREMS.CTL - Sample Remote SYSOP features menu
MENUMAIL.CTL - Sample EMAIL section menu
MENUTSIG.CTL - Sample Special Interest Group Section Top Menu
MENUSTRS.CTL - Sample TRS-80 SIG Menu
MENUSAPL.CTL - Sample Apple SIG Menu
MENUSATA.CTL - Sample Atari SIG Menu
MENUSIBM.CTL - Sample IBM SIG Menu
MENUSIGS.CTL - Sample Combined SIGS Menu

There is only one Question and answer file (created and edited using QAEDIT) in the sample system. It is:

ORDER.QAF - Question and answer file for <O>rder command. The <O>rder command itself is in MENU0000.CTL.

There are a series of text files which are used in the sample system. These are used in the "display selection from list" method of file presentation which is only one of several file display options available in TBBS. The others are detailed in the MEDIT reference section. These files are:

INFOM.TXT - Selection list for the <I>info command. The <I>info command itself is in MENU0000.CTL
INFO1.TXT - Text for Info selection 1: Hardware description
INFO2.TXT - Text for Info selection 2: Software description
INFO3.TXT - Text for Info selection 3: TBBS protocol & help
INFO4.TXT - Text for Info selection 4: TBBS detail sheet

BULLM.TXT - Selection List for the <B>ulletin command. The command itself is in MENU0000.CTL
BULL1.TXT - Text for Bulletin selection 1
BULL2.TXT - Text for Bulletin selection 2

DOWNM.TXT - Selection list for <D>ownload section. The actual <D>ownload command is in MENU0000.CTL. This file is a dummy, since no downloads are provided.

PRODM.TXT - Selection list for <P>roduct section. This is also a dummy since no product list is provided.

In order to get your sample system on the air, the following empty or minimum size system files are provided:

USERLOG.BBS - File of user name and parameter information. Supplied with SYSOP entry only. Note: the SYSOP record must always be the first entry and should never be deleted.

MSG.BBS - File of all message text, in sequential order

MSGHDR.BBS - Message header information for all messages. The two message files form the TBBS message base and are supplied with an introductory message. Note: the TBBS message base MUST always have at least one message in it. That message may be deleted, but you may not squeeze these files down to 0 messages or TBBS will malfunction. If this ever happens, just restore these release disk files.

All files above with the extension .TXT are ASCII text files. They were created with a text editor and saved without any word processing control codes. All files will be found by TBBS on any drive. Under MS-DOS, all files must be in the current logged on directory path of any drive unless they are .TXT type and the path name was explicitly specified in MEDIT. Then the specified path will be searched on all drives. Under CP/M they may also be in any user area. However, groups of files such as the .CTL group or a selection list and all of its selections in a .TXT group must be on the same drive and user area. There are no placement restrictions when operating under one of the three supported TRS-80 operating systems.

Thus we have seen that the Bread Board System is a fast, flexible telecommunications host system which requires no computer programming knowledge to install or maintain. It offers the functions of 43 separate menu commands, and may be used to simultaneously support up to 25 separate or combined, public or private, message services (BBS's).

Through an easy-to-manage system of privilege levels and user flags, it can recognize hundreds of variations in user access authorization, with or without password control.

It will function at either 300 or 1200 baud, and it can recognize the rate at which it is being accessed. It provides for transfer of either text or programmatic files via a direct "dump" from or to disk, or with the transfer verification protocols of Ward Christensen's XMODEM program. Version 1.3 will handle either the more common Checksum or the more secure CRC implementations of XMODEM protocol.

TBBS Version 1.3 can be operated on most MS-DOS 2.x computers, most Z-80 based CP/M 2.2 computers and any of three TRS-80 Model I/III/4 disk operating systems -- NEWDOS/80 Version 2.0/2.5, LDOS Version 5.1.4, and DOSPLUS Version 3.5.

We have provided several minimal control files which form a sample system to help you get your TBBS online and operating, so that you can learn how to use the system editors and develop experience with the full system. These are not "permanent" -- that is, you may edit and revise your copies of them any way you prefer. (You don't want to edit your master copies, of course, so that they remain available for reference.) The system we have provided does not even begin to tap the tremendous power or use any of the large number of advanced features available in TBBS.

Learning how to operate a TBBS is as much a matter of experience as instruction, so we want to get you up and running as fast as possible. You will continue to learn new ways to use your TBBS system for many months to come, and improve your system as you do so. From this introduction, you should proceed to Chapter 11, "Getting Started FAST" so that we can help you put the sample system online.

Later, you will need to return to the front of this manual, and study all the chapters carefully, in succession. Start with Chapter 2, AN OVERVIEW OF SYSTEM CONSTRUCTION. Greater stress is placed on "beginner"-level information in early chapters, so -- please -- start your study at the beginning.

AN OVERVIEW OF SYSTEM CONSTRUCTION

Your TBBS system is constructed using the three editor programs CEDIT, MEDIT, and QAEDIT. Each of these programs is responsible for constructing a specific portion of the overall system as follows:

CEDIT is responsible for editing the following system configuration items:

- o Describing how the user logon procedure will look and operate including pre-definition of commonly used terminal types and new user default settings.
- o Definitions of message boards
- o Setting modem handling protocol, and hardware interfacing.
- o Describing what information should be logged in a Dayfile (if any).

MEDIT is responsible for editing Menu Control Files which define the operation and appearance of your system. TBBS is a menu driven system and MEDIT creates and edits the menus which control the user's interaction with your system. The Menu Control File is the central "programming" structure of TBBS. It is here that you make commands active and structure the "look and feel" of your system as you wish. This makes MEDIT the most important tool used in constructing your TBBS system. You may even specify your own program to be remote hosted if you wish as a menu item.

QAEDIT is responsible for building and editing Survey and Voting files. One of the more powerful features of TBBS is its ability to allow you to take surveys or conduct secret ballot voting with great ease. MEDIT has a command type which may be placed in any menu to call out a QAEDIT created file. This file then controls the Survey or Voting function. QAEDIT allows many such surveys or votes to be easily created and modified almost as quickly as you can think up the function you wish performed.

To give you a better idea of how these three editors do their jobs, we will now analyze the sample system which was included on your master disk(s). During this analysis, we will avoid detailed descriptions of editor operation, as these will be explained in the following chapters. It is our intent here to acquaint you with the overall constructs used in TBBS, and to help you become familiar with the "big picture". We will also introduce some methods of planning and mapping your system which will prove helpful as you build your custom TBBS.

Configuring Logon Operation

We begin with a look at the portions of CEDIT which configure TBBS operation during logon. Logon begins when TBBS answers the telephone and detects a carrier. The first thing it does is to send the Greeting Text defined in CEDIT. For our sample system, this is:

Welcome to TBBS - The Bread Board System

(c) 1984 by Philip L. Becker

Why? Because the Greeting Text is set to this string in CEDIT. If you change this string, the header will change. After printing this text, TBBS will print the file SIGNON1.TXT (if you have indicated that it should in the LOGON OPTIONS section of CEDIT). In the sample configuration, we indicate that this file should be skipped.

Now that TBBS has printed the introduction, it is time to find out who is calling. There are 3 methods of doing this, and the sample system uses method 3. This means that it will ask the caller:

First Name?

And attempt to acquire the First and Last name of the caller. When it does so, it will look him up in the userlog. There can be two outcomes to this search.

1. If it finds him, it will print his location and ask for verification. If the user says yes to the verification question, he will be given three tries to supply his password. If he correctly does so he will be logged on.

2. If it does not find him, it will check to see if new users are allowed. In the sample system they are, so the caller will be asked for his calling location. Then he will be asked for a password for future logons. Lastly he will be asked his terminal width and be presented with a list of the CEDIT defined terminals as an aid to configuration. After he is configured, he will be logged on.

All of this behavior is due to the selection of Logon Method 3 in the CEDIT logon options section.

The sample system CEDIT configuration also indicates the password entry is to be security echoed, and that no duplicate names are allowed in the log. We also indicated the privilege and authorization a new user will be assigned. The terminal help

menu for configuration is built from the information supplied in the terminal definitions section of CEDIT.

We can see then, that much of the CEDIT information relates to the logon process. Other logon option settings in the sample CEDIT configuration indicate that the news file SIGNON2.TXT will be seen only one time by users, and will not be seen again until the "C" option has been used in ULEDIT to reset their news file flags. TBBS allows a great deal of flexibility in the operation of the logon process, and the sample system gives only one option. There are 2 other methods, and several variations on each method available.

Mapping the Menu Structure

While the menu structure of the sample system is quite simple by TBBS standards, it can still cause confusion if it is not mapped out. We will present one method of mapping your system here. You may prefer another method of representing the paths through your menu structure, but you must have some mapping scheme when planning your system, or you will find yourself with bad paths and missing links.

The mapping method presented here shows each menu by name, with all of its entry points on the left, and all of its exit points on the right. The entry points are assumed to be GOSUB type unless marked GOTO, and the exit points are always marked as to type. This mapping method has the advantage of remaining uncluttered when you design very complex menu structures. Its major disadvantage is that since the lines are not connected up, it does not give a very good visual feeling for the system's structure. You should choose your mapping method to fit your needs, and so it keeps you from becoming lost in your own system.

The sample menu set map is as follows:

```
            +---+
            :   :<M>(GOSUB)
            : 0 :---------->MAIL
            : 0 :<S>(GOSUB)
            : 0 :---------->TSIG
            : 0 :<G>(GOSUB)
            :   :---------->TERM
            :   :<E>(GOSUB)
            :   :---------->REMS
            +---+

            +---+
      0000  :   :<E>Return Up 1
---------->: M :----------> LVL
(GOTO)TSIG : A :<G>(GOSUB)
---------->: I :---------->TERM
            : L :<M>(GOTO)
            :   :---------->TSIG
            +---+

            +---+
     0000   :   :<A>Return Up 1
---------->:   :----------> LVL
     MAIL   :   :<N>Logoff
---------->:   :---------->
     TSIG   : T :
---------->: E :
     STRS   : R :
---------->: M :
     SAPL   :   :
---------->:   :
     SATA   :   :
---------->:   :
     SIBM   :   :
---------->:   :
     SIGS   :   :
---------->:   :
     REMS   :   :
---------->:   :
            +---+

            +---+
      0000  :   :<E>Return Up 1
---------->: R :----------> LVL
            : E :<D>Exit to DOS
            : M :---------->
            : S :
            +---+
```

```
                0000 +---+
            ------->:   :<E>Return Up 1
(GOTO)MAIL : T :----------> LVL
---------->: S :<1>(GOSUB)
            : I :---------->STRS
            : G :<2>(GOSUB)
            :   :---------->SAPL
            :   :<3>(GOSUB)
            :   :---------->SATA
            :   :<4>(GOSUB)
            :   :---------->SIBM
            :   :<C>(GOSUB
            :   :---------->SIGS
            :   :<M> (GOTO)
            :   :---------->MAIL
            :   :<G>(GOSUB)
            :   :---------->TERM
            +---+

                TSIG +---+
            ------->:   :<E>Return Up 2
            : S :----------> LVL
            : I :<G>(GOSUB)
            : G :---------->TERM
            : S :
            :   :
            +---+

                TSIG +---+
            ------->:   :<O>Return Up 1
            : S :----------> LVL
            : I :<E>Return Up 2
            : B :----------> LVL
            : M :<G>(GOSUB)
            :   :---------->TERM
            +---+
```

NOTE: STRS, SAPL, and SATA have identical flow to SIBM.

From this map we can see the following main points:

1. The menu TERM is treated as a general subroutine. It is called from each of the other menus using a GOSUB type command, and it either logs the user off the system, or returns to the menu which called it.

2. The top level menu (0000) only directly calls four of the other menus. Those are MAIL, TSIG, TERM and REMS.

3. The MAIL and TSIG menus in turn have a GOTO path between them at this level, 1 level below MENU0000, to allow back and forth access.

4. The menus STRS, SAPL, SATA, and SIBM form the four special interest group menus one level below the menu TSIG. They each have two returns -- up one level to the menu TSIG which called them, and up two levels to return to 0000 at the top.

5. The menu SIGS has only a return up 2 levels to the top, since it is a combined access to all SIG message areas.

6. Each of the individual SIG menus (STRS, SAPL, SATA, and SIBM) is identical except for the message board it calls out in the Opt Data field of the four message board commands. These correspond to the message board definitions in CEDIT.

How does it work?

We will now examine the Top Level Menu in detail, to see why it behaves as it does. You will need to refer to chapter 3 on MEDIT for some of the in depth details on individual menu command types. First a listing of MENU0000.CTL as it is given by the MEDIT list menu to printer option:

Listing of File MENU0000.CTL

```
TBBS Top Level Menu
-------------------

Entry  1: <R>ead Msgs on the Bulletin Board
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   R  Type=  6   Opt Data=BULLETIN BOARD
```

```
Entry  2: <Q>uickscan Msgs on Bulletin Board
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   Q  Type=  8   Opt Data=BULLETIN BOARD

Entry  3: <L>eave a Msg on the Bulletin Board
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   L  Type=  7   Opt Data=BULLETIN BOARD

Entry  4: <K>ill a Msg on the Bulletin Board
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   K  Type=  9   Opt Data=BULLETIN BOARD

Entry  5: <M>ail ... Electronic Mail Section
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   M  Type=  5   Opt Data=MAIL

Entry  6: <S>IGS ... Special Interest Groups
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   S  Type=  5   Opt Data=TSIG

Entry  7: <T>ime ... Elapsed Time
        Priv=   0  Auth 1= --------  Auth 2= --------
         Key=   T  Type= 11   Opt Data=**none**

Entry  8: <G>oodbye ... Terminate Session
        Priv=   0  Auth 1= --------  Auth 2= --------
         Key=   G  Type=  5   Opt Data=TERM

Entry  9: <I>nfo ... System Information
        Priv=   0  Auth 1= --------  Auth 2= --------
         Key=   I  Type=  1   Opt Data=INFO

Entry 10: <B>ulletins
        Priv=   0  Auth 1= --------  Auth 2= --------
         Key=   B  Type=  1   Opt Data=BULL

Entry 11: <D>ownload Programs
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   D  Type= 36   Opt Data=DOWN

Entry 12: <#>Upload ... Submit Programs
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   #  Type=27    Opt Data=**none**

Entry 13: <U>serlog ... List of callers
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   U  Type=22    Opt Data=**none**
```

```
Entry 14: <F>crmat ... Reconfigure terminal
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   F  Type= 31   Opt Data=**none**

Entry 15: <X>pert ... Change user level
        Priv=   0  Auth 1= --------  Auth 2= --------
         Key=   X  Type= 26   Opt Data=**none**

Entry 16: <C>hat ... Talk to SYSOP
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   C  Type= 23   Opt Data=**none**

Entry 17: <A>gain ... Re-logon to system
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   A  Type= 43   Opt Data=**none**

Entry 18: <P>roducts ... List of Merchandise
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   P  Type=  1   Opt Data=PROD

Entry 19: <O>rder ... Place Merchandise Order
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   O  Type= 32   Opt Data=ORDER

Entry 20: <%>Readall ... For SYSOP, reads all Boards
        Priv=  25  Auth 1= --------  Auth 2= -------X
         Key=   %  Type=  6   Opt Data=**none**

Entry 21: <E>ditors ... Remote SYSOP section
        Priv=  50  Auth 1= -------X  Auth 2= --------
         Key=   E  Type=  5   Opt Data=REMS
```

Now let's examine some of the reasons this menu works as it does. First, Entries 7, 8, 9, 10, and 15 are available to any user since they have a privilege of 0, and no A1 or A2 flags required. This means that if a caller were given a privilege of 0 in his userlog entry, his top menu level would be:

```
TBBS Top Level Menu
-------------------

<T>ime ... Elapsed Time
<G>oodbye ... Terminate Session
<I>nfo ... System Information
<B>ulletins
<X>pert ... Change user level

Command?
```

So he would be very restricted in his use of the system. He could only read the INFO and BULL files and check his time and set his menu prompting level, or log off.

The next layer of features becomes available when a user is given a privilege of 25 or greater. Then all menu entries except 20 and 21 will be available.

Setting A2 flag 8 for a user will allow him the readall command, but that command will not really operate as intended until the user's privilege is set to 255. This highest setting of privilege overrides the message board definition and makes all boards combinable during a call. Thus a combined board read as issued by Entry 20 will allow all messages from all boards to be read with a single command.

To obtain access to Entry 21 and do remote SYSOP activities, a user's userlog entry must have his A1 flag 8 set, and a privilege level of 50 or higher.

We make no pretense that the settings in the sample system are optimum or even all that sensible. This system is just intended to give you an operating TBBS and allow you to learn how it works. It is intended that you will build your own system which will exactly fit your information and communications needs.

Now that we have seen how the privilege and Auth 1/Auth 2 flag settings in the sample MENU0000.CTL file operate, let's look at some of the actual commands:

The first group is comprised of Entries 1, 2, 3, and 4. These four entries provide access to the single message board which was defined in CEDIT with the name BULLETIN BOARD. They provide the read, scan, write, and delete functions for this message board.

The next group is comprised of entries 5 and 6. These entries allow you to leave this menu and do a GOSUB to either MENUMAIL.CTL (Entry 5) or MENUTSIG.CTL (Entry 6). In other words, these entries are how you get from MENU0000.CTL to the other menus as shown in the map above. Entry 8 is also a menu GOSUB command and will transfer control to MENUTERM.CTL. This menu is used to allow a final message to be left to the SYSOP before logoff if the user desires. We will examine MENUTERM.CTL in more detail below.

The next group consists of entries 9, 10, 11, and 18. These are all commands of the "Display Selection from List" functional group. This type of command is very powerful, as it allows groups of text files to be either displayed in a word wrapped fashion when selected by the caller (entries 9, 10, and 18) or

downloaded with a choice of protocols as in Entry 11. This class of command works with files which are grouped by name in a special way. We will look at Entries 9 and 10 to explain. In entry 9, the Opt Data field is set to the string INFO. On the release disks are five files with names beginning with the letters INFO. It is these files which are displayed by the command in Entry 9. When the caller presses the I key, TBBS will load the file INFOM.TXT and display it to the user. If you type this file out, you will see that it is a list of four other files which may be chosen. These files are chosen by the user typing a 1, 2, 3, or 4. The files which will be displayed are INFO1.TXT, INFO2.TXT, INFO3.TXT, or INFO4.TXT respectively. Since up to a four character (letters or numbers) selection is accepted by TBBS, almost 40,000 different selections are possible on one command. Of course, as shown here, you really don't do it that way. You group a few related files under one selection list command, and then place more such commands on your menu for other groups of files. In the sample system, the other supplied group is the BULL file group. The groups DOWN and PROD are supplied only with a dummy selection list file (PRODM.TXT and DOWNM.TXT) since no actual product listing or download files are provided. To activate these sections, just edit the "M" files and name the selection files accordingly. Downloading takes place simply by specifying a command type which indicates you want it, as in Entry 11.

Entries 7, 13, 14, 15, 16, and 17 provide general service functions which are fairly self explanatory.

Entry 19 invokes one of the more powerful TBBS features, the question and answer file. In this case it invokes the file ORDER.QAF. This file may be examined using QAEDIT. It will take a merchandise order and record the order in a file named ORDER.SVY. See chapter 5 for QAEDIT file details, and Chapter 6 (Advanced Techniques) for some of the more powerful Q & A file applications.

Now we will look at the menu MENUTERM.CTL. This menu shows an example of the sort of "tricks" you can use to give TBBS extra capabilities. The menu looks like this:

Listing of File MENUTERM.CTL

Do you wish to leave a private message to the SYSOP?

```
Entry  1: <Y>es
        Priv=   0  Auth 1= --------  Auth 2= --------
         Key=   Y  Type=   7    Opt Data=MSG TO SYSOP

Entry  2: <N>o ... Terminate Session
        Priv=   0  Auth 1= --------  Auth 2= --------
         Key=   N  Type=  10    Opt Data=**none**

Entry  3: <A>bort Logoff
        Priv=   0  Auth 1= --------  Auth 2= --------
         Key=   A  Type=  12    Opt Data=1
```

The trick in this menu is to ask a question with the menu title, and then use the menu command entries as answers to the question. In this case a Yes answer will cause a leave message command to be activated, posting a message to the special MSG TO SYSOP board. After this message is left the user will be logged off since the Chain Logoff to MSG TO SYSOP write option in CEDIT is set to YES. A No answer will activate the Type 10 immediate logoff command, while an Abort answer will execute a return to the calling menu.

Examine the rest of your sample system and experiment with making changes in it. This is the best way to learn about TBBS. Read the following chapters trying to learn the basics about TBBS construction and operation. Finally, browse the Advanced Techniques section for some examples of how to get the most power from your TBBS system.

You will doubtless be able to devise even more unique ways to apply the power of TBBS. Have Fun!!

THE FUNCTIONS OF CEDIT

The system program CEDIT (pronounced see edit) is your TBBS configuration editor. It is used to create or edit the file CONFIG.CTL, which defines the many options and characteristics of your system -- everything from the welcome text, to the names and types of your various message boards.

Because we have provided a sample CONFIG.CTL file, you will rarely (if ever) need to create a new CONFIG.CTL file. Instead, you may edit the sample file (using CEDIT) into whatever system configuration you want.

Before you use the configuration editor, however, let us briefly describe all its functions. Here is a list of the menu options you will be shown when using the configuration editor:

```
<1> List Current Configuration
<2> Add/Change a Message Board Definition
<3>     Delete a Message Board Definition
<4> Add/Change a Terminal Definition
<5>     Delete a Terminal Definition
<6> Add/Change Greeting Text
<7> Change Options
<8> Exit and Write CONFIG.CTL File
<9> Abort Editing.
```

Here are the functions which may be performed under each of the CEDIT options:

1. List Current Configuration.

   CEDIT will display for you the full CONFIG.CTL file, making no changes. You will be prompted for the option of viewing the file on your console display, or sending it to the printer.

2. Add/Change a Message Board Definition.

   You may use up to 25 separate message boards within TBBS. CEDIT is used to name, define and establish the parameters by which each of these boards is accessed and used. If you have not defined a Message Board with CEDIT, TBBS will not be able to find the board.

3. Delete a Message Board Definition.

   Used when you want to delete a message board from your system without replacing it.

4. Add/Change a Terminal Definition.

   You may define up to 25 terminal types to be displayed to users, at logon, for them to inform your system of what type of terminal they are using.

5. Delete a Terminal Definition.

   Used when you want to delete a terminal definition from your system without replacing it.

6. Add/Change Greeting Text.

   Used to write new logon greeting text for your system, or to edit the existing text.

7. Change Options.

   Options are divided into four groups: System options, Message Board options, Dayfile options, and Logon options. These options control such items as what disk drive should receive uploaded files, what information the system should save to report to you, what privileges new users may have on your system, and other options. We will explain these in detail later in this chapter.

8. Exit and Write CONFIG.CTL File.

   Used when you are through editing with CEDIT. This option prompts you for drive placement of the CONFIG.CTL file.

9. Abort Editing.

   Used to make a graceful exit from CEDIT, without making any changes in CONFIG.CTL.

The method and style of operation of CEDIT is similar to that of all TBBS editors. We are about to guide you through a practice session in its use.

A Review of CEDIT operations

Let's practice using the configuration editor. Place a system disk containing copies of CEDIT and CONFIG.CTL, along with a system operating disk, in your drives. From the DOS ready prompt, enter CEDIT.

CEDIT will load, and ask you:

<1> Create configuration File
<2> Edit existing configuration File
<3> Abort CEDIT

Your Choice?

You are reviewing the existing file, so choose 2 -- and CEDIT will seek out CONFIG.CTL and bring it into memory for editing. Then CEDIT will display its full menu, as shown on Page 5.

Let's first examine the Current Configuration. Choose Option <1> List Current Configuration.

CEDIT will then display the first of seven screens of information, as shown here. (Our text words, SCREEN, will not be on your display.)

(You will need to key ENTER to change to successive screens, or <Q> to quit the review and return to the CEDIT menu. You are not changing the file, only reviewing it. The settings shown are for the CONFIG.CTL file supplied with the sample system on your release disk(s).)

```
-------------(SCREEN #1)------------

            SYSTEM OPTIONS

  Chat Mode = 1
  Chat command Display Method = 2
  Local Display Rate (0-7, 0=Fast) = 3
  Auto-Blank Idle Display = 3
  Inactivity timeout = 4 mins
  Uploads go to Drive B
  Reboot after all logoff = NO
  300/1200 Baud detect Enabled = NO
  European modem handling = NO
  XOFF requires XON (NO=CP/M style = YES
* ZCLOCK hardware clock = NO
* ZCLOCK using alt port set = NO

  -----------------------------------

* Present on TRS-80 Version only

-------------(SCREEN #2)------------

        MESSAGE BOARD OPTIONS

  Auto Delete on Standard Boards = YES
  Auto Reply on Standard Boards = YES
  Pause Request on Message Reads = YES
  Follow Replies if no Pause (Y/N/A) = ASK
  Chain Logoff on MSG TO SYSOP write = YES
  Passwords on Public Messages = YES

-------------(SCREEN #3)------------

            DAYFILE OPTIONS

  Day File Route = 1
  Divider Bar between Calls = YES
  Log Menu File Loads = YES
  Log Data File Reads = YES
  Log failed password attempts = YES
  Log changed passwords = YES
  Dayfile Drive = A
*  Dayfile User Area = 15

* On CP/M version only
```

```
-------------(SCREEN #4)------------

              LOGON OPTIONS

Logon Method = 3
Allow New Users = YES
Security Echo Password = YES
SYSOP Password = PASSWORD
Allow Duplicates Names in Log = NO
For New User: Priv = 25
                  A1 = X-------
                  A2 = --------
   Call Time Limit = 30 min
Count local logons as Calls = NO
Wait Msg on Userlog Search = NO
Wait Msg on Message Base search = YES
Display Active Combined Board List = NO
Extended header (SIGNON1.TXT) = NO
News file (SIGNON2.TXT) = YES
See News file once only = YES
New User Info (NEWUSER.TXT) = NO

-------------(SCREEN #5)------------

              WELCOME HEADER

Welcome to TBBS - The Bread Board System

        (c) 1984 Philip L. Becker


--------------(SCREEN #6)-------------

          MESSAGE BOARD DEFINITIONS

Board      Board        Rd   Wr     A1       A2     T P C  F T D
 #         Name        Priv Priv   Flags    Flags  Y M B  R C A
-----  -------------- ---- ---- -------- -------- - - -  - - -
 01    MSG TO SYSOP    255   0   -------- -------- S N N  Y Y Y
 02    BULLETIN BOARD   25  25   -------- -------- S N N  Y Y Y
 03    E-Mail           25  25   -------- -------- E      Y Y Y
 04    TRS-80           25  25   -------- -------- S Y Y  Y Y Y
 05    APPLE            25  25   -------- -------- S Y Y  Y Y Y
 06    ATARI            25  25   -------- -------- S Y Y  Y Y Y
 07    IBM PC           25  25   -------- -------- S Y Y  Y Y Y
```

----------(SCREEN #7)----------

TERMINAL DEFINITIONS

Term #	Terminal Name	# of Nulls	Clear Screen	Line Feed	U/C Only
01	VIDTEX	0	27:106: 0	Y	N
02	TRS80 1/3	0	28: 31: 0	Y	N

After displaying all seven screens of information of the Current Configuration, CEDIT will return you to its menu. You may wish to scan the screens again, now that you're familiar with the procedure. If your printer is ready, you will probably want to print the file for reference.

Let's begin editing the sample CONFIG.CTL file for your TBBS system. Bring CEDIT's command menu to your screen, if not already there.

Select, <6> Add/Change Greeting Text, and CEDIT will ask you if you want to <E>dit or <R>eplace the text. Choose R.

You may now type in whatever greeting text you would like to use for your system to identify itself to users, up to 255 characters. However, do NOT key ENTER until the text is complete -- use the down-arrow (line feed or CTRL-J on CP/M systems or when using remotely) to change lines. Do not use the ENTER key (carriage return for CP/M systems) until you are finished with the text.

(Even if you make an error in a line, but don't notice it until your cursor is on a later line, proceed -- you will be able to return to the text from the CEDIT menu later and edit it line by line.)

In writing your logon greeting, you should keep your potential users in mind. TBBS can format its display to any column size screen or terminal (up to 132 characters). However, your greeting will be displayed before your system knows what type of terminal is online. If your lines are longer than 21 characters, they will wraparound on many screens, and if they're longer than 39 characters, they'll wraparound on even MORE terminals. Here's a sample logon greeting:

```
WELCOME TO TBBS HQ!
   A SERVICE OF
    ESOFT, INC.
    AURORA, CO
```

How did we center the lines? The good old-fashioned way: On paper, counting spaces...

Now that you've written the greeting text for your system, the next step is to set up the System Cptions. You may wish to scan the next few pages first.

Setting Up Your TBBS Options

If you have not already done so, bring up CEDIT.

Select Item <7> Change Options.

There are 40 Options, presented in four groups. Most of these options are of the "set once and forget" variety, so try not to be overwhelmed by the number. Each option is set by your choice of a number, keying a "Y" or "N" for yes or no, or setting a flag -- "X" sets the flag "on" while "-" sets the flag off.

We will go through the options one by one, explaining the choices for you to make. As you make a change, it is saved to memory for later writing to file, and you may review your choices prior to filing them. Each option will show you its present setting -- if you key ENTER, no change is made.

Change SYSTEM OPTIONS

Select <1>Change SYSTEM Options. These options control certain aspects of overall TBBS system behavior as follows:

o Chat mode = (0, 1 or 2)

This option allows you to determine who may use the "chat" menu option, and how they may use it. (Also see the next option for how the chat mode function is handled.)

Mode 0 will allow all users (regardless of authorization and privilege levels) to see and use the "chat" function wherever it appears on a menu.

Mode 1 causes the menu "chat" option to be treated like all other menu options -- it may be seen and used only by users with the appropriate authorization and privilege levels.

Mode 2 allows the SYSOP to "turn off" the chat option altogether, regardless of how it may be listed on TBBS menus. (Say you wanted to not be disturbed for several hours. Setting Mode 2 for this option would make "chat" disappear from all system menus, but allow you to bring it back rapidly without re-editing the menus.)

o Chat command Display Method = (1 or 2)

This option determines what happens when the user selects the "Chat with SYSOP" item from a menu. In either mode, the SYSOP is summoned by a bell character from the online printer and also by a tone pulse. Note that on the TRS-80, this tone pulse is generated by the cassette port, while on the IBM it is the internal speaker. On CP/M systems, the actual method of tone generation is machine dependent.

In Mode 1, TBBS will pause and start ticking off seconds by printing periods on the user's display for 60 seconds. If the SYSOP presses the <F1> key during this time, chat mode is entered and SYSOP and user may communicate directly from their keyboards. If, during the 60 seconds, the user presses any key, the message "Paging aborted" will be sent to the user, and normal system operation will resume. Note: on a CP/M system it is the <CTRL-V> key sequence, and <F1> on the TRS-80 is either <SHIFT> key.

In Mode 2, TBBS will display a message to the user saying, "Paging System Operator... The SYSOP has been paged. If he is around he will break in. In the meantime, please continue using the system normally." Then TBBS will continue normal operation for the user.

NOTE: In either mode, the SYSOP may always enter the "chat" function from the system terminal by depressing the <F1> key at any time a user is logged on, whether he has been paged or not. (On CP/M systems the break-in chat function is entered by pressing the <CTRL-V> keys, while on the TRS-80 press the left-right arrows simultaneously.)

o Local display rate (0-7, 0=Fast) =

When you log on TBBS from your host terminal, the display on your screen is not controlled by the transmission rate of the modem, as it is for remote users. This option determines how fast the system will send information to your terminal in local mode. You will need to experiment to determine what local display rate you prefer.

o Auto-Blank Idle Display =

This option determines the number of minutes the TBBS display logo is shown on your online terminal during idle periods, or a 0 to indicate that it should not automatically turn off at all. Leaving it on constantly may cause CRT "burn-in" of the image...but if your terminal is on public display, you may not mind that risk (and we at eSoft would certainly appreciate the advertising.) However, if you have no reason to "show off" the terminal display, we recommend you select 2 minutes. The maximum value here is 255 minutes.

o Inactivity timeout =

Enter the number for the number of minutes the system should wait without input. At the end of this amount of time, the system will warn a user, "One more minute without input and you will be disconnected!" TBBS will halt the modem carrier signal one minute later, if there has been no input. An entry of 0 here will disable the inactivity timeout feature. The maximum value which may be used is 255 minutes.

(NOTE: At logon, the timeout clock starts when the two modems handshake, so an amount of three minutes or less may cause the warning to flash before some slow typists complete their logon routine. If you set the time limit in very low numbers, you may want to test the timeout function before putting it online. Also, a number lower than 3 here can cause erratic results.)

o Uploads go on Drive

Select the drive to which you want TBBS to route uploaded files.

o Reboot after all logoffs = (YES or NO)

TBBS will normally re-boot no less often than every two hours, to refresh CPU and RAM circuits; or it will re-boot on a loss of carrier signal. Choosing YES for this option will cause an additional re-boot following each user logoff. NO will leave the system to re-boot after 1-2 hours of idleness. (YES can be helpful in the situation of a system having source AC power fluctuation problems, but it will extend reset of the system from approximately 5 seconds to approximately 30 seconds.)

o 300/1200 Baud detect Enabled = (YES or NO)

If you have a 300/1200 baud modem properly wired to the system, then YES is the correct answer. Answer NO if you are using 300 baud only. If you use a 300/1200 baud modem, please see the section on modem installation for the correct special cable for your system.

o European modem handling = (YES or NO)

For European modems and some older American modems, it is your computer terminal and not the modem, which must do the monitoring for calls and timing-out for connections. (The difference here is whether your computer CPU or your modem manages the Data Terminal Ready and Ring Indicator controls of the RS-232 port.) YES will cause the CPU to set DTR low when idle, and monitor RI. NO will leave the RS-232 management up to your auto-answer modem. (The instructions with your modem should explain what functions it handles, and what it expects from the computer CPU.)

o XOFF requires XON (NO=CP/M style) = (YES or NO)

This option determines how a user's computer terminal and your host terminal will conduct flow control of characters between them. In both cases, an XOFF (CTRL-S, DC3 or CHR$(19)) will halt TBBS character transmission.

If you answer NO to this option, any character output from the user will re-start flow which is as CP/M operates from its keyboard.

If you answer YES, then TBBS must receive an XON (CTRL-Q, DC1 or CHR$(17)) to resume character flow (which is how many mainframe computers operate.) If you're uncertain of this option, then choose "NO." (You may read more about flow control functions in the Information File "System Protocols" included with your TBBS master diskettes.)

o ZCLOCK hardware clock = (YES or NO) (TRS-80 systems only)

The ZCLOCK (available from eSoft) is a battery-operated real-time clock, from which TBBS will take its clock readings if it is installed and YES is selected for this option. Answer NO if you do not have the ZCLOCK installed on your TRS-80.

o ZCLOCK using alt port set = (YES or NO) (TRS-80 systems only)

Answer NO if you do not have ZCLOCK. If you have ZCLOCK, then answer NO if you are using the standard port setting of &HB0, and YES if you are using the alternate setting of &H70.

o MODEM Initialization string

Any character string you enter here will be sent to your modem when TBBS is loaded, and no carrier is present. This can be useful in programming smart modem setup parameters.

Change Message Board Options

Next select main menu option <7> and sub option <2>Change MESSAGE BOARD options: These options control the operation of your system's message boards as follows:

o Auto Delete on Standard Boards = (YES or NO)

A Standard Board is a non-EMAIL board. Do you want to leave non-private messages where they can be read by others, even after the addressee has received them? A YES to this option will prompt the addressee with, "Do you wish to delete this message?" A NO will skip the prompt. (EMAIL boards will always ask this question.)

o Auto Answer on Standard Boards = (YES or NO)

This option determines whether the recipient of a message is prompted by TBBS to answer the message. ("Do you wish to reply to this message?") EMAIL Boards will always ask this question.

o Pause Request on Message Reads = (YES or NO)

When a user chooses a Forward or Reverse Multiple read mode for a message file, a YES to this option will have TBBS ask the user, "Do you wish to pause after each message?" If the user keys Y, the system will pause after each message, until the user keys ENTER to continue. Setting this option at NO will scroll the messages without pause, unless the user keys P to pause or S to stop. At the pause a caller also is given the option of replying to any message in the system as he reads it.

o Follow replies if No Pause (Y/N/A) = (YES, NO, or ASK)

When a user answers NO to the pause question, or if the Pause request option above forces his answer to NO, then this option controls the handling of reply chains during message retrieval. If you set this parameter to ASK, then the user is asked for each retrieve how he wishes to handle reply chains. If you set this parameter to YES, then reply chains will always be followed on retrieve, and if you set this parameter to NO, then reply chains are ignored and messages will always be retrieved in strict numerical sequence.

o Chain Logoff on MSG TO SYSOP write = (YES or NO)

Setting this option at YES will cause TBBS to terminate a call from a user after a message is written to the "MSG TO SYSOP" board. Setting it at NO will cause TBBS to return the user to whatever menu preceded the "MSG TO SYSOP" entry routine.

o Passwords on Public Messages = (YES or NO)

In order to prevent just anyone from deleting another's messages, the delete message command verifies that the user knows the message password before allowing anyone other than the sender or recipient of a message to delete it. To support this, a password is requested for each public message when it is entered. However, if you set up your system without using the delete message command, then asking for this password serves no function. If this is the case, setting this parameter to NO will suppress requesting passwords on public messages.

Change DAYFILE OPTIONS

Next select main menu option <7> and sub option <3> Change DAYFILE OPTIONS. These options control the logging of information about how your system is being used.

o Day File Route = (0, 1, 2 or 3)

This option determines how the system logs user activity. TBBS creates a file named DAYFILE.LST, making entries in it for each user access.

Mode 0 suppresses all logging. The remainder of the options in this section have no meaning if Mode 0 is selected.

Mode 1 will send the DAYFILE.LST information to only your

system printer. Note: The DAYFILE.LST options involving a printer will NOT "hang" if the printer goes off-line for some reason.

Mode 2 will send DAYFILE.LST to only an ASCII file on a disk for your later PRINT or LIST reading.

Mode 3 will send DAYFILE.LST to both the printer and a disk file.

- Divider bar between calls = (YES or NO)

  Setting this option to YES will cause a divider bar to be printed between each call on your log. Setting it to NO will suppress this bar to save disk storage or paper.

- Log Menu File Loads = (YES or NO)

  This option determines whether TBBS will write to DAYFILE.LST what menus are called up (loaded) by a user. (This, and the following option, you will probably want to set at YES initially, just so you can see the full potential of DAYFILE.LST.)

- Log Data File Reads = (YES or NO)

  This option determines whether TBBS will write to DAYFILE.LST what data files are read by a user. (Setting Options 3 and 4 both at YES can create a rather lengthy DAYFILE, so you may wish to consider NO after you have familiarized yourself more fully with TBBS.)

- Log failed Password attempts = (YES or NO)

  Setting this option to YES will cause any invalid passwords entered by callers at logon to be printed on the DAYFILE.LST log. This can aid in determining if someone is "hacking" your system for passwords.

- Log Changed Passwords

  Setting this option to YES will cause all new passwords to be logged when using the change password command.

- Dayfile Drive =

  Enter here the drive you wish the DAYFILE.LST file to be placed on if you have selected option 2 or 3 for Dayfile routing.

o Dayfile User Area = (CP/M systems only)

Enter here the user are the DAYFILE.LST file should be placed in if you have selected option 2 or 3 for Dayfile routing.

Change LOGON OPTIONS

Next enter main menu option <7> sub option <4>Change LOGON options. These options control how your systems looks and acts when a user tries to log on as follows:

o Logon Method = (1 ,2 or 3)

This option determines how your TBBS handles users logging on.

Mode 1 (password method) will first display your greeting text, then prompt the user with, "Enter Password (CR if new user):" and await user entry. The user will be given three tries at correct entry of the password. If an apparent password is entered, TBBS will check the User Log for the password. If found, the system will send back the user's name and calling location and complete the logon procedure appropriate for the user's authorization and privilege.

If the password is not confirmed after three tries, the modem carrier signal is terminated -- TBBS hangs up.

If a carriage return is entered, TBBS will initiate a new user routine, asking for name, calling location and password entry, then write the information to the user log and continue. (The password entered under this mode will be needed by the user for the next access of your system.) Even if a carriage return is entered, and the user proceeds with a name entry, TBBS checks USERLOG.BBS for the name. If the user is in the log, with a password, the password is required as above.

Mode 2 (name method) will first display your system greeting text, then ask the user to input first name and last name. If the name is found in the User Log, the system will send back the calling location for verification by the user. If a password is found in the User Log, TBBS will then give the user three tries to enter it correctly. If there is no password in the User Log, the system will log on the user by name alone. (Use of this mode allows the SYSOP to maintain an open system for new users, but also provide for a

password entry routine elsewhere by which users can be given higher privileges and authorization.)

Mode 3 is identical to Mode 2, but asks a new user to supply his own password at logon. This method seems to be the best defense against a "hacking" entry into a system.

(NOTE: Mode 1 above is the fastest logon method for your regular users, while Mode 2 allows the more open system for new users. Although you may switch back and forth between the two systems, you may not use both modes simultaneously, and switching back and forth is likely to confuse your regular users. You may wish to first put your system online using Mode 2, then switch later -- with warning to regular users -- to Mode 1. TBBS also has a small online help file to inform users who may not understand the abbreviation <CR>.)

o Allow New Users = (YES or NO)

Setting this parameter to NO will disallow new users entirely. If a user attempts to log on who is not in the USERLOG.BBS file, he will be logged off with a message. If you have specified the NEWUSER.TXT file is present below, then that file will be displayed to him instead of the message before he is hung up on. In this mode new users must be added with ULEDIT by the SYSOP to be allowed on the system. Setting this option to YES will allow new users to be automatically added to the userlog with the default new user settings specified below.

o Security Echo Password = (YES or NO)

This option determines whether the characters of a password can be read by a user at time of entry.

The NO mode will echo the entry characters from TBBS back to the user, thereby allowing the user to correct typographical errors...but also allowing anyone near the user's terminal to see the password.

The YES mode will echo back to the user a "*" for each character of the password, so the user (and anyone nearby) does not see the actual character entered. This can cause confusion to a user (particularly between upper and lower case character entries), but TBBS will allow the user three tries at getting the entry correct. (Three incorrect tries will cause TBBS to instantly terminate the modem carrier signal.)

Which of the above modes you use must be determined by your consideration of the type of users you expect on your system, and how you expect them to use it. You may switch your system back and forth (at the risk of user confusion) without changing other system options.

o SYSOP logon Password =

If your system uses Logon Method = 1, then this password has no function.

If your system uses Logon Method 2 or 3, then the SYSOP should enter this password in response to the "First Name ?" prompt at logon to indicate that he wishes to log on remotely as SYSOP. Since this password just serves as the First Name/Last Name for the SYSOP, TBBS will also check for a password in the SYSOP's User Log entry and, if one is found, ask for it too. (They can be the same or different, as you prefer.)

o Allow Duplicate Names in Log = (YES or NO)

If you select NO to this option, then your TBBS system will expect two users who the same name to have the same calling location and, hence, the same password -- resulting in them being considered as only one user by the system. Selecting mode YES will allow as many users with the same name as there may be using your system, as long as their other identifiers (city, state calling location) are different.

(NOTE: Even a YES here will not overcome the unique problem of two same-name users from the same city, who choose to enter their identifiers identically, So NO should be used except in very special circumstances.)

```
o  For new user: Priv =
o                  A1 = --------
o                  A2 = --------
o     Call Time Limit =    min.
```

These are the options where you decide the privilege and authorization level your system will automatically grant to a new user. These settings relate to the privilege levels and authorization flags on your various system menus, and will determine what a new user can see and do. The new user's parameters are taken from these settings and written into the User Log file, where you may edit them to other levels when you want to give a former new user some other type of access.

Priv is the privilege level, 0 to 255, with 255 at the SYSOP or co-SYSOP level.

A1 and A2 each contain eight "flags" which operate independently of each other. They are set by typing an "X" at any of the eight positions. (SYSOP normally has an "X" in all eight positions of both A1 and A2.)

If an "X" flag is set in a particular menu item, then only users with an "X" set in that same A1/A2 position in their User Log file may see and use that menu item. If you do not want to set the flag, enter a dash "-" at that position. For your initial set up, we recommend you use the settings in the sample CONFIG.CTL until you are more familiar with the overall TBBS system.

Call Time Limit is the amount of time, in minutes, you allow a new user to access your TBBS before a mandatory disconnect. The user will receive a One Minute Warning before being disconnected. Setting this to 0 allows unlimited use of the system and 255 minutes is the maximum setting.

o Count local logons as Calls = (YES or NO)

Many TBBS system operators perform most of their system maintenance by logging on from the host terminal. This option determines whether the system will count such logons as user calls, or not. If you are trying to keep a realistic record of user activity, we recommend NO for this option.

o Wait Msg on Userlog Search = (YES or NO)

This option controls whether or not a "Searching Userlog..." message is printed when TBBS is looking up names in its log. If your system is running on floppy disks and takes over 30 seconds to find a name at times, we recommend setting this to YES to reassure your callers.

o Wait Msg on Message Base Search = (YES or NO)

This option controls whether or not a "Searching Message Base" message is printed when TBBS is checking for personal messages at logon. As with the above option, the speed and size of your system determine the need for this.

o Display Active Combined Board list = (YES or NO)

Setting this option to YES, will cause a list of all active

combined message board names to be displayed to the caller at logon. A NO here, suppresses this display. We recommend a NO here to avoid confusion to new callers, unless you wish to advertise your combined board section.

o Extended header (SIGNON1.TXT) = (YES or NO)

The normal TBBS text buffer for your system's logon Greeting is 255 characters. Should you have need for a longer text, answer YES here. Then write your Greeting Text in an ASCII file named SIGNON1.TXT. The system will select that file for display, after it displays the Greeting Text file you write in the Welcome Header (below.) If you answer NO only your Greeting Text will be displayed before beginning actual logon procedures.

o News file (SIGNON2.TXT) = (YES or NO)

By answering YES here, you may have your system display a "news" file to each user, after logon, but prior to the top menu. The news file, written in ASCII, must be named SIGNON2.TXT. Answering NO will omit that function.

o See News file once only = (YES or NO)

Used with above option, this allows you to decide whether the SIGNON2.TXT news file is shown only once (YES) to a user as they logon, or every time they logon. The SIGNON2.TXT could also be an advertisement for the system or its sponsor, in which case your answer could be NO and the ad would appear at each logon. (You can use a function of ULEDIT, the User Log editor, to reset the display flags in your user file so that when you change the SIGNON2.TXT, the new file will be again shown to each user.)

o New User Info (NEWUSER.TXT) = (YES or NO)

If you answer YES to this option, the text file you write and name, "NEWUSER.TXT" will be shown at logon to each user whose name is not presently in your system's User Log. NO will tell your TBBS to skip the file.

(NOTE: If you choose to use any of the files SIGNON1.TXT, SIGNON2.TXT, or NEWUSER.TXT -- then they should be online at runtime. Your TBBS will not "crash" if it doesn't find them, but logon time will lengthen considerably while the system searches for, and fails to find them.)

This completes your setup of the TBBS system configuration

options. By now, you are more aware of some of the variety and flexibility of the system you are building. Also you have become aware that changing much of your system is easy and fast -- call up CEDIT, and do it!

Defining Your Message Boards (BBS's)

Your sample CONFIG.CTL file contains several Message Boards already defined in it. Prior to editing, you may again wish to list the current configuration for reference (or refer to a printout).

You may establish as many as 25 separate message boards, and TBBS will keep track of each of them. On the other hand, only one Board (MSG TO SYSOP) must be on line for the system. So, you could operate a full TBBS system with as few as two boards.

We recommend you start small while you're becoming familiar with the system. Let's examine the sample Message Board Definitions. You may later edit them to your taste.

You determine the name of each Message Board (except MSG TO SYSOP), and, by the combination of read/write privileges and the A1/A2 flags, you also determine what users may see it, read it and write to it.

And, you determine whether each board is an EMAIL (messages to and from users by name) or Standard board, whether the messages on the board are combinable with messages from other boards, and what information is displayed on a Scan of the board's Message Headers.

Later, when you create or edit your various command menus for message boards, your Menu Control File information must correlate with your Message Board Definitions. (In Chapter 3, we will show you how to do that.) If you name a board in a menu, without creating it with CEDIT, it will produce a system error when users try to access the board. Or, if your Menu Control File privilege and flag information is more restrictive than that for the board in the Message Board Definitions, users won't even see the board!

Let's look at the Current Configuration of your sample CONFIG.CTL. (Refer to Page 8.) The definition information is listed in columns.

Board # is the system's number for the board -- the way it keeps track, in the message files, as to where a message belongs.

CEDIT will automatically increment the board numbers if you add a board, or allow you to edit an existing board by its number.

Board Name is the name which must be used in your Menu Control Files to refer to the board. It is also the name displayed to users, when the system shows messages from that board or scans the message headers.

Read Privilege is the lowest number a user's USERLOG.BBS parameters must match or exceed to read messages on that board. As you can see by your sample of the MSG TO SYSOP board, the numbers do not need to match -- only the SYSOP, with a 255 privilege, can read the MSG TO SYSOP board, but anyone with a 0 or higher privilege level can write to the board.

The flags need not necessarily be set. If they are, only users with flags set correspondingly in their USERLOG.BBS file may access that board.

The letters atop the right-hand columns are abbreviations for:

TY - Type of board. Standard or EMAIL?

PM - Private Messages. Allow them or not? (Not needed for an EMAIL board.)

CB - Combine Board. Can the board's messages be combined with messages from other boards for display to a user with the proper access privileges and flags? (Not possible with an EMAIL board.)

FR, TO and DA - From, To and Date. How much information should be displayed to a user on a Scan of the message headers?

Let's add a board, experimentally, for practice. (Later, in Advanced Techniques, we will show you how to turn this board into a remote co-SYSOP Message Board for one of your helpful users.)

If it is not in front of you, bring up the CEDIT menu.

Select item <2> Add/Change a Message Board definition.

CEDIT will respond with (1) Add or (2) Change? Choose 1.

CEDIT will now ask you questions to define the board. Remember you are only writing to memory, and you can later review the information before writing to file.

Give the board a name of your choice...up to 16 characters

(spaces may be used, and they count as characters).

Set the Read and Write privileges, both, to 255. (We're deliberately hiding the board for now.)

Set no flags in A1 -- enter "--------" (eight dashes.)

Set the first flag in A2 -- "X-------" (capital X and seven dashes.)

Make the type "S" -- standard.

Allow private messages -- "Y", yes.

Allow the board to be combined -- "Y", yes.

Have a SCAN display all the header information -- "3"

Now, you are back to the CEDIT menu. Select <1>, and skip through the screens to review the definitions of the new Message Board you have created. Double-check that your entries are as above, because we will later use this same information to create the menu by which your helper can use the new board.

There is one other CEDIT function involving message boards -- item <3> Delete a Message Board Definition. If you select <3>, CEDIT will ask you what board, by number, to delete -- and it will delete its definitions. (As soon as you write the edited CONFIG.CTL to file.)

> (NOTE: If you delete a board, all boards with a higher number will move up one. That will cause the messages, if any, on the deleted board and those with higher numbers to show up on the wrong board. You can use MFSQZ.COM to reassign board message numbers, prior to or after deletion of a board. Deleting a board may be simple, but it sometimes can involve considerable clean-up afterward).

TBBS provides a "message waiting" service, informing each regular user at logon, of any messages addressed to them and on what board they are posted. Hence, as you name the various boards of your system, you should keep in mind that those names will be displayed to each user who receives a message on that board.

Also, remember that your Menu Control File text display lines should probably contain some reference to the board's name -- otherwise, the user with a "message waiting" notice can become very frustrated trying to find the menu command to access the board for the message.

TERMINAL DEFINITIONS

Term #	Terminal Name	# of Nulls	Clear Screen	Line Feed	U/C Only
01	VIDTEX	0	27:106: 0	Y	N
02	TRS80 1/3	0	28: 31: 0	Y	N
03	APPLE II	0	13: 13: 0	N	N
04	ATARI	0	125:126:126	Y	N
05	H19/H89/Z19	0	27: 69: 0	Y	N
06	IBM PC	0	12: 0: 0	N	N
07	TELEVID 925	0	26: 0: 0	Y	N
08	OSBORNE	0	27: 26: 0	Y	N
09	VIC 20	0	19:147: 0	Y	N
10	TI-99/A	0	13: 13: 0	Y	N
11	VT100	0	26: 0: 0	N	N
12	TRS80/OKI	0	13: 13: 0	Y	N
13	KAYPRO	0	26: 13: 0	Y	N
14	TRS-COCO	0	27:106: 0	Y	N
15	TX100/ZX81	0	251: 0: 0	Y	Y

(The above is a sample table of terminal definitions, which may or may not be appropriate for your user area. It also demonstrates the variety of characteristics necessary to meet particular user-hardware configurations -- Terminal #12 above was necessary in an area where several TRS-80 owners were using a printer which was toggled on and off by a portion of the standard TRS-80 clear screen coding.)

Defining Terminal Characteristics

When you log on to a TBBS system for the first time, you are asked to define your terminal so TBBS will know how to configure its display output best for your system.

Then the system will offer you a menu of various terminals from which you can pick yours, if it is among them. The Terminal Definitions file within CONFIG.CTL is where the system stores the information for that menu.

As you have used CEDIT to List the Current Configuration of your startup CONFIG.CTL file, you have seen the Terminal Definitions table which we have supplied you. Bring it to your screen now, or refer to the table on Page 22.

You may define up to 25 terminals, and variations of their settings, from which users may choose. As those listed in your startup file are very common, we recommend you not change or delete them -- but you will likely want to add to them to suit the users in your area.

The information is listed in columns. Here's what it means:

Terminal Number - Actually used only by you for editing purposes, because the display shown to users is alphabetical. If you choose <4> Add/Change a Terminal Definition from the CEDIT menu, you will be prompted for (1) Add or (2) Change, and the system will add a number if that is your choice.

Terminal Name - The name, up to 15 characters, which your system will display the option to users for their choice.

Number of Nulls - A null is blank code (00) which can be sent by your system after each carriage return, to accommodate the slow terminal of a user. This is rarely necessary for a video display. However, users who access your TBBS from a printer terminal may need nulls to avoid losing transmitted characters while the carriage is moving leftward across the printer. This definition option allows you to provide them such a choice.

Clear Screen - Almost all video computer terminals have some combination of code to clear the video display. TBBS allows up to three bytes (listed in decimal here) as a code sequence to "clear screen." As you can see from the startup file, there is considerable variation in what that code can be. (Generally, you may have to depend upon newer, but knowledgeable, users to supply you with the correct code for adding a new terminal definition.) Remember: Your entry here is in decimal.

Line Feed - Some terminals, video or printer, need a line feed command following a carriage return -- some don't. Y for yes, N for no. Again, as you add new terminal definitions, ask the user with that kind of terminal.

U/C Only - TBBS accommodates upper and lower case easily. Some computer terminals, however, cannot. But, some computer terminals originally programmed for only upper case letters, may have been modified. Others, which only input upper case, can convert lower case to upper case easily -- but some, can't. This

is a definition option in which you may want to provide variations, depending upon what your users are actually using.

When you add a terminal definition, CEDIT will ask you, as in the above sequence, to provide each of the settings shown in the table.

This entire table, however, still may not accommodate some users. What then? When your TBBS displays the information in this table, it also offers users the option of not choosing any of the Terminal Definitions. Then, they're prompted to give answers to three questions which should get them logged on at least well enough so they can leave you a message about whatever their unique terminal needs from your end. (We try to keep you ready for anything!)

As new users are written to your User Log, you probably should check their terminal settings (as explained in Using ULEDIT.COM) to be sure they at least appear rational. We have heard from one SYSOP who had a user complaining bitterly about how "slow" his system operated. Seems the user had told the system the terminal needed 50 nulls -- which meant every line of text sent, ended with a two-second pause!

This section concludes your introduction to CEDIT. Because its use determines the overall configuration of your system, it probably needs to be best understood of all your TBBS editors. If there is a topic of the past several sections which you have not understood, we strongly urge you to review this text again and experiment with a practice CONFIG.CTL file.

You need not actually put your system on-line for such experimentation. Bring it up, as we instructed in Getting Started FAST, and run through it as though you were a user seeing it for the first time. If YOU have problems in such practice, then your users will, too! We're proud of TBBS, as we hope you are -- we like to see it used in the best way possible for every SYSOP and every telecommunicator!

If something "just won't work" -- well, give OUR system a call -- the eSoft Software Support BBS, at (303) 699-8222. It's available 24 hours a day, with a special SYSOP area, to which you will have password access after initial registration. Our technicians read the trouble messages, and, what's more, so do a lot of other SYSOP's who probably have an answer to your difficulty from their own experience.

eSoft expects all TBBS owner/operators to check the Software Support BBS at least once a month. You will receive notice there first, of all system modifications, corrections and updates.

THE FUNCTIONS OF MEDIT

The system program MEDIT (pronounced em edit) is your TBBS Menu editor. It is used to create or edit the menu control files which all have names of the form MENU????.CTL, and actually form the command interface between your system and the callers.

MEDIT, along with the menu control file structure, is the secret to the tremendous flexibility of TBBS. The power comes from the available functions which may be invoked by a menu entry, and the system wide use of the privilege and authorization functions. This overall system design allows for many subtle capabilities which are not at first obvious. Some of these may be seen in the sample system discussion in Chapter 1, and others will be demonstrated in the Advanced Techniques section. In all probability over 90% of the time you spend building your personalized information utility you will be using MEDIT. This chapter will therefore go into a large amount of detail on MEDIT and the menu structure it is manipulating.

How Command Menus are Constructed

Command menus are screen displays from which your system users may select a system function to be performed -- write or read a message, display the user log, enter a question-and-answer file, or similar "command" functions.

We will refer to these various functions as "Types" of commands. (Command menus are not "selection lists" for reading or downloading text files. Selection lists are dealt with later in the file display command descriptions.)

All command menus are created and edited with MEDIT, and are stored as Menu Control Files. Each Menu Control File follows the same naming rules. The first four letters of the standard filespec are "MENU" and the extension is .CTL. The remaining letters of the name (as in MENU????.CTL) determine the actual name of the menu, where ???? is any one-to-four alphanumeric characters you choose.

Here is a list of Menu Control Files which could make up a fictitious TBBS system:

MENU0000.CTL	MENUTI99.CTL
MENUTBBS.CTL	MENUVIC2.CTL
MENUMAIL.CTL	MENUTRS8.CTL
MENUTERM.CTL	MENUHELP.CTL

Menu Control Files are not seen by users by the names you give them. They see only menu commands which invoke other menus implicitly. It is wise for you to choose names, however, by which you can readily identify each menu when necessary.

You may have as many Menu Control Files (one for each command menu) as you want. Each menu may "call" another menu, and menus may be sequenced as a series of sub-routines up to 20 levels deep -- TBBS will still remember where it started, and the path to return. Or, a menu might be in the path of several other menus, with you determining the call and return sequence. Menus may also be strung together using GOTO constructs allowing you to create any structure you wish. It is strongly suggested that you make a map of your menu structure to avoid getting lost and creating menu branching inconsistencies. See the section on mapping your system in the Advanced Techniques portion of this manual.

One Menu Control File -- MENU0000.CTL -- must always be on-line for TBBS operation. It is called into memory by TBBS at boot or after a logoff sequence, ready for use at the next user logon. It is displayed to users after they have completed their logon procedure, and is the "top" or "prime" menu for all other system functions. In other words, TBBS goes to menu 0000 to start things rolling and to recover from problems. This is the only menu name which TBBS predefines. All other names are up to you.

Among the key factors to consider when constructing a menu is its appearance to your users. On a 16 line calling terminal, as an example, a menu with a two-line heading can display only 11 items before starting to scroll off the screen in the Beginner menu display mode. Also, some computer terminals have displays narrower than 80 columns, 64, 40, 32, and 22 column screens are in common use among low cost computers. Long text lines will wrap around tc the next line on them, rapidly filling the screen. Together, these limitations mean you can consternate many of your users with menus that have many visible options and long text lines. But, if you are accommodating to that group of users, those with 80-column, 40-line displays may wonder why you cram everything into the top left corner of their screen. You must be the judge of what is appropriate, depending on your expected users. TBBS will support up to 132 column terminals smoothly.

Each menu entry consists of three main parts. Part 1 is the text which will be displayed to the caller if he qualifies for this menu entry. Part 2 is the privilege level and A1/A2 authorization flags which determine if a given user does qualify for this menu entry. Part 3 specifies the key the caller needs to press to invoke the menu entry and the type of command which

will be executed if he presses that key.

Using the privilege level and A1/A2 flag settings, you can create lengthy menus but have the display controlled by a user's authorization. (TBBS will skip those lines in a menu which the user is not authorized to see or use.)

On your distribution disk(s) we have supplied you with a sample MENU0000.CTL Menu Control File. On your screen, it looks like this when listed using the MEDIT list text only command:

```
TBBS Top Level Menu
-------------------

<R>ead ... Read Msgs on TBBS board
<Q>uick-scan Msgs on TBBS board
<L>eave ... Leave Msg on TBBS board
<K>ill ... Kill Msg on TBBS board
<M>ail ... Electronic Mail Section
<S>igs ... Special Interest Groups
<T>ime ... Elapsed Time
<G>oodbye ... Terminate Session
<I>nfo ... System Information
<B>ulletins
<D>ownload Programs
<#>Upload ... Submit Programs
<U>serlog ... List of callers
<F>ormat ... Reconfigure terminal
<X>pert ... Change user level
<C>hat ... Talk to SYSOP
<A>gain ... Re-logon to system
<P>roducts ... List of Merchandise
<O>rder ... Place Merchandise Order
<%>Readall ... For SYSOP - Reads all Boards
<E>ditors ... Remote SYSOP section
```

If you log on as SYSOP, this is also the top level menu you will see. Shown are 22 choices a user can make, IF the user is authorized by privilege level and A1/A2 flags to see and use the full menu. In actual use, however, all entries in our sample MENU0000.CTL would not be seen by all users, because the last two items are flagged for the SYSOP and co-SYSOPs. (Please note that our startup sample is to demonstrate operation, not appearance, verbiage or format.)

Note that there is no duplication of <key letters or characters. To avoid using "U" twice, we used a "#", and instead of a second "R", we used a "%". Any ASCII characters may be used, except upper-and-lower case are treated the same, while the "?" and

"ENTER" are reserved for system use. Even control characters are legal if you want to use them for special menu applications.

Now let's look at the first nine lines cf the menu from MENU0000.CTL as they are actually entered in MEDIT. This listing is in the form produced by the MEDIT list to printer. The same form is presented in paged manner when you use MEDIT to list to your screen the full menu display.

```
Listing of File MENU0000.CTL

TBBS Top Level Menu
-------------------


Entry  1: <R>ead ... Read Msgs on TBBS board
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   R  Type=  6   Opt Data=TBBS

Entry  2: <Q>uick-scan Msgs on TBBS board
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   Q  Type=  8   Opt Data=TBBS

Entry  3: <L>eave ... Leave Msg on TBBS board
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   L  Type=  7   Opt Data=TBBS

Entry  4: <K>ill ... Kill Msg on TBBS board
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   K  Type=  9   Opt Data=TBBS

Entry  5: <M>ail ... Electronic Mail Section
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   M  Type=  5   Opt Data=MAIL

Entry  6: <S>IGS ... Special Interest Groups
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   S  Type=  5   Opt Data=SIGS

Entry  7: <T>ime ... Elapsed Time
        Priv=   0  Auth 1= --------  Auth 2= --------
         Key=   T  Type= 11   Opt Data=**none**

Entry  8: <G>oodbye ... Terminate Session
        Priv=   0  Auth 1= --------  Auth 2= --------
         Key=   G  Type=  5   Opt Data=TERM
```

```
Entry  9: <I>nfo ... System Information
        Priv=   0  Auth 1= --------  Auth 2= --------
         Key=   I  Type=  1   Opt Data=INFO
```

Notice that each menu entry has three lines in its display. These lines represent the three parts of a menu entry we described above. In total there are six elements comprising the three parts of each command menu entry. These six elements are:

1. Text line.

   Up to 200 characters, explaining the choice to users.

2. Privilege level.

   A number from 0 to 255, which must be matched or exceeded by the user's userlog file privilege entry for the item to appear on the user's display.

3. Auth 1 & Auth 2 Flags.

   These entries are always listed separately for easier visualization, but they operate together to give a total of 16 feature authorization flags. The user's USERLOG authorization must have a flag set in the same position as any marked here to see and use the menu entry. These are also sometimes referred to as the A1 or A2 flags.

4. Key.

   The letter or other keyboard character by which the user makes a single-key stroke selection from the menu.

5. Type.

   The type of function TBBS is to perform when the item is selected. (There are 43 types, which we will define and describe for you later.)

6. Opt Data.

   Additional data needed by TBBS for it to follow through on some menu command functions, as determined by the type of command. Some "types" require you to have a null entry in the Opt Data field -- MEDIT will show a null entry in the Opt Data field by displaying "**none**".

   (NOTE: When using MEDIT to change an Opt Data entry in an existing Menu Control File, entering a "-" will delete the

field of any previous entry. This is how you get rid of any Opt Data and re-specify a null entry (**none**) during editing.)

In the listing above, the Opt Data entry TBBS refers to the Message Board Definition TBBS in your sample configuration file. The four menu entries showing TBBS are all message type functions affecting the TBBS message board. MAIL, in Entry 5, refers to the menu MENUMAIL.CTL which will be loaded and given control if this option is selected. Likewise, the SIGS Opt Data in Entry 6 refers to the menu MENUSIGS.CTL, just as TERM in Entry 8 refers to MENUTERM.CTL. Remember that configuration file entries are created or changed using the CEDIT utility program while menus are created or changed using MEDIT.

In addition, each separate menu may have some form of title or heading so the user is aware of the overall system purpose of that menu. It can be up to 255 characters long, on multiple lines.

Now that we've seen MENU0000.CTL from both the user and system viewpoint, it's time to practice using MEDIT.

A review of MEDIT operations

We'll now go step-by-step through a practice session with MEDIT, using the MENU0000.CTL Menu Control File. Place a system disk with copies of both files online.

From the DOS ready prompt, enter MEDIT.

MEDIT will load, and ask you:

```
<1> Create a new Menu file
<2> Edit an existing Menu file
<3> Abort MEDIT

Your Choice?
```

Choose 2.

MEDIT will respond:

MENU Name (last 1-4 Char.)?

Enter 0000, which will appear on your screen.

Then MEDIT will display its full menu of options, as below:

<1> List Menu entries
<2> Add an entry at the end
<3> Insert an entry
<4> Change an entry
<5> Delete an entry
<6> Move an entry
<7> Enter/Change Menu Title Text
<8> Exit and Save Menu file
<9> Abort Editing

Here is what each option will allow you to do:

1. List Menu entries

   This option gives you three choices -- list the Menu Control File with all control information (three entries at a time), list only the text of each item, or send the full file to the printer.

2. Add an entry at the end

   The number of menu entries will be incremented by one, and MEDIT will prompt you for the various elements of information it needs for the entry.

3. Insert an entry

   Again, the number of menu entries will be incremented by one, and MEDIT will ask you what existing entry the inserted entry should follow. MEDIT will then prompt you (as in Option 2) for the new entry's information.

4. Change an entry

   MEDIT will ask you which entry to change, then display it to you, field by field. If you do not enter a new value for an field, it will remain as previously entered.

5. Delete an entry.

   MEDIT will ask you which entry, by number, to delete, then display it to you for confirmation before deleting it from memory wich will decrement the number of entries by one.

6. Move an entry

   This option will first ask which entry should be moved, then ask where it should go. It is used to re-arrange your menu.

7. Enter/Change Menu Title Text.

   Create a new menu title or edit the present title, up to 255 characters long. A down-arrow or line feed (CTRL-J) is used to change lines, and ENTER should not be keyed until the text is complete. If you use a down-arrow (or line feed) at the end of your last line, before keying ENTER, it will leave a blank line between your title and the first menu entry.

8. Exit and Save Menu file

   This option will prompt you to use the existing Menu Control File name, or select a new name. This makes it easier to create new menus by editing existing menus, and filing them to a new Menu Control File name.

9. Abort Editing

   MEDIT will ask you first if you want to save the Menu Control File in memory, prior to exit. This gives you the freedom of working on a file at length, but still exiting MEDIT without making any system changes. You will be given a chance to save the file in case you selected this entry by accident.

Listing Menu Entries

Let's review the present MENU0000.CTL, which you have placed in memory. With the MEDIT menu on your screen, choose <1> List Menu Entries.

MEDIT will respond:

> <1> Full Menu Listing, <2> List Menu text only, or
> <P> to Print full menu?

Choose 2 for a fast review of the MENU0000.CTL text. Then, Choose 1 for an entry-by-entry, three-at-a-time, review of all the menu's fields. (If your line printer is available, you will also want to choose P so you have a hard copy for your continuing practice.)

As noted earlier, the startup MENU0C00.CTL is not a model, but a file to get you started -- you probably already have some changes in mind to make. Before making them, however, it is time to review all the "types" of command menu functions at your disposal. At this point, you may be wise to dismount your disks, turn off your computer, and prepare to study the "types" -- all 43 of them.

It is the versatility and flexibility of the menu commands -- 43 types of them -- which has helped The Bread Board System to develop an international reputation among BBS systems for its speed, ease of operation, and unprecedented power. Many of the first TBBS operators and users were veterans of one or more other telecommunications systems and (they have told us), their decision to move to a TBBS was a move to an entire new level of power, performance, and capability.

Each "type" is a command, setting in motion some function from TBBS. As you've already seen, a single-keystroke entry is all that is necessary for a user to command that "type" of function. We will list the menu control types on the next few pages briefly, then explain in detail how to use each one in your system's Menu Control Files.

As you review the listing, do not try to commit the types of functions to memory, or to fully understand how each type and function works. The purpose of the brief listing is to give you a rapid familiarization with the system's power, and a consolidated listing to which you will be able to refer easily later -- when you're more familiar with full system operation.

The first list is in numerical order, and the second list is organized by function groupings. The number is what you will enter, when using MEDIT, in the "Type" field. On the pages following the listing, we will explain the "Types" in detail.

TBBS Menu Control Function Type Code Summary - numeric order

TYPE	Function Performed
1	Display selection list menu, display selection with word wrap, return here. Opt Data = 1st four characters of selection file family name.
2	Display selection list menu, display selection, with word wrap and download protocol options, return here. Opt Data = 1st four characters of selection file family name.
3	Same as Type 1, except return is to MENU0000.CTL. Opt Data = 1st four characters of selection file family name.
4	Same as Type 2, except return is to MENU0000.CTL. Opt Data = 1st four characters of selection file family name.
5	Go to sub-level MENU????.CTL file. Current menu is saved on the menu stack. Return is via the type 12 sub-level exit. Opt Data = the one to four ? characters of MENU????.CTL file.
6	Retrieve Message Command. Calls TBBS message retrieval program. Return here. Opt Data = Name of a defined message board (defined with CEDIT) or No Opt Data for combined board retrieve.
7	Enter Message Command. Calls TBBS enter message program. Return here. Opt Data = Name of a defined message board (defined with CEDIT) or No Opt Data for combined board write.
8	Scan Message Command. Calls TBBS scan message program. Return here. Opt Data = Name of defined message board (defined in CEDIT) or No Opt Data for combined board scan.
9	Delete Message Command. Calls TBBS delete message program. Return here. Opt Data = Name of a defined message board (defined with CEDIT) or No Opt Data for combined boards.
10	Terminate session. Calls TBBS logoff processing prompts. No Opt Data.
11	Display elapsed logon time. Return here. No Opt Data.

TBBS Menu Control Function Type Code Summary - numeric order

TYPE	Function Performed
12	Return to menu level saved in stack. Opt Data = number of levels to go up, or 0 for MENU0000.CTL.
13	Display text file with word wrap. Return here. Opt Data = full text file name.
14	Display text file, with word wrap and download protocol option. Return here. Opt Data = full text file name.
15	Same as Type 13, except return to MENU0000.CTL. Opt Data = full text file name.
16	Same as Type 14, except return to MENU0000.CTL. Opt Data = full text file name.
17	Display partitioned text file. Calls partitioned text prompts. Return here. Opt Data = 1st six characters of file family, then number of parts in file.
18	Same as Type 17, with download protocol option. Return here. Opt Data = 1st six characters of file family, then number of parts in file.
19	Same as Type 17, except return to MENU0000.CTL. Opt Data = 1st six characters of file family, then number of parts in file.
20	Same as Type 18, except return is to MENU0000.CTL. Opt Data = 1st six characters of file family, then number of parts in file.
21	Exit TBBS to remote hosted DOS, without return: Recommended for online use by SYSOP only! No Opt Data.
22	List User Log. Display name, calling location, and last call time of up to the last 255 users. No Opt Data.
23	Chat mode. Invoke chat mode, by options specified in CEDIT. No Opt Data.
24	Select boards. Call TBBS prompts for user to select boards in combined display. No Opt Data.
25	Change password. Call TBBS prompts for password change. Return here. No Opt Data.

TBBS Menu Control Function Type Code Summary - numeric order

TYPE	Function Performed
26	Change Xpert user level. Call TBBS prompts to select menu display level. No Opt Data.
27	Upload file with user-specified name, .UPL as .ext. No overwrites. Return here. No Opt Data.
28	Upload file with user-specified filespec. No overwrites. Return here. No Opt Data.
29	Upload file with user-specified filespec. Overwrites allowed. Return here. No Opt Data.
30	Upload file with user-specified partial name. You force from the first 1 to 6 characters of the file name, and the caller completes the name and extension. Overwrites allowed. Opt Data = 1st one to six characters of file name.
31	Reconfigure terminal. Calls TBBS prompts for user to respecify his terminal options. No Opt Data.
32	Run a Q&A file, created by QAEDIT.COM. Return here. Opt Data = name of QAF file (without .QAF ext.)
33	Reserved.
34	Reserved.
35	Same as Type 1, without wordwrap. (For ASCII program text files.) Opt Data = 1st four characters of selection file family name.
36	Same as Type 2, without wordwrap. (For ASCII program text files.) Opt Data = 1st four characters of selection file family name.
37	Same as Type 3, without wordwrap. (For ASCII program text files.) Opt Data = 1st four characters of selection file family name.
38	Same as Type 4, without wordwrap. (For ASCII program text files.) Opt Data = 1st four characters of selection file family name.
39	Same as Type 13, without wordwrap. (For ASCII program text files.) Opt Data = full text file name.

TBBS Menu Control Function Type Code Summary - numeric order

TYPE	Function Performed
40	Same as Type 14, without wordwrap. (For ASCII program text files.) Opt Data = full text file name.
41	Same as Type 15, without wordwrap. (For ASCII program text files.) Opt Data = full text file name.
42	Same as Type 16, without wordwrap. (For ASCII program text files.) Opt Data = full text file name.
43	Re-enter logon. Calls TBBS prompts for log-on sequence. Returns to calling menu after new user logs on. No Opt Data.
44	Exit to DOS, execute DOS command, return to TBBS (if modem carrier present.) Opt Data = DOS command string
45	Go to MENU????.CTL. Same as Type 5, but no change in return stack. Opt Data = the one to four ? characters of the MENU????.CTL file name.

(NOTE: The term, No Opt Data field means you make NO entry. MEDIT will display **none** for such entries.)

The command types fall into seven functional groups. These seven groups are:

- o MENU BRANCHING GROUP
- o MESSAGE BOARD COMMAND GROUP
- o SINGLE FILE DISPLAY GROUP
- o SELECTION LIST FILE DISPLAY GROUP
- o PARTITIONED FILE DISPLAY GROUP
- o UPLOAD FILE GROUP
- o GENERAL SYSTEM FUNCTION GROUP

TBBS Menu Control Function Type Code Summary - group order

Menu Branching Group

TYPE	Function Performed
5	Go to sub-level MENU????.CTL file. Current menu is saved on the menu stack. Return is via the type 12 sub-level exit. Opt Data = the one to four ? characters of MENU????.CTL file.
45	Go to MENU????.CTL. Same as Type 5, but no change in return stack. Opt Data = the one to four ? characters of the MENU????.CTL file name.
12	Return to menu level saved in stack. Opt Data = number of levels to go up, or 0 for MENU0000.CTL.

Message board command Group

TYPE	Function Performed
6	Retrieve Message Command. Calls TBBS message retrieval program. Return here. Opt Data = Name of a defined message board (defined with CEDIT) or No Opt Data for combined board retrieve.
7	Enter Message Command. Calls TBBS enter message program. Return here. Opt Data = Name of a defined message board (defined with CEDIT) or No Opt Data for combined board write.
8	Scan Message Command. Calls TBBS scan message program. Return here. Opt Data = Name of defined message board (defined in CEDIT) or No Opt Data for combined board scan.
9	Delete Message Command. Calls TBBS delete message program. Return here. Opt Data = Name of a defined message board (defined with CEDIT) or No Opt Data for combined boards.

Single File Display Group

TYPE	Function Performed
13	Display text file with wordwrap. Return here. Opt Data = full text file name.
14	Display text file, with word wrap and download protocol option. Return here. Opt Data = full text file name.
15	Same as Type 13, except return to MENU0000.CTL. Opt Data = full text file name.
16	Same as Type 14, except return to MENU0000.CTL. Opt Data = full text file name.
39	Same as Type 13, without wordwrap. (For ASCII program text files.) Opt Data = full text file name.
40	Same as Type 14, without wordwrap. (For ASCII program text files.) Opt Data = full text file name.
41	Same as Type 15, without wordwrap. (For ASCII program text files.) Opt Data = full text file name.
42	Same as Type 16, without wordwrap. (For ASCII program text files.) Opt Data = full text file name.

Selection List File Display Group

TYPE	Function Performed
1	Display selection list menu, display selection with word wrap, return here. Opt Data = 1st four characters of selection file family name.
2	Display selection list menu, display selection with word wrap and download protocol options, return here. Opt Data = 1st four characters of selection file family name.
3	Same as Type 1, except return is to MENU0000.CTL. Opt Data = 1st four characters of selection file family name.
4	Same as Type 2, except return is to MENU0000.CTL. Opt Data = 1st four characters of selection file family name.

Selection List File Display Group (cont)

TYPE	Function Performed
35	Same as Type 1, without wordwrap. (For ASCII program text files.) Opt Data = 1st four characters of selection file family name.
36	Same as Type 2, without wordwrap. (For ASCII program text files.) Opt Data = 1st four characters of selection file family name.
37	Same as Type 3, without wordwrap. (For ASCII program text files.) Opt Data = 1st four characters of selection file family name.
38	Same as Type 4, without wordwrap. (For ASCII program text files.) Opt Data = 1st four characters of selection file family name.

Partitioned File Display Group

TYPE	Function Performed
17	Display partitioned text file. Calls partitioned text prompts. Return here. Opt Data = 1st six characters of file family, then number of parts in file.
18	Same as Type 17, with download protocol option. Return here. Opt Data = 1st six characters of file family, then number of parts in file.
19	Same as Type 17, except return to MENU0000.CTL. Opt Data = 1st six characters of file family, then number of parts in file.
20	Same as Type 18, except return is to MENU0000.CTL. Opt Data = 1st six characters of file family, then number of parts in file.

Upload File Group

TYPE	Function Performed
27	Upload file with user-specified name, .UPL as .ext. No overwrites. Return here. No Opt Data.
28	Upload file with user-specified filespec. No overwrites. Return here. No Opt Data.
29	Upload file with user-specified filespec. Overwrites allowed. Return here. No Opt Data.
30	Upload file with user-specified partial name. You force from the first 1 to 6 characters of the file name, and the caller completes the name and extension. Overwrites allowed. Opt Data = 1st one to six characters of file name.

General System Function Group

TYPE	Function Performed
10	Terminate session. Calls TBBS logoff processing prompts. No Opt Data.
11	Display elapsed logon time. Return here. No Opt Data.
21	Exit TBBS to remote hosted DOS, without return: Recommended for online use by SYSOP only! No Opt Data.
22	List User Log. Display name, calling location, and last call time of up to the last 255 users. No Opt Data.
23	Chat mode. Invoke chat mode, by options specified in CEDIT. No Opt Data.
24	Select boards. Call TBBS prompts for user to select boards in combined display. No Opt Data.
25	Change password. Call TBBS prompts for password change. Return here. No Opt Data.
26	Change Xpert user level. Call TBBS prompts to select menu display level. No Opt Data.
31	Reconfigure terminal. Calls TBBS prompts for user to respecify his terminal options. No Opt Data.

General System Function Group (cont)

TYPE	Function Performed
32	Run a Q&A file, created by QAEDIT.COM. Return here. Opt Data = name of QAF file (without .QAF ext.)
43	Re-enter logon. Calls TBBS prompts for log-on sequence. Returns to calling menu after new user logs on. No Opt Data.
44	Exit to DOS, execute DOS command, return to TBBS (if modem carrier present.) Opt Data = DOS command string

Detailed Explanation of Command Types

There are seven groupings of command types, related within a group by their function. In the next portion of this chapter, we will identify each group, list the types (by number) which belong to that group, and explain their functions.

If you do not at first understand how a particular type operates, proceed -- understanding them individually becomes much easier after you have a broad view of the full system.

The differences among the various types within the groups depend upon such factors as whether a displayed file should be wordwrapped or not; whether download options are given or not; whether a message board is combinable with other boards or not; and so forth.

Each command type can be used as often as you choose, and as many times in each Menu Control File as you like. In fact, a few of them you will likely want to use several times on various menus.

We also provide samples of how each one is used, after the explanation of its particular group.

(Because the Type 32 command and its ancillary files are totally unique TBBS functions, they are dealt with in a Chapter of their own. See Chapter 5, USING QAEDIT.)

Menu Branching File Commands

The menu branching group controls the flow of system operation from one Menu Control File to another. By using only three command types, the SYSOP may create a simple or complex structure, up to 20 levels deep.

The structure may be tree-like or "threaded" or a combination of the two. By planning the structure, and by using a combination of privilege and A1/A2 flags, entire sub-systems may be built (and hidden) within your TBBS system. Sub-systems can be set up for subscribers of some particular system area, or for members of a users' group, or for business clients, or for...well, for however you want to define a group of your users.

We recommend that you draw a map, or organization chart, of how you want your system to function. But we're also aware that it's difficult for you to draw a map of your TBBS system until you become familiar with its full power and flexibility. We'll just say, sooner or later, you're going to wish you had a map of your system. We think such a diagram will be helpful, and you will know when you need one (which will be about :30 seconds after you get lost in your own system...) See Chapter 2 for an example map of the sample system provided on the release disk(s).

Here are the menu branching command types:

TYPE 5 - Transfer operation to another MENU????.CTL file.

Operates similarly to the GOSUB statement in BASIC programming, i.e. - it remembers from which menu it was sent so that it may RETURN to the sending menu. The Opt Data field identifies the Menu Control File to which operations are being transferred by filling in the ???? portion of the file name.

TYPE 45 - Transfer operation to another MENU????.CTL file.

Operates similarly to the GOTO statement in BASIC programming, i.e. - assumes new menu is now the upper menu on the stack. The Opt Data field identifies the Menu Control File to which operations are being transferred by filling in the ???? portion of the file name.

TYPE 12 - Transfer operation back to some higher menu level.

The Opt Data field, with a single ASCII character, determines how many levels "up" the transfer should go. This takes advantage of the TBBS ability to stack menus up

to 20 deep, while still remembering its path. A typical path would be a "1" to return to the immediate next higher menu level. A "0" will return control to MENU0000.CTL. A "2" would skip the calling menu and return to the menu that called it and so forth. (For this type, requiring a single ASCII character in the Opt Data field, the two-digit numbers 10-20 are represented by, in order, the characters :;<=>?@ABCD.)

Should you build a menu system that becomes too complex for TBBS to follow, you (or a user) will receive the error message "* MENU STACK OVERFLOW *". This indicates that the path from menu to menu has become more than 20 levels deep. After displaying the error message, TBBS will safely recover by returning to MENU0000.CTL -- but you should check your menu construction, and correct the path. (Your runtime log, DAYFILE.LST, should show the series of branches which caused the error.)

Now lets examine the branching commands in a fictitious system's menu 0000.

```
Listing of File MENU0000.CTL

TBBS #1 PRIME OPTIONS
=====================

Entry  1: U>ser Registration
        Priv=   0  Auth 1= X-------  Auth 2= --------
         Key=   U  Type=  5   Opt Data=HELP

Entry  2: O>nLine Magazine
        Priv=  25  Auth 1= ---X----  Auth 2= --------
         Key=   D  Type=  45  Opt Data=DATA


Entry  3: G>eneral BBS (PUBLIC)
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   G  Type=  5   Opt Data=DAIL

  ------------------------------------------------
                    .
                    .  (other non-branch entries)
                    .
  ------------------------------------------------

Entry  7: R>eturn to Prime Options
        Priv=  25  Auth 1= --X-----  Auth 2= --------
         Key=   G  Type=  12  Opt Data=0
```

In the above example, a user keying a U will next be shown the file of MENUHELP.CTL, for registration. The user keying a G will be selecting MENUDAIL.CTL for a BBS. A return from either menu will bring the user back to MENU0000.CTL, from which these have been excerpted.

However, if a user (with A1 flag set "---X----) keys O, it will bring up MENUDATA.CTL, which will then be the top command menu on the stack, and the user will be in OnLine Magazine.

The easiest way to send the user back to MENU0000.CTL from OnLine would be as shown in Entry 7.

*** Caution!!! ***

Each Menu Control File you create MUST provide some exit -- either a logoff termination or a transfer to another menu. Otherwise, users will be trapped at that menu. This is one of the most common mistakes made by a beginning TBBS menu constructor. Well, now we can say "told you so!"

The GOSUB-GOTO-RETURN characteristics of TBBS Menu Branching are also discussed in Advanced Techniques, Chapter 6.

Message Board Commands

Although there are only four message board function types, they are used repeatedly in Menu Control Files for each message board you define with CEDIT.

Each type calls up a TBBS system routine, which prompts the user for additional information on the particular function. There are also some slight differences in how the system handles access to EMAIL message boards and Standard message boards. We will point out the differences, after explaining the general function of each command type.

The Opt Data field for each type MUST contain the correct name of the Message Board for which that particular Menu Control Type function is intended. Or, if the Opt Data field is blank, the type function will pertain to all Message Boards to which the user is authorized access and which were defined as combinable.

(NOTE: The latter usage, leaving the Opt Data field blank, allows you to provide a Scan or Read "all" message boards command from one menu. It would access all boards for which the user has authorization, and which are defined as combinable with CEDIT. You might also use it for yourself as SYSOP, using a Leave message type which would allow you to choose on which board you want to leave a message, or a read all messages in the system to allow you to quickly review your message base. In anticipation of this need, any user who has a privilege of 255 in his USELROG entry is assumed to be the SYSOP and all boards become combinable for him regardless of their definition. He is also given ownership of all messages both public and private to aid in this review process.)

By your use of a combination of privilege and A1/A2 flag settings, it is possible for TBBS to display a particular message board command type to a user, yet NOT allow use of that function. The Menu Control File settings determine who SEES the Message Board's commands; but the Board's definition by CEDIT determines who may actually USE the Board.

Here are the message board command types:

TYPE 6 - Read messages.

This command calls up the following TBBS display:

```
<F>orward or <R>everse Multiple
<N>ew Messages
<M>arked Messages
<S>elective Retrieval
<I>ndividual Messages
<A>bort Retrieve

Which One?
```

If the user keys F or R, he is prompted to enter the number from which to start the multiple message read -- with an ENTER or carriage return indicating the first or last message in the file, respectively.

If the user keys N, the system displays all messages forward from when the user last accessed your system.

If the user keys M, the system will display those messages marked to be delivered to the user, either from the automatic "message waiting" check at logon or by the user having scanned and marked messages for reading.

If the user keys S, the system will ask for additional information by which to make the selection -- <F>rom, <T>o or <S>ubject. TBBS will then search the message file for appropriate matches to the selection data.

If the user keys I, the system will prompt for the number of message to retrieve and display it.

If the user keys A, the system will return him to the calling menu.

TYPE 7 - Leave message.

The system will prompt the user for entry of a message to the board listed in the Opt Data field, or -- if the field is blank -- prompt the user for a choice of message boards, then proceed with the entry routine. This routine operates as follows:

Unless this is a message being posted on the MSG TO SYSOP board (in which case the To field is forced to SYSOP) the first prompt will be:

Who is the message to?

After the caller enters the To field, TBBS will prompt:

What is the subject?

After the subject is entered, if the message is being posted to a standard message board which allows private messages, the next prompt will be:

Is the message private (Y/N)?

If the message being entered is public (i.e. not being posted to an E-MAIL type message board, or the MSG TO SYSOP board, or explicitly marked private by the previous prompt) and if the CEDIT message board options are set to request passwords on public messages, the next prompt sequence is:

Enter 1-8 character Password:

The user may ENTER or key a carriage return at this point, which will leave the password field for the message empty. If the user enters a password, the password will be required by another user (not SYSOP) to delete the message. This gives protection to users against malicious deletion of messages, if you should encounter that problem with your system. The password is not intended to guard against

reading of messages by other users; for such protection, you should advise your users to select EMAIL boards, or provide (within the CEDIT definition) for PRIVATE messages on your choice of message boards.

The message entry prompts continue:

```
Enter message text.
Max of 2048 Chars total, 132 max/line.
Type <CR> only to end input.

<L>ine or <B>lock entry mode?
```

The <L>ine mode will give the user an ongoing line count as the message is entered. The <B>lock mode is generally intended for messages uploaded from a file, prepared earlier -- but it may be used for direct input, as well. It will introduce its own carriage return at the end of 132 characters, which will later be passed over by the wordwrap function when the message is displayed to other users. If the user keys B, the system will additionally ask the user:

```
Do you want prompts (Y/N)?
```

If the user keys Y, the system will provide a ">" prompt every 132 characters or on receipt of a carriage return, and wait for additional input. If the user keys N, the system will turn off the echo function (the user will not see the text uploading, saving system time) and receive the message direct to disk, without prompting. For the latter to function correctly, the file must have been prepared offline to conform with all other "leave message" parameters -- but it is the speediest way for a user to upload a message online, after offline preparation.

TYPE 8 - Scan Messages.

This command can be used to rapidly scan the message headers on file, displaying information as determined by the Message Board Definitions in CEDIT. The system will ask if it should "Mark for Later Retrieval?" -- a Y will bring a pause after each header scanned, with the prompt "Mark? (Y/N)" so that marked messages may be later fully read with the <M>arked Message function of the Type 6 command. The system will prompt the user as follows:

<F>orward Scan
<R>everse Scan
<N>ew Message Scan
<S>elective Scan
<A>bort Scan

Which One?

If the user keys F or R, the system responds by asking from which number it should start the scan -- with an ENTER or carriage return indicating the first or last message in the file, respectively.

If the user keys N, the system displays all messages forward from when the user was last online.

If the user keys S, the system will ask, "Select <F>rom, <T>o, or <S>ubject?", by which it will search in making the scan.

If the user keys A, the system will return to the calling menu.

TYPE 9 - Delete message.

The system will prompt the user with the following lines, showing the numbers of messages on file:

```
The System has Msg's  712 to 1024
Enter number of message to delete:
```

When the user enters the number, TBBS will check to see if the caller is either the SYSOP, sender, or recipient of the message. If not, and there is a password on file with the message the system will send the following prompt:

```
Enter Msg Password:
```

The user must then enter the password correctly, or the system will return to the command menu and refuse to delete the message. If the user enters the password or if there is no password, any user with access privileges to the BBS may delete the message. TBBS will inform the user, "Message Deleted" and return to the command menu.

Note: The delete message command is the only use for the message password, so if you construct your system without using this command type, you will probably want to set the message board option which requests passwords on public messages to NO in CEDIT.

Here is an example of a short, simple, general user BBS with no privilege or A1/A2 flags:

```
Listing of File MENUDAIL.CTL

TBBS #1 DAILY BBS
=================


Entry  1: S>can Topics
          Priv=  25  Auth 1= --------  Auth 2= --------
           Key=   S  Type=  8   Opt Data=PUBLIC

Entry  2: R>ead Messages
          Priv=  25  Auth 1= --------  Auth 2= --------
           Key=   R  Type=  6   Opt Data=PUBLIC

Entry  3: L>eave Message
          Priv=  25  Auth 1= --------  Auth 2= --------
           Key=   L  Type=  7   Opt Data=PUBLIC

Entry  4: D>elete Message
          Priv=  25  Auth 1= --------  Auth 2= --------
           Key=   D  Type=  9   Opt Data=PUBLIC

Entry  5: G>o To Prime Menu
          Priv=   0  Auth 1= --------  Auth 2= --------
           Key=   G  Type= 12   Opt Data=0
```

In the example above, each of the Message Board Commands are used for a Standard BBS, as defined in CEDIT by the name PUBLIC. Note that the user has an exit -- to MENU0000.CTL.

```
** Authorization required to read this Board **
```

You may display a Message Board's presence, as above, but still control how and by whom it is accessed. In defining the Board with CEDIT, you may establish read and/or write privilege levels and A1/A2 settings that are more restrictive than those of the Menu Control File read or write entry. That allows the user to see the Board's commands, but TBBS will display an "Authorization required" message when an unqualified user attempts to access the Board. As example, if the CEDIT board definition carried a privilege level of 50 for the message board named PUBLIC, then users with a privilege of 25 would see the display above but not be able to use the Board itself.

EMAIL Message Boards

Message Boards defined by CEDIT with their type as EMAIL use the same Menu Control File commands, but operate slightly differently. Because they are sender-to-recipient private (and vice-versa), there is no multiple message read. When the user selects a Type 6 command from a menu, TBBS displays the following prompts:

```
Read Electronic Mail:
<T>o you
<F>rom you
<B>oth To and From you
<M>arked Mail
<A>bort Reading Mail
```

Keying a T will show all EMAIL addressed to the user, an F will show all EMAIL sent by the user, and a B will show both. An M, however, will only display the EMAIL not yet marked "rcvd" or that marked by the user during an EMAIL scan. Note: The SYSOP (any user with a privilege of 255) is assumed to "own" all messages, so the <B>oth option will display all messages on this email board for the SYSOP.

Because only the SYSOP, sender or the recipient can access EMAIL, there is no password protection, and either may delete a message.

How You Control the Message System

This is the message board power of TBBS as seen and accessed by your users. Later, when we introduce you to MFSQZ, the message file editor, you will learn the SYSOP has additional powers, as well. You will learn how to move a message from one board to another, print messages for your files, prepare a message off-line and upload it to your own system, delete a message board while retaining the messages, and many other functions.

In our next example we have displayed an example of a Menu Control File combining both a Standard and EMAIL Message Board on the same menu -- with a two major differences. The privilege level to write to the Standard board has been set to 100, so that only those users with comparable or higher privilege can write to the CLUB Message Board. The EMAIL portion of the same Menu Control File, meanwhile, is hidden from all users who have no A1 flag set as -X------. Users with a privilege level of only 25 may read the CLUB board but not write to it. This might make non-CLUB members a little envious of what they are missing!

```
Listing of File MENUMAIL.CTL

TBBS #1 CLUB MAIL
=================


Entry  1: S>can Topics
         Priv=  25  Auth 1= --------  Auth 2= --------
          Key=   S  Type=  8   Opt Data=CLUB

Entry  2: R>ead Messages
         Priv=  25  Auth 1= --------  Auth 2= --------
          Key=   R  Type=  6   Opt Data=CLUB

Entry  3: L>eave Message
         Priv= 100  Auth 1= --------  Auth 2= --------
          Key=   L  Type=  7   Opt Data=CLUB

Entry  4: D>elete Message
         Priv= 100  Auth 1= --------  Auth 2= --------
          Key=   D  Type=  9   Opt Data=CLUB

Entry  5: S>can EMAIL
         Priv=   0  Auth 1= -X------  Auth 2= --------
          Key=   S  Type=  8   Opt Data= EMAIL

Entry  6: R>ead EMAIL
         Priv=   0  Auth 1= -X------  Auth 2= --------
          Key=   R  Type=  6   Opt Data= EMAIL

Entry  7: L>eave EMAIL
         Priv=   0  Auth 1= -X------  Auth 2= --------
          Key=   L  Type=  7   Opt Data= EMAIL

Entry  8: D>elete EMAIL
         Priv=   0  Auth 1= -X------  Auth 2= --------
          Key=   D  Type=  9   Opt Data= EMAIL

Entry  9: T>erminate /Hang Up
         Priv=   0  Auth 1= --------  Auth 2= --------
          Key=   T  Type=  5   Opt Data=TERM

Entry 10: G>o To Prime Menu
         Priv=   0  Auth 1= --------  Auth 2= --------
          Key=   G  Type= 12   Opt Data=0
```

As with all Menu Control File displays in the manual, this is a "minimal" example. It will work, certainly -- but we hope you add your imagination and creativity to the use of MEDIT and TBBS.

Single File Display Commands

The file display groups make full use of the TBBS "wordwrap" power, as well as an internal binary-to-ASCII conversion function which can save you disk storage space.

The wordwrap algorithm (explained in Chapter 10) creates a formatted text output that will match a user's terminal display, once the terminal information is properly entered in the user log. It avoids broken lines and word breaks whenever possible, giving displayed text a natural appearance regardless of the user's terminal width. A wordwrap type command should be used for all general text display.

However, BASIC programs in ASCII for downloading should NOT use a wordwrap type, because it will introduce unwanted linefeeds or carriage returns into the file.

TYPE(s) 13 and 15

Display a file, with word wrap, no download options. Opt Data field contains the full filespec of the file to be displayed. This is the command type you would choose to display a bulletin file or a system information file not intended for downloading. Type 13 returns to current menu level; Type 15 returns to MENU0000.CTL.

TYPE(s) 14 and 16

Display a file, with word wrap, and download options. Opt Data field contains the full filespec of the file to be displayed. This is the command type you would choose to display an information file which you believe your users will want to save. Type 14 returns to current menu level; Type 16 returns to MENU0000.CTL.

TYPE(s) 39 and 41

Display a file, without word wrap, and no download options. Opt Data field contains the full filespec of the file to be displayed. You would use this command type to display a file with tabular material or other special formatting. Type 39 returns to current menu level; Type 41 returns to MENU0000.CTL.

TYPE(s) 40 and 42

Display a file, without wordwrap, and download options. Opt Data field contains the full filespec of the file to be displayed. This is the command type to use for your users to download text files of programs. If the file is in ASCII, it will be displayed in ASCII. If it is in binary (.COM file) or tokenized BASIC, it will be converted at runtime to hexadecimal ASCII. (Users will need to convert it back to its original form, using any of numerous available ASCII-to-binary conversion programs.)

Here is an example of use of the Single File Display command types:

```
Listing of File MENU0000.CTL

TBBS #1 PRIME OPTIONS
=====================

Entry  1: B>ulletins (1/1/83)
         Priv=   0  Auth 1= --------  Auth 2= --------
          Key=   B  Type= 13   Opt Data=SYSBULL.TXT

Entry  2: U>ser Registration
         Priv=   0  Auth 1= X-------  Auth 2= --------
          Key=   U  Type=  5   Opt Data=USER

Entry  3: G>eneral BBS (PUBLIC)
         Priv=  25  Auth 1= --------  Auth 2= --------
          Key=   G  Type=  5   Opt Data=DAIL

Entry  4: H>elp: User Notes
         Priv=   0  Auth 1= --------  Auth 2= --------
          Key=   H  Type= 14   Opt Data=HELP.TXT

Entry  5: T>erminate /Hang Up
         Priv=   0  Auth 1= --------  Auth 2= --------
          Key=   T  Type=  5   Opt Data=TERM
```

In the example, a user keying in an entry of B would display the ASCII file SYSBULL.TXT. An Entry of H would display HELP.TXT, and also request a download protocol for the user who wished to save the HELP file for later study. (See display file from selection list in the next section for download operation).

Display File from Selection List Commands

These command types function with a "menu" of sorts -- you well may refer to them that way in messages to and from your users. However, they are -- as named -- "selection lists" and do not operate like a TBBS Menu Control File.

Using a selection list allows you to designate a "family" of files for display. Let's name the "family" DOWN. The Opt Data field of your command menu file would contain DOWN and the selection list would be named DOWNM.TXT. Using your text processing program, you would write DOWNM.TXT, saving it in ASCII for system use.

As you prepare it, keep in mind that selection lists are ALWAYS displayed wordwrapped and without download protocol options. To prepare a formatted list will require study of the wordwrap algorithm in Chapter 8.

Selection lists may be as long as you like. DOWNM.TXT might look like this:

```
    DOWNLOAD MENU
=====================

-----TITLE/EXT-----Load Time

<1> FLIGHT.BAS     :  4:45 :T
 -- How much is honey?.
<2> ELECTRON.BAS   : 11:32 :T
 -- Coil/filter designer.
<3> MICROADV.BAS   :   :58 :T
 -- Adventure in 1293 bytes!
<4> SOUNDEMO.BAS   :  1:10 :T
 -- A new TRS-80 key word.

Type Selection or <M> for list,
P to set protocol, <CR> to exit:
```

The prompt at the bottom tells the user how to make a selection. Because the selection key may be up to four characters, this is not a single keystroke function -- users must ENTER the selection key. When TBBS reads the user's input character(s), it searches for a DOWN????.TXT file that matches, and displays it. The 1-4 input characters become the ???? portion of the file name.

Within your files, FLIGHT.BAS would be on disk as DOWN1.TXT, ELECTRON.BAS would be DOWN2.TXT, MICROADV.BAS would be DOWN3.TXT, and so forth, indicating that all the files are members of the DOWN family. When the user enters 1, TBBS seeks DOWN1.TXT and

displays it.

Your selection keys could be letters, so long as you make the files match. If the key to FLIGHT.BAS were AAA>, then you would name the file DOWNAAA.TXT.

(Should a user key characters for a nonexistent selection, TBBS will respond with "Data File Not Found" and return to the selection prompt. This does not produce a system error.)

The six command types below determine how files chosen from a selection list are displayed -- with or without download protocol options and with or without wordwrap. After TBBS displays the selected file, it will return to the selection prompt. A <CR> will return the user to a command menu, according to your choice of command type.

TYPE(s) 1 and 3

Display a file from the family of the selection list, with wordwrap and no download protocol options. Opt Data field contains the four-character family name. Type 1 returns to current menu; Type 3 returns to MENU0000.CTL.

TYPE(s) 2 and 4

Display a file from the family of the selection list, with wordwrap and download protocol options. Opt Data field contains the four-character family name. Type 2 returns to current menu; type 4 returns to MENU0000.CTL.

TYPE(s) 35 and 37

Display a file from the family of the selection list, without wordwrap and no download protocol options. Opt Data field contains the four-character family name. Type 35 returns to current menu; Type 37 returns to MENU0000.CTL.

TYPE(s) 36 and 38

Display a file from the family of the selection list, without wordwrap and with download protocol options. Opt Data field contains the four-character family name. Type 36 returns to current menu; Type 38 returns to MENU0000.CTL.

Now for an example usage of these functions. In our example the selection lists would be files named DOWNM.TXT and NEWSM.TXT. Note the difference in menu return routing.

```
Listing of File MENU0000.CTL

TBBS #1 LIBRARY
===============
Entry  5: A>ppleSoft Programs
       Priv=  25  Auth 1= --------  Auth 2= --------
        Key=   A  Type= 36   Opt Data=DOWN

Entry  9: L>ocal News
        Priv=  10  Auth 1= -------X  Auth 2= X-------
         Key=   L  Type=  2   Opt Data=NEWS
```

If the caller presses the A key, he will invoke Entry 5 in the above example. In that case, TBBS would construct a file name based on the Opt Data field as follows:

Take the Opt Data filed and append "M.TXT" to it to form the file name DOWNM.TXT.

This file is considered to be the selection list file and is opened and displayed to the user. It will always be displayed with word wrap and not downloaded, regardless of which command type in this group is chosen.

After displaying the selection list TBBS prompts with:

Type Selection or M for list,
P to set protocol, <CR> to exit:

This allows the user to select his file for downloading. If it is his first download, or if he enters P, then the following menu will appear:

The following protocols are available:

1 - ASCII with Buffer Control Codes
2 - ASCII only, no Control Codes
3 - CP/M MODEM protocol

Choose one:

The user may then choose which of the download protocol options he wishes to use. THe CP/M MODEM protocol option will automatically recognize either the Checksum or the CRC variations and operate accordingly. Once chosen, this protocol will remain in effect and the user will no longer be asked on a file selection which one he wishes to use for the remainder of the current call. To change, he must use the P selection.

When the user enters a 1 to 4 character selection code, the file name is formed by appending his selection code to the Opt Data filed and then appending a .TXT extension. As an example, if the caller enter a selection code of 1A, the file name formed would be: DOWN1A.TXT. This file is opened and transmitted to the user using the protocol he has selected. This name generation scheme means that all selection list files, and their selection lists form a "group" of like named files with the Opt Data as the leading characters and the extension of .TXT. If one of the ASCII protocols is used, and any of the first 64 characters of the file have bit 7 set (value greater than 127 decimal) then TBBS will automatically convert the file to a HEX-ASCII format on the fly.

If at the menu COMMAND? prompt, the user had pressed the L key to select the Local News, then the file NEWSM.TXT would be displayed as his selection list. After that display he would be prompted with the following:

Type Selection or M for list, <CR> to exit:

Notice, that he has no download protocol options. The selection mechanism is the same as described above, except that the file will always be displayed with word wrap and in ASCII.

Display Section(s) of Partitioned File

This file display type, like the selection list group, also functions with a "family" of files -- but it is intended to deal with long text files by conveniently dividing them into chapters. Because these types are intended for serialized text (not programs), they all operate with full wordwrap.

The display of these files does not function from a selection list, but directly from a Menu Control File display.

The family is given a six-character common name, followed by a two-digit number showing the maximum number of chapters. As example, you might have the family, "MYBOOK," which has 14 chapters. In the Opt Data field of the Menu Control File, you would enter MYBOOK14. On disk, you would have .TXT files:

```
MYBOOK01.TXT
MYBOOK02.TXT
MYBOOK03.TXT
and so forth, up through...
MYBOOK14.TXT
```

When a user selects this type of display, your TBBS will respond:

There are 14 parts. Which part do you want to start with?

You could have as many as 99 chapters to MYBOOK. TBBS will then scroll through all chapters of MYBOOK from whatever part the user selected as the starting point, to the end. The number of parts from which the user may choose will be the number added to the family name you entered in the Opt Data field; so, when chapters are added, that number must be updated.

If there are any missing chapters -- that is, if MYBOOK10 were not on disk -- then the chapter scroll would abort to the menu, with the message "Data File Not Found." This does not produce a system error, but will terminate the display function.

TYPE(s) 17 and 19

Display a series of parts from the partitioned file with wordwrap and no download protocol options. Opt Data field contains the six character family name with a two-digit number suffix showing the number of parts. Type 17 returns to current menu; Type 19 returns to MENU0000.CTL.

TYPE(s) 18 and 20

Display a series of parts from the partitioned file with wordwrap and with download protocol options. Opt Data field contains the six character family name with a two-digit number suffix showing the number of parts. Type 18 returns to current menu; Type 20 returns to MENU0000.CTL.

This is an extremely powerful display group. By a combination of one of these types with the Type 30 upload function, your TBBS can be the host of a user-written computer novel with your author(s) doing their own editing and file maintenance -- including the addition of chapters and the editing of earlier chapters.

Here's an example:

```
Listing of File MENUNOVE.CTL

TBBS #1 USER NOVEL
==================

Entry  9: R>EAD BOOK
        Priv=  50  Auth 1= X-------- Auth 2= --------
         Key=   R  Type= 17   Opt Data=MYBOOK14

Entry 10: A>DD CHAPTER
        Priv=  50  Auth 1= X-------  Auth 2= X-------
         Key=   A  Type= 30   Opt Data=MYBOOK
```

A user keying R from this menu would be prompted for the chapters available for reading, as explained earlier.

A user keying A would be shown the name "MYBOOK" from the Opt Data field, and prompted to "Complete file name.ext:" As one of your authors, the user could then enter 14.TXT, thereby producing the filespec MYBOOK14.TXT, and be allowed to overwrite the existing file.

Because it does allow overwriting, the Type 30 command must be used carefully -- note the A1/A2 flags above permit some to read the book, who cannot write to it. (The Upload File Group command types are explained in full next.) If a chapter number higher than that specified in the Opt Data field is added, the SYSOP will have to use MEDIT to change the number there before the added chapter will display. If you expect contributions to up the chapter number and don't wish to have to use MEDIT each time, you may place 99 in the "number of parts" position in the Opt Data field and this command will halt when the first missing chapter is encountered. No system error will result, but the message Data File Not Found will be printed to your caller.

Upload File Commands

There are four TBBS upload command types, varying by how much power you allow users to name files and overwrite files existing on the system. All uploads go to the drive specified in CEDIT's System Options section.

The real TBBS upload power is in the variety of data transmission protocols from which a user may choose to upload a file to your system. Let's first review the upload types, then look at the protocols in detail.

TYPE 27

User is prompted for 1-to-8 character name of file to be uploaded, and TBBS adds .UPL extension. If a file by that name already exists, user is prompted for another name -- thereby preventing overwrites. Opt Data field is blank.

Having .UPL as an extension can make it easier for the SYSOP to find uploaded files on a crowded disk directory.

TYPE 28

User is prompted for both name and extension. If the file exists, user is prompted for a different entry. Opt Data field is blank.

TYPE 29

User is prompted for both name and extension, and if the file exists on the system, it is overwritten. CAUTION: Any file could be overwritten, including your online runtime TBBS command files! Even the drive can be specified with this form of the upload command.

TYPE 30

The Opt Data field contains a 1-to 6 character name -- the first part of an existing or non-existing file name. The user is shown that name, and prompted to complete it. If the completed name is that of a file that exists, it is overwritten.

As noted earlier, this type has great flexibility in allowing users control over writing and editing of online .TXT files. Because the SYSOP controls the first part of the name through menu construction, this is a more restricted overwrite command than Type 29

NOTE: Users just becoming accustomed to uploading should be informed that TBBS writes the file name to a directory when the upload begins. If the upload is aborted, the filename will still be there -- hence, preventing a second attempt of a file of the same name for Types 27 and 28. The solution, of course, is for them to choose a different file name.

Upload Protocols

Whenever a user selects an upload command, this prompt is displayed:

```
Choose upload protocol:
1 - Prompted ASCII
2 - ASCII, XON after <CR> rcvd
3 - ASCII, XOFF/XON flow control
4 - CP/M MODEM Protocol - Checksum
5 - CP/M MODEM Protocol - CRC
6 - Help with protocols
```

The small help file called by 6 is online, part of TBBS. It will display this information:

```
TBBS allows the following upload protocols:
1 - Prompted ASCII.  Your terminal program must stop sending
after each line until it receives the prompt character ">".
2 - ASCII, XON after <CR>.  Your terminal program must stop
sending after each line until it receives the X-ON (Ctrl-Q)
character.
3 - ASCII, X-OFF/X-ON.  Your terminal program must stop
sending when it receives an X-OFF (Ctrl-s) until it receives
and X-ON.
4 - CP/M Modem protocol.  This is an error checking protocol
for direct sector transfers.
5 - CP/M Modem Protocol using CRC error checking.
```

After the user selects one of the options, your TBBS will wait for the appropriate response or handshake from the user's system. For the prompted mode ASCII upload protocols, the word "END" entered at a prompt concludes the upload.

At this point, the great variety of TBBS options must leave you and your users on your own. Obviously, the user must have some form of software or terminal emulator to meet one of the above broad options -- and there are dozens or more software packages or programs which will do the job.

We will point out, however, that the most accurate file transfer options available with TBBS are the CP/M MODEM protocols. Developed in 1977, and enhanced several times since, this system provides for TBBS and the user's system to compare checksums or CRC's of all bytes in each sector. The sector is sent again, up to ten times, if the checksum or CRC is not verified. The transfer is in binary, not ASCII, and because there is no conversion for screen display, this is the fastest file transfer process.

MODEM is a public domain protocol and is available in many public domain terminal programs. Although it was written for CP/M, programs for other non-CP/M systems have been developed to match its protocols. (No, your users do NOT need CP/M for file transfer.) The Protocols INFO text file on the TBBS master disk also has data communications information your users may find helpful for file transfer.

General System Function Commands

This group of command types allows users to access several TBBS housekeeping or utility functions. Only Type 44 requires an Opt Data field entry. The remainder are used with a blank Opt Data field.

Although some of these functions should be accessible only by the SYSOP or co-SYSOPS, you should consider providing all of them in one or more Menu Control Files -- if only for your own use.

As example, among the commands below are those a user will need to change a password, speed up the menu display, change terminal parameters, chat with SYSOP, and so forth.

TYPE 10

Terminate session. Logs off user, and re-initializes system as determined by CEDIT Options.

TYPE 11

Displays user's time on system during current call. (Call on-line clock is restarted by user with a Type 43 function. You may -- or may not -- wish your users to know that.)

TYPE 21

Close User Log and Message files, exit TBBS.COM to DOS, and leave TBBSDVR.COM active for remote hosting. This type (or Type 44) can be used by SYSOP for remote access to DOS for system maintenance.

So long as TBBSDVR.COM is not disturbed the system may be brought back up with the DOS command TBBS and the system will restart at MENU0000.CTL with all flags intact. System re-start takes about 15-30 seconds on an average TBBS.

TYPE 22

Display User Log. TBBS will read CALLERS.BBS, use it as an index to read USERLOG.BBS and display a list of all users by name, calling location, and date/time of last call. Listing is in reverse chronological order from the current user back. Up to the last 255 users will be listed.

TYPE 23

Chat with SYSOP. Allows user to summon SYSOP via a tone. Actual user screen procedure is determined by CEDIT System Options. During the time a user is in the "chat" mode, the on-line timeout clock effectively stops.

Note: the paging tone varies on different hardware configurations. For the IBM PC and workalikes, the internal speaker is used for the paging, on TRS-80's a tone is generated on the cassette port and also a bell char is sent to the printer. In CP/M systems, check the installation files for tone generation information.

TYPE 24

Select message boards. Allows user, during the current logon period, to select from authorized message boards for combined access. (See also, Message Board Command Group.)

TYPE 25

Change Password. Allows user to check current logon password, and revise it, if desired. If the user keys ENTER, the existing password remains unchanged. If the user enters a new password, the old password is discarded and the new one is placed in the User Log. However, the user cannot "change" from the existing password to the existing password -- because it will already be in the file.

TYPE 26

Select Menu Display Mode. TBBS offers users four menu display modes, 0-3, each ended by the system prompt, "Command: ?" Using this type function, the user has the option of choosing among these:

0 - Beginner: Displays full menu text line for each authorized option.
1 - Intermediate: Displays first word (up to 15 characters or space, whichever comes first) of each authorized menu text line.
2 - Expert: Displays first letter of first word of each

authorized menu text line.
3 - Super Expert: Displays only the system prompt (Command ?).

When the user keys a "?" at the system prompt, the full Beginner level menu will be displayed, regardless of the user's normal display mode level. If the user keys ENTER or a carriage return, the display mode will revert one level lower for that particular menu display.

TYPE 31

Reconfigure terminal. Places user into the computer terminal selection routine used at new user logon, so that user may revise the parameters shown for his terminal in the User Log file.

TYPE 43

Log on again. This type of function takes the user back to the beginning of the logon process, beginning with display of your Greeting Text.

The system will react to this function as though it has just answered an incoming modem carrier, checking the User Log, re-starting the on-line timeout clock, checking for messages, etc. This command will, however, return to the calling menu instead of loading MENU0000.CTL as a new logon normally does.

TYPE 44

Exit TBBS to DOS and perform a DOS command. The Opt Data field contains the command to be accomplished, which could range from SYSOP calling up a TBBS editor for remote maintenance, to running a program in BASIC for users.

The exact nature of the return to TBBS varies slightly from system type to system type as follows:

MS-DOS systems will automatically return to TBBS when the program being run executes an INT 20 exit to DOS command, or the SYSTEM exit from BASIC.

CP/M systems will automatically return to TBBS when the program being run executes a JMP 0000 or CTRL-C warm boot function, or the SYSTEM exit from BASIC.

TRS-80 systems must return by explicitly calling TBBS as a DOS command. An example is to exit a BASIC program under

NEWDOS/80 and return to TBBS use the CMD"S=TBBS" command.

While this remote hosting function of TBBS is very powerful and most programs can be successfully remote hosted in this fashion, It must be noted that security is slightly lower when you leave TBBS since the program being run now may have a flaw which could allow a determined board breaker to penetrate your system. TBBS cannot cover this, since you are now running your own programs. Just a note of caution.

Here is an example of a system utilities menu you might find useful to offer:

```
Listing of File MENUUTIL.CTL

TBBS #1 UTILITIES
=================

Entry  1: @>Terminal Options
        Priv=   0  Auth 1= --------  Auth 2= --------
         Key=   @  Type= 31   Opt Data=**none**

Entry  2: *>Chat With Sysop
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   *  Type= 23   Opt Data=**none**

Entry  3: P>assword Entry
        Priv=   0  Auth 1= --------  Auth 2= --------
         Key=   P  Type= 25   Opt Data=**none**

Entry  4: X>pert! Fast Menus
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   X  Type= 26   Opt Data=**none**

Entry  5: $>Time on Your Call
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   $  Type= 11   Opt Data=**none**

Entry  6: U>ser Log List
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   U  Type= 22   Opt Data=**none*

Entry  7: S>elect Mail BBS's
        Priv=  25  Auth 1= --------  Auth 2= --------
         Key=   S  Type= 24   Opt Data=**none**

Entry  8: L>og On Again
        Priv=  25  Auth 1= X-------  Auth 2= --------
         Key=   L  Type= 43   Opt Data=**none**
```

```
Entry  9: T>erminate /Hang Up
        Priv=   0  Auth 1= --------  Auth 2= --------
         Key=   T  Type=  5   Opt Data=TERM
```

USING MEDIT: Builder at Work

We've now completed the instruction phases on the actual construction of your TBBS system. With CEDIT and MEDIT (and a dozen or so MENU????.CTL files), you can -- as we said at the start -- literally build your own system. It probably won't look like any other TBBS, anywhere, and there's no reason it should. There is no final model, no absolute "right one" -- we hope you didn't expect that.

Be creative, and don't be afraid to experiment. At the same time, keep in mind your users are being "used" as guinea pigs if your experiments get a little far out. Maybe some of your creative effort is better spent off-line, before letting everyone see the experiment that didn't work -- while it didn't work.

You can, as example, develop a work disk loaded with a full TBBS system, message files, text files, download programs and so forth, for just your own use. It can be a testing system by taking your regular system off-line (phone off the hook, of course). Or, you may have the luxury of another computer system to use for practice.

Either way, TBBS does not need to be on-line to function with you as SYSOP (with the exception of the file transfer commands.) You can set and re-set privilege and A1/A2 flags, and log yourself on as a new user so you get the feel of your system from a user viewpoint. Try changing your terminal column width occasionally, to see how the menu displays appear to the narrow-screen users.

We suspect you will be constantly checking for ways to increase your system's "friendliness" to users, whether they be general public hobbyists or commercial clients. You're likely to find your system will always have the "under construction" sign posted, even if it's not visible.

That's the POWER of TBBS!

Use it!

FUNCTIONS OF QAEDIT

The system program QAEDIT (pronounced que a edit) is used to develop the question and answer control files which are in turn used by the type 32 Menu Control File command. These files allow you to easily design your own custon survey and voting functions, along with some other subtle control functions which will be explained in the Advanced Techniques section. You are given the option of changing the A1/A2 flags in a user's userlog entry after the question and answer file has been successfully run, so that you can achieve such effects as "one caller, one vote", staged registration procedures etc. We will now look at designing these functions in detail.

Designing Survey and Voting functions

One of the unique TBBS functions is its ability to take a survey of your users, or to set up an election by which they may cast a secret ballot for one of a list of items -- the "candidates" or "issues." These functions may also be made to set and/or clear any desired A1/A2 flags after they run allowing a variety of powerful effects, starting with the ability to have a vote only be taken one time. See advanced techniques for other capabilities.

The survey file function may be used for on-line merchandise ordering, or interactively accepting answers from users to any series of questions you ask -- with an option for them to validate and correct their answers. The voting file function has been used in one small city to stage a computerized "mock" municipal election, with the ballot taken from a sample of an official city election ballot.

Survey type files may be constructed to accept individual answers of up to 250 characters for each question. Voting type files accept a key stroke numerical entry, limited by the number of items from which the "voter" may choose.

How Question and Answer (.QAF) Files Function

Although the survey and voting functions use different forms of question files, both use the Type 32 Menu Control File command, and both are created and edited with the QAEDIT system program. Your TBBS system will recognize the difference between the two types of files, and operate properly in each case.

How TBBS uses names for QAEDIT files is significant. When you have completed writing a series of questions with QAEDIT, it will

ask you to supply a 1-to-8 character file name. QAEDIT will write the file to disk, appending an extension of .QAF to it. When you use the Type 32 command in a menu control file, you must enter the file name (but NOT the .QAF extension) in the Opt Data field. At runtime, as the menu option is selected and used, TBBS will create a .SVY file by the same file name (if it is a survey) or write the entries to the .QAF file (if it is a vote).

For a survey type .QAF function, the .SVY file is in ASCII, and may be typed or printed from DOS, or loaded into a word processing program for review. For a voting type .QAF function, the TBBS program VOTESUM will count the votes in the .QAF file and LIST or PRINT the results. The .QAF file, however, uses control codes which may only be created or edited by QAEDIT.

The Survey Function

A survey file is created by typing in each question and appending a single digit control code. It is the six different control codes -- called "terminator bytes" -- which tell TBBS how to handle each .QAF survey question. Here is a brief list:

0 - Display question, wait for answer, write answer to file or save it.
1 - Display question, wait for answer, save for verification.
2 - Display question, wait for Y/N, abort to menu on N.
3 - Display buffer contents, question, wait for Y/N, restart on N.
4 - Display question, wait for Y/N, restart on N.
5 - Display text line, proceed without answer.

How a Survey File is Constructed

The way each question in your survey file is handled is governed by the terminator bytes. With QAEDIT, you enter a question of up to 255 characters and key ENTER. (The question text will be wordwrapped at runtime. If you need additional lines during preparation, use the down arrow or line feed key). QAEDIT will then ask you to enter the terminator byte for that question. Here is how the bytes function:

0 - Display the question and accept an answer. If this question is in the file following a 1 terminator byte, save the answer in a buffer for later verification. If the question stands alone (not following a 1), then append the answer to the .SVY file.

(NOTE: All answers entered by users may not exceed 250 characters

and are terminated by ENTER or a carriage return. This means a user may not "edit" the answer at entry, except by backspacing. However, all answers may be re-entered as part of a verification procedure.)

1 - Display the question, accept an answer and begin a verify section. This answer, and answers to succeeding questions with a 0 terminator byte, will be saved in a buffer until verified. (The verify buffer has a limit of 2048 characters, and an overflow will cause the .QAF file to abort to the menu. It is risky to ask more than nine questions, without providing for verification.)

2 - Display the question, accept only a Y/N answer. Y will continue to the next question, N will abort the .QAF function back to the menu. This allows a bail-out for users who changed their minds about answering the questions.

3 - Display the contents of the verify buffer, then display this question, and accept only a Y/N answer. An N answer will clear the verify buffer, and return the user to the most recent byte 1 question, for re-entry of all answers. (This limited looping ability allows the user to continue entering and re-entering answers until satisfied they are correct.) A Y answer will append the contents of the verify buffer to the appropriate .SVY file.

4 - This operates like byte 3, but without displaying the verify buffer.

(NOTE: A Y answer to either a byte 3 or 4 question ends the verify process which started with a byte 1 question. A byte 0 question asked after the close of verification will append the answer directly to file. There is no further verification, unless a new loop is started with another byte 1 question.)

5 - The text line is displayed, with no answer expected. This byte allows you to write explanatory comments into the question file.

The contents of the verify buffer begin with the User Log name of the person taking the survey, plus the date and time of answer entry. For a survey on your TBBS system with answers of interest to your users, you may provide a runtime review of the full .SVY file by placing a file display type command in a Menu Control File. The Opt Data field would contain the appropriate XXXXXXXX.SVY name.

How a Voting File is Constructed

Voting files take a cumulative vote for a multiple choice question. Because there is no entry of the user's name, the vote is secret. TBBS adds within the Voting .QAF file as each vote is cast, maintaining a count in an accumulator byte (16 bit = 65535 total potential) for each item on the ballot. There is no Voting equivalent to the .SVY file since totals are kept in the .QAF file.

In preparing a Voting .QAF file with QAEDIT, you will likely want to defeat the TBBS wordwrap function. Your multiple choice list can be composed by using the down-arrow or line feed (CTRL-J) to change lines, so they look something like this:

> In what year do you think the first NFL-USFL football game will be played?
>
> 1. 1983
> 2. 1984
> 3. 1985
> 4. 1986
> 5. 1987
> 6. 2000
> 7. Never

Because TBBS expects only a single keystroke for a Voting file, you will need to plan your questions accordingly -- numbers will limit you to nine possible choices in a single question. But, by a succession of questions, you may stage an election for as many choices as you like. Remember also, that a Voting question is limited to 255 characters.

In creating a Voting file, QAEDIT will ask you for the number of answers -- this is number of choices you have in your multiple-choice question. If your ballot has only six items on it, the keying of "7" by a user will be disregarded by the system, and the ballot displayed again.

The utility program VOTESUM is used to compile a total of votes cast. For our ballot above, with a few test votes, here's the totals from the summary printout:

Summary of Results for Voting File: GAME.QAF

Results Summary:

1.	1
2.	5
3.	0
4.	6
5.	1
6.	2
7.	12

(For the record, we don't recommend this election. You might need to run it for quite a few years to get a winner.)

How to Use QAEDIT

When you bring up QAEDIT.CMD from DOS, it prompts you:

<1> Create a Q & A File
<2> Edit existing Q & A File
<3> Abort editing

Your Choice?

If you key 1, QAEDIT will ask if you intend to create a Survey or Voting type of .QAF file.

If you key 2, QAEDIT will further ask for the name of the .QAF file, tell you whether it is a Survey or Voting type, and load it into memory for editing. (The sample survey file ORDER.QAF, was supplied on your TBBS master disks. You may wish to use it for study and practice.)

If you Key a 3, QAEDIT will exit to DOS immediately and take no action.

QAEDIT operates similarly to the other TBBS editórs, from these menu options:

<1> List Questions
<2> Add a Question at the End
<3> Insert a Question
<4> Change a Question
<5> Delete a Question
<6> Change A1/A2 flags SET after Q & A
<7> Change A1/A2 flags RESET after Q & A
<8> Exit and Write .QAF File
<9> Abort Editing

Your Choice?

In starting a new file, choose 2, to add a question.

When you edit a Type 32 survey or voting command into a Menu Control File, the Opt Data field must contain only the name, NOT the .ext of the file (no .QAF or .SVY) -- TBBS needs the name alone to create or find the response file.

You also can provide your users a runtime review of any .QAF or survey), use a Type 39 command with the full filespec in the Opt Data field in the Menu Control File (yes, including .SVY, because the system will be displaying it as an ASCII file, without wordwrap.) For a .QAF file, use a Type 44 command, with the Opt Data Field containing VOTESUM (NAME) TBBS (with NAME,in parentheses, matching the name of the Type 32 file, without the extension .QAF). This will call VOTESUM; tell it the name of the file to total; send a display to the user and then call TBBS back. VOTESUM, of course, must be online for this use.

Note now the menu options 6 and 7 dealing with A1/A2 flags. You might ask yourself what they have to do with voting and survey files. Well they allow you to specify that certain A1 and/or A2 flags will be SET or RESET when this Q & A file is run to completion. This allows you to do some special effects such as preventing users from voting more than once. To do this, set an A1 or A2 flag of your choice on for all users. This may be done by using the gang set option in ULEDIT, and also setting this flag in the New User Defaults section the CEDIT logon options so new callers also have it set on. Then mark your menu entry which calls out the voting file with the same A1/A2 flag. In QAEDIT, indicate that the selected flag should be reset when the vote is taken. When the user takes the vote, the menu item will disappear and he cannot vote again!

The flag set/reset options may be used on either voting or survey files, and in the advanced techniques section you will see some interesting applications of them. Doubtless you can think of even more ways to use this feature of TBBS.

The overall question-and-answer function of TBBS can contribute great variety to your system. It can be used seriously, as already noted, as well as for user enjoyment by staging contests and opinion polls. The power of its function and the simplicity of its operation gives you the opportunity to offer users something "new and different" frequently -- we hope you take full advantage of it!

SETTING UP A REMOTELY MAINTAINED SECTION

Mel is one of your veteran users -- reads all the messages he can every day, helps out other users with advice and information, and really enjoys telecommunicating. Let's give him a sub-system of his own within your TBBS.

He'll have his own Message Board, with full authority to read and delete messages (public or private) on it. He will be able to upload files for others to read, and write new files to replace those he earlier placed online. Yet, in other areas of your system, he will be just an ordinary user -- the authority and power you give him remains totally under your control.

Here's what an MEDIT printout of Mel's primary menu (MENUMEL.CTL) would look like:

```
MEL'S TBBS SYSTEM
=================

Entry  1: I>nformation
         Priv=   0  Auth 1= --------  Auth 2= --------
          Key=   I  Type=  1   Opt Data=MELA
Entry  2: B>ulletins
         Priv=   0  Auth 1= --------  Auth 2= --------
          Key=   B  Type=  1   Opt Data=MELB
Entry  3: H>elp Files
         Priv=   0  Auth 1= --------  Auth 2= --------
          Key =  H  Type= 13   Opt Data=MELHELP.TXT
Entry  4: R>ead Messages
         Priv=   0  Auth 1= --------  Auth 2= --------
          Key=   R  Type=  6   Opt Data=MEL'S BOARD
Entry  5: L>eave Messages
         Priv=   0  Auth 1= --------  Auth 2= --------
          Key=   L  Type=  7   Opt Data=MEL'S BOARD
Entry  6: D>elete Messages
         Priv=   0  Auth 1= --------  Auth 2= --------
          Key=   D  Type=  9   Opt Data=MEL'S BOARD
Entry  7: S>can Messages
         Priv=   0  Auth 1= --------  Auth 2= --------
          Key=   S  Type=  8   Opt Data=MEL'S BOARD
Entry  8: E>xit Mel's TBBS
         Priv=   0  Auth 1= --------  Auth 2= --------
          Key=   E  Type= 12   Opt Data=1
Entry  9: #> Mel's Upload Command
         Priv=  50  Auth 1= --------  Auth 2= X-------
          Key=   #  Type= 30   Opt Data=MEL
```

Obviously, building Mel's sub-system requires more than just writing a menu. Always keep in mind that TBBS is a system -- a change in one area usually requires a complimentary step elsewhere, and maximum power comes from combining its features.

Here are the additional steps to build Mel's system to match MENUMEL.CTL:

We started this process in Chapter 3, when you were learning to use CEDIT. There, we had you create a Message Board which has not been used. Let's convert it for Mel.

Using CEDIT, review your Configuration File to recall the number of that Message Board.

Then, with CEDIT, choose Change a Message Board definition. Rename the board, MEL'S BOARD. Set the read and write privileges at 0, to conform with MENUMEL.CTL. Set the A1/A2 flags at all --------'s. Now, you can write the file back to disk and exit CEDIT -- but don't forget the number of the board!

Using ULEDIT, find Mel's entry in USERLOG.BBS. In the SYFGS item (25 flag positions, left-to-right, from 1), put an X at the correct position to conform to the number of MEL'S BOARD (this gives him message read/delete authority.) Also, in the A2 flags, make them look like this: X-------. (This gives Mel an uploading, overwrite-file authority with the Type 30 menu command.)

Now, using MEDIT.CTL, create MENUMEL.CTL (if you haven't done so already) and your system-editing work is done. It really is that easy!

Next, you need to explain to Mel how several menu command types work. By using the Type 30 upload command, he can write a Selection List (named MELAM.TXT) for the information files he'll write. And he can upload MELBM.TXT as a Selection List for his bulletins.

Mel can change any of those files from his own menu -- with Entry 9 being "hidden" from all other users by the privilege and flag settings, while other users may use the message area freely.

There is nothing fixed about the pattern of flags and privilege we have shown here -- choose your own, anyway you want. The important factor is to keep the system approach in mind...a change in one item, almost always means a change somewhere else.

The techniques above are minimal -- expand on them as far as you like!

Remote Operation of TBBS File Editors

Each of the TBBS File Editors may be operated remotely, if you have them online at the host system to be called up for use. Here's how. In one of your Menu Control Files, create a SYSOP-only menu with commands such as below:

```
Listing of File MENUEDIT.CTL

Entry  1: MEDIT
         Priv= 255  Auth 1= XXXXXXXX  Auth 2= XXXXXXXX
          Key=   M  Type= 44   Opt Data=MEDIT TBBS
Entry  2: ULEDIT
         Priv= 255  Auth 1= XXXXXXXX  Auth 2= XXXXXXXX
          Key=   U  Type= 44   Opt Data=ULEDIT TBBS
Entry  3: SQUEEZE
         Priv= 255  Auth 1= XXXXXXXX  Auth 2= XXXXXXXX
          Key=   S  Type= 44   Opt Data=MFSQZ TBBS
Entry  4: RECONFIGURE/NEW MENU
         Priv= 255  Auth 1= XXXXXXXX  Auth 2= XXXXXXXX
          Key=   R  Type= 44   Opt Data=CEDIT MEDIT TBBS
Entry  5: VOTESUM
         Priv= 255  Auth 1= XXXXXXXX  Auth 2= XXXXXXXX
          Key=   V  Type= 44   Opt Data=VOTESUM TBBS
Entry  6: + DOS
         Priv= 255  Auth 1= XXXXXXXX  Auth 2= XXXXXXXX
          Key=   +  Type= 21   Opt Data=**none**
Entry  7: DAYFILE
         Priv= 255  Auth 1= XXXXXXXX  Auth 2= XXXXXXXX
          Key=   D  Type= 13   Opt Data=DAYFILE.LST
```

Each of the editors called through the Type 44 command will function normally from your remote terminal, then exit to a TBBS warm start when you're through using it. Note that Entry 4 chains CEDIT and MEDIT together -- when CEDIT exits, it calls MEDIT. You may chain such a command of editors in the Opt Data field up to 64 characters long.

Entry 6 would allow you to access DOS remotely, and, so long as TBBSDVR is not disturbed, return to TBBS by merely calling it up.

Entry 7 would allow your remote reading of DAYFILE.LST, if it has been sent to a disk file with CONFIG.CTL System Option 3.

Note that on MS-DOS and CP/M systems, the drive number of the editors must also be specified in the Opt Data field if it is not the currently logged on drive.

The Ups and Downs of Menu Control Files

Menu nesting and flow can become somewhat confusing. The example here is an attempt to show the proper uses of the Type 5 (GOSUB), Type 45 (GOTO) and Type 12 (RETURN) commands within a system structure.

You may have already mapped out your system. If you haven't, we again suggest it. It's the only way to keep your perspective when doing actual menu construction.

The structure in this example will involve four menus. The top of this structure is assumed not to be MENU0000, but a menu which is already at least one level down in the system. The menus involved are MENUSECT, the top level for this section; MENUMSG with message board commands; MENUMAIL with E-MAIL commands; and MENUSYST, with some general system functions.

The goal is to allow the user the easiest path from message to mail and back, while also making the system menu available from both places -- with a return from the system menu to whichever other menu called it.

```
                              SECT---------------->--------------+
                               :                                 :
                               :                                 :
            +-----------------------------------+                :
            :                                   :                :
  (GOSUB)   :                                   : (GOSUB)        :
        MAIL <------------------------------> MSG               :
            :              (GOTO)               :                :
            :                                   :                :
            +-----------------------------------+                :
             (GOSUB)           :          (GOSUB)               :
                               :                     (GOSUB)    :
                             SYST <---------------<--------------+
```

The GOTO allows travel from MAIL to MSG and back, without causing the menu stack to be either pushed or popped. This means that a return from either will go up one level to SECT. The GOSUBs from SECT, MSG or MAIL to SYST will establish the return to the proper menu from SYST.

Here are the Menu Control File entries, by file, to accomplish the routing of SECT, MAIL, MSG and SYST as desired:

```
MENUSECT.CTL

Entry   : M>sg Section     Key=   M  Type=  5   Opt Data=MSG
Entry   : E>MAIL           Key=   E  Type=  5   Opt Data=MAIL
Entry   : S>ystem Help     Key=   S  Type=  5   Opt Data=SYST

MENUMSG.CTL

Entry   : E>MAIL           Key=   E  Type=  45  Opt Data=MAIL
Entry   : S>ystem Help     Key=   S  Type=  5   Opt Data=SYST
Entry   : R>eturn To Sect  Key=   I  Type=  12  Opt Data=1

MENUMAIL.CTL

Entry   : M>sg Section     Key=   M  Type=  45  Opt Data=MSG
Entry   : S>ystem Help     Key=   S  Type=  5   Opt Data=SYST
Entry   : R>eturn To Sect  Key=   I  Type=  12  Opt Data=1

MENUSYST.CTL

Entry   : E>xit Help       Key=   E  Type=  12  Opt Data=1
```

The structure here will allow travel to meet our goal. All return commands will return as expected, no matter how many times the path between MAIL and MSG is taken.

The simplicity of the Type 12 return is easy to use -- it will return to whatever menu called it.

The Type 5 and 45 commands, however, are likely to have you drawing diagrams to follow their potential paths.

Consistency in Menu Text

Most TBBS systems will contain several MENU????.CTL files. There are a few menu entries that will be found on nearly every menu in your system. By keeping the wording of these entries consistent, your users will find it easier to get around from menu to menu. It can get quite confusing if one Menu uses <T>erminate Call while another one uses <G>oodbye.

Here is an example of a menu file that contains all of the commonly used menu entries:

Listing of File MENUXXXX.CTL

```
Entry  1: <T>ime on System
         Priv= 25  Auth 1= ------- Auth 2= --------
          Key=  T  Type=  11   Opt Data=**none**

Entry  2: <C>hat ... Talk to SYSOP
         Priv= 50  Auth 1= ------- Auth 2= --------
          Key=  C  Type=  32   Opt Data=**none**

Entry  3: <E>xit ... Return to calling menu
         Priv=  0  Auth 1=-------- Auth 2= --------
          Key=  E  Type=  12   Opt Data=1

Entry  4: <G>oodbye ... Terminate Session
         Priv=  0  Auth 1=-------- Auth 2= --------
          Key=  G  Type=   5   Opt Data=TERM
```

When you want to create a new MENU????.CTL file you can load MENUXXXX.CTL first, add your additional menu entries, add the Menu Title Text and save the menu under the desired name. You can either insert your additional menu entries at the front or put them on the bottom. The example above only contains 4 menu entries but if you are creating several new menus, it gets old having to retype the same entries into each new menu.

Consistency can also include placement of the menu entries. The flow of the system will be much smoother if the commonly used entries occur in the same place from menu to menu. You can either place them at the top or the bottom of the menu, which ever you prefer.

Those Offensive Messages: One Answer

You now have a lot of friendly users on your system. Even full families -- husbands, wives, sons and daughters. You also have some GUNGA DIN's, KING KONG's and "good ol'" BILL BAUDY -- all of whom seem bent on trying to shock either you or your users with X-rated messages. Just what their real names may be is as obscure as their rationality.

There's really nothing you can do to prevent the input of such message traffic, except post a 24-hour-a-day editor at the host terminal. But you can frustrate the pseudonymous phonies, and make it less fun for them.

Here's a very deceptive Menu Control File listing:

```
TBBS #1 MESSAGES
================

Entry  1: S>can Topics
          Priv=  25  Auth 1= --------  Auth 2= --------
           Key=   S  Type=  8   Opt Data=PUBLIC
Entry  2: R>ead Messages
          Priv=  25  Auth 1= --------  Auth 2= --------
           Key=   R  Type=  6   Opt Data=PUBLIC
Entry  3: L>eave Message
          Priv=  25  Auth 1= --------  Auth 2= --------
           Key=   L  Type=  7   Opt Data=PUBLIC
Entry  4: D>elete Message
          Priv=  25  Auth 1= --------  Auth 2= --------
           Key=   D  Type=  9   Opt Data=PUBLIC
Entry  5: S>can Topics
          Priv=   0  Auth 1= -X------  Auth 2= --------
           Key=   S  Type=  8   Opt Data= PUBLIC
Entry  6: R>ead Messages
          Priv=   0  Auth 1= -X------  Auth 2= --------
           Key=   R  Type=  6   Opt Data= PUBLIC
Entry  7: L>eave Messages
          Priv=   0  Auth 1= -X------  Auth 2= --------
           Key=   L  Type=  7   Opt Data= PUBLIC
Entry  8: D>elete Message
          Priv=   0  Auth 1= -X------  Auth 2= --------
           Key=   D  Type=  9   Opt Data= PUBLIC
Entry  9: G>o To Prime Menu
          Priv=   0  Auth 1= --------  Auth 2= --------
           Key=   G  Type= 12   Opt Data=0
```

Look very closely at the menu listing. You'll see there are two message boards involved. One is named "PUBLIC" and the other is named " PUBLIC" -- (see the space in front of the P?)

You may also have noticed the differing privilege levels and A1 flag settings. If all your regular users have a privilege level higher than 0 (25 is shown, but it could another value), then it provides you the means to reduce privileges for some -- or never let them get very high. (CEDIT allows you to set the default values for new users.)

Here's how this menu works: When you get an objectionable message, don't just delete it. Use MFSQZ to move it to " PUBLIC" -- and reset the privilege and A1 level of the writer to compare with that board. Next time a user by that name logs on, all will

appear normal (just try spotting that extra space at a terminal), but the only messages accessible are those from your phonies.

Of course, after four or five scrolls of the message file, BILL BAUDY will know that something is not as usual. Just what, however, will keep him busy for quite awhile. Maybe he'll just get tired and go away. You may also salt his special message board with messages and text files which simulate system failures. The offender will then likely go away thinking you have a "messed up" board.

Again, the technique we're suggesting is minimal. You probably can easily think of refinements.

Private sections that can be advertised.

Maybe you are a member of the local computer club and you have graciously offered to provide a section of your BBS for club use. The club wants to put up its library of software for download but it wants to restrict it to members only. There is a simple way to meet the club's demands through duplicate menu items.

Take a look at the following sample club menu:

Listing of File MENUCLUB.CTL

```
Computer Club BBS Section
=========================

Entry   1: <R>ead Club messages
         Priv=  25  Auth 1= --------  Auth 2= --------
          Key=   R  Type=  6   Opt Data=**none**

Entry   2: <S>can Club messages
         Priv=  25  Auth 1= --------  Auth 2= --------
          Key=   S  Type=  8   Opt Data=**none**

Entry   3: <L>eave a Club message
         Priv=  50  Auth 1= --------  Auth 2= --------
          Key=   L  Type=  7   Opt Data=**none**

Entry   4: <K>ill a Club message
         Priv=  50  Auth 1= --------  Auth 2= --------
          Key=   K  Type=  9   Opt Data=**none**
```

```
Entry   5: <D>ownloads in Club Library
         Priv=  25  Auth 1= -X------  Auth 2= --------
          Key=   D  Type=  36  Opt Data=CLUB

Entry   6: <D>ownloads in Club Library
         Priv=  25  Auth 1= X-------  Auth 2= --------
          Key=   D  Type=  13  Opt Data=JOINCLUB.TXT

Entry   7: <G>oodbye ... Terminate Call
         Priv=  25  Auth 1= --------  Auth 2= --------
          Key=   G  Type=  5   Opt Data=TERM
```

In this example we have duplicate menu entry text for both 5 and 6. You will notice that the command type for each is radically different and that they have unique Auth 1 flags set.

Entry 5 is set up for club members and actually calls a download display list file named CLUBM.TXT. It requires a caller's A1 flag bit 2 be set on.

Entry 6 is for non members and has bit one of the A1 flag turned on. This menu entry will display a text file called JOINCLUB.TXT which could be information on how to join the club and obtain access to the club's library of files.

If you have added this new section to an existing system you will have to change the new user A1 flag bit one to be turned on (using CEDIT). Next you will need to use ULEDIT and set bit one on for every existing userlog entry. This can be quickly done using the command:

A = Alter A1/A2 for all users.

Then you can search for club members and modify their Auth 1 flag bit one to a '-' and bit two to a 'X'. Now your club members will have a private download section. They also have a way of attracting new members by letting non-members know that a special download section is available to club members.

ADVANCED MENU DISPLAY FORMATTING

There are a variety of ways to format the text which will be displayed to a caller on a menu. In this section, we will show you some examples of ways to tailor your menu text which are not immediately obvious. Remember that the purpose of formatting

your menu is to make the system as easy and natural for your caller to use as possible.

In most systems, there is a core group of commands which appear on every menu. These are such things as <G>oodbye, <C>hat, <U>tility menu callup and <T>ime on system etc. Often, these can clutter a menu and obscure the commands which are unique to a given area. A solution to this problem follows which makes use of the fact that if a menu entry has no text field, nothing will be displayed.

In this case, we put the common command text for the common commands in the menu title. Since the title is not displayed in the expert and super expert modes, and the KEY fields make up the expert command prompt, all will appear to be normal, but we have much more freedom to format the display.

```
TBBS Top Level Menu

<G>oodbye, <C>hat, <T>ime, or <U>tilities


Entry  1:
        Priv=  0  Auth 1= --------  Auth 2= --------
         Key=G    Type=  5   Opt Data=TERM

Entry  2:
        Priv=  0  Auth 1= --------  Auth 2= --------
         Key=C    Type= 23   Opt Data=**none**

Entry  3:
        Priv=  0  Auth 1= --------  Auth 2= --------
         Key=T    Type= 11   Opt Data=**none**

Entry  4:
        Priv=  0  Auth 1= --------  Auth 2= --------
         Key=U    Type=  5   Opt Data=UTIL .

        .... remainder of normal menu items
```

This could also be accomplished by placing the text for all of the commands in one menu text item, except that the intermediate mode would not display correctly. By placing the combined text in the title, all menu prompting levels will display correctly, and your menu will be formatted as you really want it.

Another Menu formatting trick which may be used is to add a blank line to your menu text using the down arrow to separate specific

groups of items in your menu display. This must be done with careful attention to the intermediate display however, or it will not be correct.

If you design menus which will not work correctly at higher expert levels of menu prompting, you may freely do so. If you do this, simply do not place a type 26 command (change expert level) anywhere in your system and a user will be forced to remain in the beginner level of menu prompting. This will allow you great freedom to construct highly formatted menus, even drawing pictures by combining the text of various commands. The important thing to remember in doing fancy menu formatting, is that the text of a menu entry does not have to describe that command. As long as the full screen describes all commands, the user will get the proper action from the system. You can even use the A1 flags to cause certain users to get different menu formats if you wish.

ADVANCED QAEDIT USES

The A1/A2 "bit toggling" features of QAEDIT allow many unexpected side uses other than straight survey and voting functions. This section will give a few examples of such usage.

A Staged Registration Technique

If you wish to run a fairly open system, but still wish to make new callers register to try to filter out abusers, here is one method. It forces the new caller to read a policies statement, and also to register with you, but does not force him to wait until you register him to have at least read access to your system. This allows long distance callers to explore your system on their first call, while still screening out the abusive callers. The secret to this technique is using the A1/A2 bit setting and clearing features of QAEDIT, along with recognizing that a survey type Q & A file may be composed of all comments which will cause it not to ask any questions or to create a .SVY type file, while still allowing A1/A2 manipulation. Also demonstrated here is one use of the fact that a menu control file's text entry may be used without actually being a command. In other words, there's a lot of food for thought in this example!

First, we will set a new user's A1 flag 1 on in the default section of CEDIT Logon Options as follows:

```
For New User: Priv = 25
              A1 = X-------
              A2 = --------
```

Then we create MENU0000.CTL according to the following rules:

1. All menu entries which comprise the "normal" top level menu will have A1 flag 3 set on in their MEDIT entries.

2. The <G>oodbye command has no A1 flags set, so that it will always be available to all users at all times.

3. The menu then will look like this:

```
Entry  1: You must read the <P>olicies of this system before
            you will see any more menu entries.
         Priv=  0  Auth 1= X-------  Auth 2= --------
          Key=?    Type=  0   Opt Data=**none**

Entry  2: <P>olicies of this system
         Priv=  0  Auth 1= X-------  Auth 2= --------
          Key=P    Type= 32   Opt Data=POLICY

Entry  3: If you now register completely you will be given
            immediate read only access to this system.
         Priv=  0  Auth 1= -X------  Auth 2= --------
          Key=?    Type=  0   Opt Data=**none**

Entry  4: <R>egister with the SYSOP
         Priv=  0  Auth 1= -X------  Auth 2= --------
          Key=R    Type= 32   Opt Data=REGISTER

Entry  5: <R>ead Msgs on the Bulletin Board
         Priv=  25 Auth 1= --X-----  Auth 2= --------
          Key=R    Type=  6   Opt Data=BULLETIN BOARD

       ... Remaining Normal system menu entries with Auth 1 Flag
       3 set in each one

Entry  8: <G>oodbye ... Terminate Session
         Priv=  0  Auth 1= --------  Auth 2= --------
          Key=G    Type=  5   Opt Data=TERM
```

Some notes about this menu. Goodbye has a Priv= 0 so that you can set a "turkey" user's privilege level to 0, and all of his A1 flags off, and then all he would see was the <G>oodbye menu entry when he called. Note that Entry 4 and 5 use the same KEY=R

letter. This is allowable because they have mutually exclusive A1 flags and will not ever both be active at the same time. This requires you to set the A1 flag 2 bit off in the SYSOP userlog entry, or else both of these entries will display for you. If they both display, TBBS will make only the first one active in response to an R command. You will probably also want to set off the A1 flag 1 for your SYSOP log entry as well, so that the <P>olicies option is suppressed.

The two menu entries 1 and 3 are used only so their text fields will display conditional messages to a user in his menu. The Key field was set to ? which is a reserved character and therefore cannot be activated. The Type field was set to 0 to indicate it would not be used. Anything could actually have been put here for type, since the command cannot be activated.

Now let's take a look at the two QAEDIT files named POLICY.QAF and REGISTER.QAF which perform the "magic" in this example.

First, the policies file. This file consists of only comment type "questions", which have a terminator byte of 5 (see QAEDIT, Chapter 5 for full details on Q & A file construction). A sample QAEDIT listing of this file might look like:

```
Text of Question # 1

Thou shalt not be a turkey

   << Terminator Byte  5 >>

Text of Question # 2

Thou shalt not take the name of TBBS in vain

   << Terminator Byte  5 >>
```

Obviously, you would want your own systems policies stated here! This file is an all question, and no answer Q & A file. The user cannot abort the listing, and can only pause it using Ctrl-S. He must, therefore, read it all once he starts. This file also is set to CLEAR A1 flag 1, and SET A1 flag 2 after it has run, using the QAEDIT menu options 6 and 7 for A1/A2 flag alteration.

Next we look at the Q & A file REGISTER.QAF. This file will actually ask the user for information about himself, which hopefully will make him less likely to become a system abuser. A QAEDIT listing of this file might look like:

```
Text of Question # 1:

**NOTE** Failure to completely answer all questions
in this registration survey will disallow full use
of this system!!!

  << Terminator Byte  5 >>

Text of Question # 2:

What is your full name (First Last)

  << Terminator Byte  1 >>

Text of Question # 3:

Where are you calling from (City, ST)

  << Terminator Byte  0 >>

Text of Question # 4:

What is your telephone number (XXX-XXX-XXXX)

  << Terminator Byte  0 >>

Text of Question # 5:

What type of computer/terminal are you using?

  << Terminator Byte  0 >>

Text of Question # 6:

I have your information as:

  << Terminator Byte  5 >>

Text of Question # 8:


Is this correct?
  << Terminator Byte  3 >>
```

This QAEDIT file will build a file named REGISTER.SVY with the registration information of each user appended to it. We also set this file to turn A1 flag 2 off, and A1 flag 3 on, so that the normal menu appears when they return from registering. Asking for the full name here allows users who logged on with a pseudonym to give you their full name and address. You can then

decide if you wish to allow pseudonyms on your system, and register them accordingly. It also helps correlate a user who is trying to log on a several different names for reasons of his own.

Now lets take a look at how all of this looks to the new user when he calls your system:

The first time he logs on, his top level menu will show as follows:

```
Title of Top Level Menu

You must read the <P>olicies of this system before
you will see any more menu entries.
<P>olicies of this system
<G>oodbye ... Terminate Session

Command?
```

If he selects the <P>olicies command, then the text of the comment questions in the file POLICY.QAF will be displayed to him and when he returns to the top level menu it will look like:

```
Title of Top Level Menu

If you now register completely you will be given
immediate read only access to this system.
<R>egister with the SYSOP
<G>oodbye ... Terminate Session
```

If he now selects the <R>egister command, then the Registration QAEDIT file (REGISTER.QAF) will run and record his answers to the file REGISTER.SVY. When he returns to the top level menu, it will look like this:

```
Title of Top Level Menu

<R>ead Msgs on the Bulletin Board

        ... Remaining Normal system menu entries with Auth 1 Flag
        3 set in each one

<G>oodbye ... Terminate Session
```

He will have a privilege level of 25, since that is what we set in the CEDIT new user defaults and if message boards are defined

as requiring a privilege of more than 25 to write, he will get the ** Not authorized to write on this Board ** error message when he tries to leave messages until you alter his privilege using ULEDIT. You would do this after reviewing the REGISTER.SVY file and deciding he had given good information.

Some easy ways of verifying his information for consistency is to see if the city he is calling from has the area code of the telephone number he left. Total verification is only possible by calling the telephone number and checking the user out. This is rarely necessary, as most turkeys trip themselves up by the time they have registered. How closely you wish to control your system is up to you. It is unfortunate that a few trouble makers require SYSOPs to even think of running anything but fully open systems, but it is a fact of life.

This technique has many applications other than registration. As an example it can be used to implement Adventure Game techniques using only menus and QAEDIT files. Here is another glimpse of some of the power of TBBS...

A piece of Colossal Cave

Let's assume that you used A1 flags 1 and 2 to represent the famous ROD in the original adventure game. A1(1) is set if you are holding the ROD and A1(2) is set if you are not. A1(3) is set if you are holding the cage and A1(4) is set if you are not. Now you enter the room (containing the bird) whose MEDIT menu list looks like this:

```
Title: This is the east room, a bird is flying around.

Entry  1: <T>ake Bird
         Priv=  0  Auth 1= -XX--X--  Auth 2= --------
          Key=T    Type=32    Opt Data=TAKEBIRD

Entry  2: <T>ake Bird
         Priv=  0  Auth 1= X-X--X--  Auth 2= --------
          Key=T    Type=13    Opt Data=BIRDFLEW.TXT

Entry  3: <T>ake Bird
         Priv=  0  Auth 1= ---X-X--  Auth 2= --------
          Key=T    Type=13    Opt Data=WITHWHAT.TXT

Entry  4: <D>rop Bird
         Priv=  0  Auth 1= ----X---  Auth 2= --------
          Key=D    Type=32    Opt Data=DROPBIRD
```

```
Entry  5: <N>orth
        Priv=  0  Auth 1= --------  Auth 2= --------
         Key=N    Type=45    Opt Data=RM12

       .... Other entries as needed

Entry 30: <Q>uit .. Get me out of Here!!
        Priv=  0  Auth 1= --------  Auth 2= --------
         Key=Q    Type=12    Opt Data=0
```

Now its explanation time. I hope you don't have a headache now, because you will when we're done here. The QAEDIT files TAKEBIRD and DROPBIRD are the all comment type we discussed in the registration example above. They have text indicating in purple prose that the bird has been put in the cage, or released from the cage. They also are set with QAEDIT to switch the Al flags 5 and 6 to represent that you have the bird (Al(5) on, Al(6) off) or do not have the bird (Al(5) off, Al(6) on). If we now construct a truth table to show the possible states of the first 6 Al flags it would look like this:

Al(1)	Al(2)	Al(3)	Al(4)	Al(5)	Al(6)	Description
Off	On	Off	On	Off	On	You do not have anything
On	Off	Off	On	Off	On	You have only the ROD
On	Off	On	Off	Off	On	You have the ROD and Cage
Off	On	On	Off	Off	On	You have only the Cage
--	--	--	--	On	Off	You have the bird

The -- entries above mean "don't care" since when you have the bird, the <T>ake bird command should disappear.

Now if you match up the truth table with the menu entries, you'll see that if you are holding nothing, the Entry 3 version of the <T>ake Bird command will be active. This will display a text file which will indicate you have nothing with which to take the bird. If you are holding only the ROD, the same entry will still be active. If you are holding the cage and the rod, then the Entry 2 version of this command would be active, and the text file displayed would indicate that something was scaring the bird away. However, if you have the cage, and don't have the ROD, the QAEDIT file named TAKEBIRD.QAF would be run setting Al(5) on and Al(6) off which indicates you are holding the bird in the cage. This will cause all versions of the <T>ake command to disappear and the <D>rop bird command to appear in its place. Since the Al flag settings become a part of the userlog entry, a partially completed adventure can be resumed with inventory intact.

You can add much more to this, such as condition specific menu

text which displays according to inventory, direction commands which act differently according to what you have done before, and even an inventory command based on condition menu text and the current A1 flag settings. But we'll leave all that fun for you!

Disk Space Management for Efficiency

Operation of your TBBS system will require your close attention to disk space management, especially if you are using floppy disks on 40-track drives. In the past, you may have left this matter entirely to the DOS -- however, now is the time to better acquaint yourself with how your DOS manages files, and do some management of your own.

Each of the DOS's used by TBBS has a lowest level of disk allocation. For MS-DOS systems, it is the cluster. This is two 512 bytes sectors or 1K on a floppy disk. For CP/M 2.2, the basic allocation unit is the block, and may be either 1K, 2K, or 4K depending on the total size of the disk involved. The lowest level of space allocation for TRSDOS-standard systems is a gran -- which may be either five or six sectors depending on the DOS and whether the disk is single or double density.

In general terms, once the last previously-allocated basic unit of a file has been filled, the DOS will write additional file sectors to the lowest-number unallocated basic unit on the disk. If that unit happens to not be contiguous or adjacent to the units previously comprising the file, then DOS builds "extents" to the file. This can result in a file jumping from, say, sector 590 (near the inner edge of the disk) to sector 30 (near the outer edge of the disk.)

That, in turn, causes your drive head and stepper mechanism to be put to a real workout, if a file grows to four or five -- or more -- extents.

Three files written to by TBBS -- the .BBS files -- will grow as new users join your files, and new messages are written. The ideal layout (which we expect few TBBS SYSOPs will have the luxury of matching), would be to have each .BBS file on a separate drive, as the last file on the disk. Then, new units would always be allocated to the file's previously-allocated units in an orderly and contiguous manner.

On any system with two or all three of the .BBS files on the same drive, normal user activity will have the DOS basic unit

allocation routine skipping from one file to another. So, the DOS will be adding extents to the files frequently. To minimize drive wear, you should have a goal of renewing these files on a regular basis to minimize your drive stress. And if possible, place MSG.BBS and MSGHDR.BBS on different physical drives, as these two files will "intertwine" allocations the most quickly.

MFSQZ and ULEDIT both provide file backup routines whose functions will minimize extents to a file. They should be used regularly, even if you have ample room left on a disk. And, just as with any software that writes frequently to disk, it is wise to occasionally totally renew your operating disks -- re-format them, and refresh your files from a backup disk. An occasional investment in new floppy diskettes will pay off in fewer disk related difficulties during operation.

If you want to provide for maximum efficiency, studying your file layout with a disk monitor program might help you improve on the arbitrary placement of files by the DOS. You can create "filler" files of measured lengths to force the placement of a TBBS file where you want it; then, kill off the filler files. Such precautions can be tedious to perform, but they can prolong the life of your disks and disk drives.

If you are running TBBS on a very small amount of disk space, you will likely have to have a separate set of editing disks for such programs as MEDIT, CEDIT, MFSQZ, ULEDIT etc. This means that you will need an entire disk for MFSQZ and the .BAK files it produces. Unfortunately this will force you to have both the MSG.BBS and MSGHDR.BBS files on one disk when you run, and live with the extra disk wear and tear.

When you are running TBBS on a very small amount of disk space, you will not be able to "on line" SYSOP your system. If you have moved your editors to separate editing disks, there is not much you will be able to do remotely.

The Message File Editor - MFSQZ

The message file editor MFSQZ (pronounced em ef squeeze) is the utility program which you use to maintain your TBBS message base. It is used for the following functions:

- It squeezes out deleted messages to reclaim disk space used by them.
- It allows you to edit the message headers and change such items as the from, to and subject text, move a message from one message board to another, change the private public status, the deleted and received status etc.
- It allows you to archive deleted messages before they are actually removed. This archiving may be to a disk file which is a standard ASCII text file, or to a printer listing. It will also allow you to archive messages by message board, or chronological grouping.
- It allows you to edit the main text of a given message, or to replace it with totally new text of any size.
- It allows you to renumber your message base when you wish to do so.

How to Use the Message File Editor

The TBBS message system is contained in two files -- MSG.BBS and MSGHDR.BBS. MSG.BBS is a continuous, sequential file which contains the text of all the messages posted on all the message boards in your system. Each message is stored in ASCII. (For the technically minded among you, a 00 byte separates each message from the next in the disk file. Full technical details of the message base record structure may be found in the technical reference section.)

MSG.BBS can rapidly become the largest file you have, because it grows with each additional message. Although a user or you may mark a message deleted, it nevertheless remains in the file until you actually remove it with MFSQZ.COM. This is to allow you to decide to undelete a message, archive it before deleting it, or just to pitch it away as you choose. While this feature gives you a lot of flexibility and error recovery capability, the price it exacts is the need for a "squeeze" utility to finally actually remove the unwanted messages.

MSGHDR.BBS is the real workhorse of the message file system. It has a header for each message, containing all the file information about the message. When you edit your message file, you are actually editing the headers, because they control the file.

MFSQZ is fully menu-driven, with a runtime display of its commands always available as you are editing. When you call MFSQZ from DOS, the initial menu gives you these options:

<1> Squeeze Message Base
<2> Edit Message Headers
<3> Abort MFSQZ

Your Choice?

Keying 1 (Squeeze Message Base) will take you immediately to the squeeze utility which removes messages previously marked for deletion from the file, allowing you to reclaim the disk space for new messages or other use. The "squeeze" operates by reading the message headers, and skipping those headers and messages marked for deletion and copying the remaining messages and headers to the files MSG.BAK and MSGHDR.BAK. For your files' protection, it then verifies the disk write of the backup. After it has completed revision of the files, MFSQZ kills the original MSG.BBS and MSGHDR.BBS files, then copies the .BAK files back to the original disk as .BBS files. (The .BAK files are saved for your disposition, and use as a defense against catastrophic file loss due to hardware trouble.)

During the squeeze operation, you will also be given the choice of renumbering your message base should you choose to do so. This renumbering occurs before the message base is actually squeezed, so if you want to end up with full continuity in your new numbers (none missing in the middle) then you will need to squeeze once without renumbering to remove all deleted messages, and the squeeze again with renumbering. Also, when you are renumbering, the file USERLOG.BBS must be online, since the "Last Message Read" field in it must be updated to reflect the new numbering system. Also, you should save backups of all three of these files after renumbering so that you can restore a consistent system if you have a fatal disk crash.

When you are using the "squeeze" mode, MFSQZ will ask for

the drive number on which to place each .BAK file. We recommend this be on a drive other than that which holds your original files, so that drive head motion will not be excessive, and protection against hardware failure is maximum. MFSQZ will also ask if it should, "Print message text during squeeze?" -- an option which will allow you to rapidly read your edited message file as MFSQZ copies it.

A STRONG note of caution: Although MFSQZ.COM has a built-in verifying stage, even the most reliable hardware does have occasional disk I/O errors. For maximum safety of your message base, we recommend you make a VERIFY'ed copy of MSG.BBS and MSGHDR.BBS on a backup disk prior to "squeezing" the files. Taking 4-5 minutes for such a precaution can save four or five hours work trying to re-construct a message file that has been clobbered with a parity error!

Keying 2 (Edit Message Headers) from the MFSQZ.COM initial menu, will bring you this larger menu:

```
<1> Review Deleted Msg Headers
<2> Review all Msg Headers
<3> Review all Headers on one Board
<4> Review all Rcvd message headers
<5> Print all messages
<6> Print deleted messages
<7> Print all msgs on one board
<8> Exit and do squeeze
<9> Exit Editing

Your Choice?
```

Print options 5, 6, and 7 will allow you to send your choice of messages to the system printer, or archive them to a disk file. The print options will ask you for a starting message number, and whether you wish the output to go to printer or disk. If you select disk, then you will be asked for a file name. All qualifying messages from the specified number (or next higher number if the specified number does not exist) will be either printed, or archived to an ASCII text file which is suitable for most word processors or editors.

Review options 1, 2, 3, and 4 will allow you to scan the headers of the messages appropriate to that option. As you are reviewing headers, you will have available to you a number of commands with which to edit or revise the message files, as shown here:

Header Edit Command Summary:

```
    ; = Advance to next message       - = Back up one message
    G = Goto a message number (or next higher qualifying msg)
    L = List this msg to printer      T = Display text of message
    D = Toggle Msg Deleted status     P = Toggle Priv/Pub msg status
    R = Toggle Rcvd Msg status        B = Assign board number to msg
    E = Edit the text of the To, From, or Subject fields

*   S = Save msg to EDLIN file        M = msg text from EDLIN file
    Q = Quit the review               <Enter> = exit menu

    Which one?
```

* Note: For CP/M systems this option will use ED compatible files and for TRS-80 systems the files will be ASCII files which are compatible with the NOBUG text editor or most TRS-80 word processors such as Electric Pencil.

The options from the Command Summary are available at any time you are reviewing the file. The display itself is available by keying ? while a header is on your screen. Keying Q returns you to the MFSQZ edit menu. Those options marked "toggle" change the status of a message back-and-forth as you prefer. The options operate as follows:

; = Advance to next message:

This option will advance the header display forward one message. If there are no more messages in your base, it will return to the edit menu.

- = Back up one message

This option will back the header display up one message. If you are now at the first message in your message base, this command will return to the menu.

G = Goto a message number (or next higher qualifying message)

This option will prompt you for a message number. It will then position you to the header which has that number, or the next header above the specified number if that header does not exist or qualify. If there is no message at or above the specified number, you will be returned to the edit menu. A Goto 0 command will always get you to the first qualifying message in the message base.

L = List this message to printer

This option will list the message and its header to the printer. Be sure your printer is online and ready before giving this command, as otherwise MFSQZ can hang on printer I/O.

T = Display text of message

You are normally looking at just the message headers during any review operation. This command allows you to view the text of the corresponding message. The text is printed as it was entered and is not word wrapped here as it will be when it is displayed by TBBS.

D = Toggle Msg Deleted Status

This command allows you to delete or undelete a message. It may be cycled back and forth as many times as you wish.

P = Toggle Priv/Pub msg status

This command allows you to make a message private or public. It may be cycled back and forth as many times as you wish. Pressing the key reverses the current status of the message.

R = Toggle Rcvd Msg status

This command allows you to mark messages as received, or to remove that marking. It may be cycled back and forth as many times as you wish.

B = Assign board number to msg

This command allows you to move a message from the board it is currently posted on to any other board you wish. When you press this key you will be asked:

Board Number (1-25, 0 for names)?

If you already know the board number you wish to post the message to, just enter it. If you have forgotten, just enter 0 for a list of the board numbers and names. The question will be repeated and you may then enter the number you wish from the name list.

E = Edit the text of the To, From, or Subject fields

This command allows you to edit these fields to make any corrections or changes you wish. If you change the To

field, and remove any (Rcvd) markings using the R command, the newly named message recipient will be notified on his next logon that the message is waiting. When you press this key you will be asked the following:

Edit <F>rom, <T>o, or <S>ubject?

Press the correct key and you will be asked:

Enter New Text:

Type in the new text desired for the entry and press ENTER to make your change.

S = Save msg to EDLIN file

This command will save the message to a file which is EDLIN compatible and thus may be processed by most word processors. You can use this option to extract the text of a message for correction, editing, or expansion. You will be asked:

Filespec?

Enter the name of the file you wish to hold the saved text.

M = msg text from EDLIN file

This command will replace the text of the current header's message with the text from the specified EDLIN compatible file. This replacement text may be as large as you wish and is not limited to the normal 2048 characters that messages entered by TBBS are. If used in conjunction with the S command above, a message's text may be edited. You will be asked:

Filespec?

Enter the name of the file which contains the new text for the message. Note: It is strongly recommended that you squeeze after Merging many messages into the base, as this process will fragment your MSG.BBS file and make message retrieval somewhat slower. A squeeze operation re-aligns the text in the most efficient manner.

<Enter> = exit menu

This command will cause the current header to be redisplayed.

We cannot guide you through a practice session with MFSQZ, until you have an actual message base with which to work. You may wish to bring up your TBBS system, and write some messages to yourself to create a practice base ... or wait until your on-line system has accumulated a few messages for you to review. Use a backup for practice! Once your users grow accustomed to the ease of message entry and use with TBBS, you will find that even the very large message capacity of TBBS is quickly strained. This means that you will be using MFSQZ to remove messages for space fairly frequently to avoid filling your diskette space.

NOTE: Your message files, together with USERLOG.BBS, can rapidly fill a 40-track disk on a busy TBBS system. You need to be watchful that your files do not become unmanageable. Please read, "Disk Space Management" in Advanced Techniques, Chapter 6.

A further note on disk utilization. MSG.BBS and MSGHDR.BBS are generally the most heavily used files in a TBBS system. Since they are used concurrently, it is best to place them on different drives if possible. This minimizes the head movement while reading and saving messages, and greatly reduces general disk drive wear and tear. It also allows less fragmented allocation of the disk and correspondingly better performance. Good disk drive maintenance is essential to trouble free TBBS operation.

THE FUNCTIONS OF ULEDIT

The maintenance utility program ULEDIT (pronounced you el edit) is used to maintain the USERLOG.BBS file. This file contains a record for each user of your system. This record contains the user's terminal configuration, his authorization time limit and privilege, his logon password (if he has one) and statistics about his system usage.

The ULEDIT program allows you to edit the userlog and update each individual's record as needed. In addition, it allows you to delete records entirely, and create backups of the log. You can also do global clearing of the News file read flag (for when the news file changes), Global setting or clearing of A1/A2 flags (for when you change the usage of those flags in your system), and also you can change the caller number for your system and add a new user to the log.

How to Use the User Log Editor

You may find yourself using ULEDIT on nearly a daily basis, if yours is a busy TBBS system. There will be entries from new users to review, or helping the user who's forgotten a password (there's always one!), or re-setting the read flags for your SIGNON2.TXT when you update the news or any of several other tasks.

The User Log Editor, ULEDIT, is used to maintain and edit entries in USERLOG.BBS, where all the parameter information about each of your users is filed. As a new user logs on, TBBS writes a new entry to USERLOG.BBS (if you allow new users on your system in CEDIT), and, at each ensuing logon by that user, TBBS writes an ongoing record of their activity to USERLOG.BBS. You may also create new USERLOG.BBS entries using ULEDIT.

Many of the USERLOG.BBS entries can be revised and updated by the user, such as terminal configuration, password and expert level (if you have provided access to the appropriate Menu Control File command types.) Other entries, however, may be made only by you -- such as privilege level, A1/A2 flag settings, call timeout limit, and the like.

ULEDIT is fully menu-driven, and it immediately shows you any changes you make to a USERLOG.BBS record. The changes, however, are not written to file until you move to another entry or exit ULEDIT -- you can "unmodify" any changes prior to then.

ULEDIT also maintains a file called CALLERS.BBS although this action is invisible to you. This file contains an index of the

last 255 callers to your system and is used by the type 22 command to list the users on your system. It is important that your CALLERS.BBS and USERLOG.BBS files match, so they are backed up together. If they do not match, the userlog listed by the type 22 command will be dramatically incorrect.

When you call ULEDIT from DOS (USERLOG.BBS and CALLERS.BBS must be on-line), it will display this list of commands:

```
  ; = Move Forward one entry        - = Move Backward one entry
  G = Go to specified entry #       N = Go to next New entry
  D = Mark record deleted           R = Mark record reviewed
  K = Mark record registered        S = Selective Search
  M = Modify selected entry         U = Unmodify record

  + = Add New User to Userlog       # = Change total Call number
  P = Print to end of Userlog       C = Clear SIGNON2 read flags
  A = Alter A1/A2 for all users     B = Quit editing and do backup
  Q = Quit editing and exit         Any other = view current entry

  Which one?
```

ULEDIT always starts at the SYSOP entry (the first) in USERLOG.BBS. From there, keying a - (dash) will take you to the last, or newest, entry in the file ... very helpful when you want to do a rapid update of new users' files.

Now, let's examine a typical entry in USERLOG.BBS

```
                    Userlog Entry Number   151

<A>Name JOHN SMITH                <B>From PLYMOUTH ROCK
<C>Password ==> SQUAW             <D>User Flags ==> ---XX---
<E>Timeout ==>    30 Minutes      <F>Privilege ==>   125
<G>Auth #1 ==> ---X----           <H>Auth #2 ==> --------
<I>Terminal Width ==>    40       <J># of Nulls ==>     0
<K>Configuration ==> X---X-XX     <L>CLS ==>    13    13     0
<M>Expert Level ==>     0         <N>Last Msg Read ==>   522
<O>SYFGS ==> -----X-------------------
   Last 10 Logons from  64 Total Logons:
05/02/83  07:52         05/02/83  07:52         05/06/83  14:04
05/06/83  19:17         05/07/83  16:04         05/08/83  17:54
05/08/83  18:12         05/08/83  21:14         05/09/83  08:03
                        05/10/83  07:52
```

Here's what the log shows about John Smith from Plymouth Rock.

His most recent logon to the system was May 10, 1983 and he has logged on 64 times. The flag set in <D>User Flags shows his file has been reviewed by the SYSOP, and the flag set in <G>Auth #1 shows he has authorization to Menu Control File commands with a matching flag.

The flag set in the sixth position of <O>SYFGS shows he has SYSOP authority over the No. 6 message board (as defined with CEDIT). He may read and delete all messages posted on that board, regardless of whether they are public or private, or password-protected.

(NOTE: There are 25 positions in <O>SYFGS, numbered 1-to-25 from left-to-right. You may set any one, with an X, to give the user co-SYSOP authority on the correspondingly-numbered message board, so long as other privilege and A1/A2 flags are appropriate. In the SYSOP entry, the flags shown in the SYFGS field are used for system purposes, and MUST NOT BE ALTERED.)

For you, as SYSOP, to modify any of the A-through-O fields, you need only key M. The system will ask which field to modify, and show you the change. As noted above, the changes are not actually written to file until you move to another record, or exit ULEDIT via Q or B...and you may un-modify the record by keying U, which will remove all changes made prior to a disk write.

The P command will begin from the displayed userlog entry, and print all entries to the end of the file.

If you key D, K or R, ULEDIT will set a flag in the <D>User Flags field. Marking a record deleted (D) will remove it from the file at the next backup; marking it reviewed (R) means it will not be shown to you as New entry with the N command; and marking it registered (K) allows you to further categorize your users. TBBS does not currently utilize the K flag. It was intended that any automatic roll-off utilities should keep any records with the K flag set, even if they have otherwise expired but this feature has not yet been implemented in the standard TBBS software.

The C command will clear all the SIGNON2 read flags, so that (if you are using the CEDIT option to see the news file only once), SIGNON2.TXT will again be shown to all users at logon. Use this command when you update the news.

The S command is a powerful searching tool, especially useful when your USERLOG/BBS grows to hundreds of entries. It prompts for a search by name or A1/A2 flags, and will continue a search for all matches until you have reviewed the full file or halt the search.

The + command allows you to add a new user to your userlog. This command is handy at any time, and mandatory when you have disallowed new users in CEDIT. After you press the + key, you will be presented with a blank user entry. Fill in the information using the M (modify) commands. This blank entry will have a terminal configured for 80 columns, upper/lower case, line feeds required. You may change this with M commands if you wish.

The # command will allow you to set a new number for the total number of callers to your system if you wish.

The A command allows you to "gang" set or clear any A1/A2 flags for all users. This command is useful if you are re-organizing your system and change the meanings of some flags. You will be prompted to mark any flags you wish set or cleared. To do so, type a - for any flags you don't want changed, and an X for any you want to alter.

The Q command exits editing after writing to disk all modifications you have made.

The B command will prompt you to choose the drive on which you want the USERLOG.BAK file placed. As with MFSQZ, ULEDIT scans USERLOG.BBS, removing from it all files marked for deletion, then writing the remainder to the backup file. The backup is verified prior to deletion of the original USERLOG.BBS -- but you would be wise to make a VERIFY'ed copy of the master file prior to a backup, just for insurance.

IN CASE OF TROUBLE

Most TBBS problems stem from one of two causes: Hardware problems, or incorrect system construction. Do not assume that you have found a TBBS bug, until all other avenues have been checked out. TBBS has been tested for over 500 man years and is one of the most bug free pieces of software on the market today. In the even you do discover a bug, see the bug reporting section below. This reporting method will help us help you fix the problem as quickly as possible.

The following table lists the TBBS run time system error messages along with their most probable causes.

System Errors and Probable Causes

Error Message	Probable Cause	Correction
Board Not Defined with CEDIT	Menu Control file calls for board name not defined by CEDIT	Use CEDIT to define the board name.
	or Opt Data in Menu Control File is not spelled correctly or has incorrect case.	Use MEDIT to correct the Opt Data field.
Cannot Open MSGHDR.BBS or MSG.BBS, or USERLOG.BBS)	File not online, or DOS error.	Check system.
Menu Control File Command XX Not Implemented	Wrong number in Menu Control File Type Field.	Use MEDIT to fix incorrect Menu file entry.
Menu Stack Overflow	Sub-menus more than twenty deep.	Revise menu flow.
Q&A Buffer Overflow	Too many questions before verification.	Re-write Q&A file.
Unexpected DOS Error XX	(Various)	Refer to DOS Manual

TBBS displays other "error" messages, as well, but they are self-explanatory.

System errors are written to DAYFILE. DOS errors, as example, will write to file the error, the location calling the error, and a dump (in hexadecimal) of the FCB contents. (Refer to your DOS Manual for information on how the FCB is maintained by your system.) The FCB contents may be examined for clues of the error's origin -- it may contain, in ASCII, the name of the file which caused the error. Reading FCB's is quite technical, but the eSoft support personnel may ask you about this display.

Not all DOS errors, however, are indicative of some type of problem in your operating hardware or DOS. We must leave to you the final determination of the source of the problem, and its correction. eSoft support personnel are experts in TBBS, but not in the many DOS types it runs on. They will try to help you resolve your problems, but cannot offer support for DOS related problems.

General Troubleshooting Chart

Problem	Probable Cause and Resolution
System hangs when TBBS loads.	Modem has carrier detect forced on. Set the CD force switch on your modem to normal.
System does not recognize 1200 baud caller.	You have not set the CEDIT option to recognize 1200 baud or you do not have a properly modified cable installed. See the section on 1200 baud modem operation in the technical section for cable details, or set the CEDIT option properly. Another possibility is that your modem does not provide the HS signal on pin 12. In that case you need another modem.
System cycles to top menu after caller hangs up or doesn't hang up after timeout.	Modem has DTR forced on. Set the DTR force switch off. Or you have an Anchor Signalman Mark XII. This modem does not implement the DTR correctly and may not be used.
System freezes with the idle display not fully printed.	Diskette has parity error in either the MSG.BBS MSGHDR.BBS, USERLOG.BBS, CONFIG.CTL, or the MENU0000.CTL. The logo should show which was being loaded at the time.

General Troubleshooting Chart

Problem	Probable Cause and Resolution
Strange caller number, or menus list incorrectly.	You have accidentally deleted the SYSOP record from record 0 in the USERLOG. It needs to be restored.
User has no exit command from a menu.	You failed to provide one or made it require a higher privilege or authorization than the caller possesses in his userlog.
Relogon command causes loss of menu options.	The new user has lower privilege than it took to access the menu originally. Move the command to a menu which new users can access.
Custom CP/M TBBS behaves erratically or not at all.	Your INS?????.ASM file is incorrect. Check the BASE equates and the individual routines for accuracy. If the INS?????.ASM file is correct your system will run. If not, many subtle errors can be introduced. See the section on created custom installations for details.
TRS-80 Hard Disk system will not behave correctly.	If NEWDOS/80 2.5, you must use the /HD versions of both TBBS and TBBSDVR. If LDOS or DOSPLUS, You must set the High Memory protection when all drivers are removed, and copy SYS0 to your boot floppy. Most common is failure to set high memory FIRST as described in the installation section. Some custom DOSPLUS drivers will not relocate and require use of the /HD modules.
Modem answers phone, but the system appears not to know or hangs.	Last Caller hit +++ when system was echoing. Your modem was placed in the command mode. Use the Modem Init string in CEDIT to program the esc character to a number larger than 127.
System streams endlessly at first name?	Your modem is set in the echo mode. Reprogram it to non-echo with switches or modem init string in CEDIT.
Download files have extra newlines.	You used a wordwrapping file display command type. Change to the corresponding non-word wrap display type.

General Troubleshooting Chart

Problem	Probable Cause and Resolution
File Selection List Does not Display as it was created.	Selection lists are always word wrapped. See Page 10-21 for explanation of the word wrap algorithm. Format each selection line with a leading space.

Bug Reporting

As released, TBBS Version 1.3 is as bugfree as we can make it with many months of testing, and after online operation at several beta sites prior to release.

If you have what appears to be a bug, please report it to the TBBS Support system at (303)699-8222 or by mail to eSoft. To help you, we MUST know the following:

1. Registered TBBS owner's name.
2. Registration No. of your TBBS
3. Your DOS and hardware configuration.
4. Circumstances of problem -- what files in use, FCB dump (if any), Dayfile printout etc.
5. Circumstances of occurrence -- user on-line, SYSOP editing, etc.

We'll help you to the limits of our knowledge of the system AND the problem!

THE RS232 STANDARD

TBBS communications are carried out via yor computer's RS-232 port. The designation RS232 comes from the standard number given this mode of data transmission by the Electronics Industries Association (a national standards organization). This data transmission method is "serial" data transmission -- which means the data is sent one bit at a time down a pair of wires.

There are two methods of serial data transmission, both of which meet the RS232 standard. The two are asynchronous and synchronous. The difference between the two is in the data format itself. The data format is not covered by the RS232 standard. The standard only specifies the pin connections and the voltage levels which an RS232 device must use.

Most microcomputers are capable only of asynchronous transmission with their standard RS232 port. Thus, that is the TBBS method.

Asynchronous means that characters are fully self-defining and may be sent at any time with any amount of time between them. Without getting too technical, this is possible because each character is framed by a start bit which designates the beginning of the character and a stop bit which designates the end.

TBBS (and most other microcomputer-based asynchronous communications systems) uses a total character width of 10 bits. This width includes the start bit, eight data bits and one stop bit. The eight data bits are further subdivided to be the seven bits required for an ASCII character, plus a parity check bit. The parity bit allows checking for transmission errors. If the system is configured for eight data bits and no parity, then transmission of graphics and binary data is possible. This mode is becoming more popular among BBS software authors and system operators, because most terminal programs don't check parity and eight bits allows secure error-checking data transfer methods such as MODEM and DFT to be used.

Let's take a look at what the RS232 standard specifies. It first states the voltage levels used for transmission. It states a binary 0 (also called a "space" due to old-line Teletype usage), will be a voltage between +3 and +25 volts. It states that a binary 1 (or "mark"), will be a voltage between -3 and -25 volts.

The standard states that the voltage range between +3 and -3 volts, is undefined. Most RS232 interfaces use voltages in the range of +/-9 to +/-12 volts. The standard also indicates that drivers and receivers shall be constructed so that any accidental cross-connection will not cause damage. This means that any

signal line must be able to withstand being connected to + or 25 volts with no damage. (Not all "RS232" interfaces meet this portion of the standard.)

The standard defines transmission between two points. It defines these points as the Data Terminal Equipment (DTE) and the Data Communications Equipment(DCE). With your TRS-80, the computer is the DTE and the modem is the DCE. The standard further defines that all control signals are considered "true" (or on) when they are at a binary 0, or at a voltage level of +3 to +25 volts. It then defines a standard connector and the pin wiring arrangement. This definition has been revised twice, so that there are really three "standards" -- known as RS232A, RS232B and RS232C. In most applications, only a very small subset of the full set of signals are used.

The following table should prove very useful in all of your encounters with RS232 devices, not just when you are using TBBS:

Table of RS232 Standard Signals and Uses

Pin	Name	DTE	DCE	EIA CODE	CCIT CODE	RS232 Function	Notes
1	FG			AA	101	Frame (Protective) Ground	
2	TD	-->		BA	103	Transmitted Data	
3	RD	<--		BB	104	Received Data	
4	RTS	-->		CA	105	Request To Send	
5	CTS	<--		CB	106	Clear To Send	
6	DSR	<--		CC	107	Data Set Ready	
7	SG			AB	102	Signal Ground	
8	DCD	<--		CF	109	Received Line Signal Detect (in RS232 A and B, this is Data Carrier Detect)	
9		<--				Reserved for testing (+ voltage)	
10		<--				Reserved for testing (- voltage)	
11		<--		SA		Unassigned in RS232 B and C. In RS232 A this is (SA) Supervisory Transmitted Data. Bell 208A modem uses as QM equalizer mode	
12	SDCD	<--		SCF	122	Secondary Received Line Signal Detect Unassigned in RS232B. Is (SB) or Supervisory Received Data in RS232A. Becoming the standard for High Speed Detect in dual speed modems (103/212A, et al)	
13	SCTS	<--		SCB	121	Secondary Clear To Send	

					(Unassigned in RS232 A and B)
14	STD	-->	SBA	118	Secondary Transmitted Data (Unassigned in RS232 A and B. Used as (NS) or new sync signal in Bell 208A modems.
15	TC	<--	DB	114	Transmitter Clock (unassigned in RS232 A)
16	SRD	<--	SBB	119	Secondary Received Data (unassigned in RS232 A and B. Used as (DCT) Divided Clock Transmitter in Bell 208A modem.
17	RC	<--	DD	115	Receiver Clock (unassigned in RS232A)
18		<--			Unassigned. Used as (DCR) Divided Clock Receiver in Bell 208A modem.
19	SSRTS	-->	SCA	120	Secondary Request to Send (Unassigned in RS232 A and B)
20	DTR	-->	CD	108.2	Data Terminal Ready
21	SQ	<--	CG	110	Signal Quality Detector (Unassigned in RS232 A and B)
22	RI	<--	CE	125	Ring Indicator
23		-->	CH	111	Data Rate Selector (Direction
		<--	CI	112	optional, Code changes with direction implemented) (Unassigned in RS232 A and B)
24	ETC	-->	DA	113	External Transmit Clock (Unassigned in RS232A)
25		-->			Unassigned. Usually used as busy signal.

Control Signals: + = ON Data Signals: 0 (Space) = +3v to +25v
- = OFF 1 (Mark) = -3v to -25v

TBBS RS232 Driver

TBBS uses a high memory driver to host your computer for remote communications. This driver has several very powerful features which may be when running any programs via the TBBS-exit-to-DOS commands (Types 21 and 44). Upon exit from TBBS, the driver holds information reflecting certain User Log parameters. These may be examined by the program running, if desired.

The exact form of the driver and the location of these values vary with computer type.

The MS-DOS Driver

In the MS-DOS version of TBBS, the TBBSDVR program runs as a foreground task using the "exit and stay resident" MS-DOS function call. It patches itself into the BIOS interrupt vectors 10h (video) and 16h (keyboard). It then processes all BIOS calls normally, except that it slaves the keyboard and video to the RS-232 port if there is a carrier present. It also patches into the COM1: RS-232 hardware and input interrupt to process the serial data stream. The variable table discussed below is at the front of the driver and may be located by using the segment paragraph found in the INT 10 location and noting that the table begins at offset 130h in that segment.

The CP/M 2.2 Driver

In CP/M 2.2, the driver is loaded as an integral part of the TBBS program, and does not exist as an obviously separate program. However, as soon as TBBS loads, it examines the system to determine the available memory and re-locates the driver just below the CCP. The driver occupies 900h bytes of memory and protects itself by altering the BDOS jmp vector at location 0005h to jmp to the driver lowest location and there the original BDOS jmp is placed. This means that all CP/M programs should run properly, but that the CCP is never overwritten and there is 4.25K less TPA available when remote hosting. The driver's variable table is located immediately following the JMP instruction pointed to by the BIOS cold boot entry point. This means that you can obtain the bas table address with the following operations:

```
LHLD    0001H           ;get BIOS wm start vector
DCX     H
MOV     D,M             ;Fetch Cold Boot vector
DCX     H
MOV     E,M             ;In DE
INX     D
INX     D
INX     D               ;DE now points to the table
```

You may then use offsets added to DE to get to the individual table elements.

The TRS-80 Driver

In the TRS-80, the driver occupies memory locations from &HF9F4 through &HFFFF for the normal driver and locations &HF1F4 through &HF7FF for the TBBSDVR/HD version for NEWDOS/80 2.5. It sets HIMEM to protect itself. The variable table will therefore be at location &HFFE2 (normal) or &HF7E2 (TBBSDVR/HD). In addition the TRS-80 driver allows break control as follows:

&HFCB6 (nrm)	Poke 0 here to turn off BREAK from CTRL-A on RS232.
&HF4B6 (/HD)	Poke 1 here to turn on BREAK from CTRL-A. The default at TBBS load time is 0, for break off.

The TBBS Driver Variable Table

All versions of the TBBS driver maintains a variable table which contains information for its use. Some of this information is read only (that is it cannot be changed by you, only observed) and other portions of the table are read/write in that you can control them. The table structure is as follows:

Table location	Function
0	Inactivity timeout value in minutes (R/W). Time in minutes without input or output which causes system to logoff user. A zero here causes no timeout.
1	Call length allowed (R/W). Time in minutes, which -- when added to logon time -- gives time caller will be logged off, when time limit reached. This limit will logoff user, whether in DOS, BASIC, TBBS or anywhere, unless a file transfer is under way. A zero here allows unlimited time.
2	Dayfile Route flag (R/W). Printer to disk file routing flag. Setting bit 1 will send all dayfile output to file DAYFILE/LST. Setting bit 0 will cause the Dayfile output to go to the printer. Resetting both bits will cause Dayfile output to be thrown away.
3-4	CD drop debounce time (R/W)
5	User's privilege level, 0-255 (Rd only)

6	User's A1 authorization flags (Rd only)
7	User's A2 authorization flags (Rd only)
8-9	Last message read by user on last call (R/W)
10-12	User's logon time in binary (R/W)
13-14	Relative sector of user's USERLOG entry (Rd only)
15-16	Block adr in buffer of user's log entry (Rd only)
17-19	User's Clear Screen codes (Rd only)
20	Output U/C force (R/W). Anything other than zero forces all terminal output to be upper case.
21	Input U/C force (R/W). Anything other than zero forces RS232 input to upper case. This is not cleared by a TBBS warm start, although TBBS expects it to be zero. If not, no lower case input to TBBS will be possible.
22	Echo On/Off (R/W). Anything other than zero shuts off output to RS232. All display output will go to the host display screen only. TBBS does not reset this location, so be sure it is zero before a warm start.
23	Number of nulls which follow a <CR>. (R/W)
24	Re-boot on disconnect (R/W). If > 0, TBBS will re-boot if loss of carrier occurs. DO NOT change this flag as system lockup may occur if you do.
25-26	These two locations will have a non-zero number if a caller is currently logged on. (Rd only)
27	Flag to give SCMD routine to set DTR High.
28	Flag to give SCMD routine to set DTR low.
29	Line Feed flag (R/W). Other than 0 = add line feed after <CR> on RS232. A 0 = don't add line feed.

What TBBS expects from a MODEM

TBBS is written to be able to use either a smart or dumb modem. In order to do this, it interfaces to the modem using only the signals provided at the RS-232 DB25 connector. It will work properly with most modems.

The signals used by TBBS are:

RD and TD for data transmission.

DCD from the modem for notification of carrier detect.

DTR to the modem to allow it to auto answer and to hang up on a current call.

Optionally it uses the HS (SDCD) signal to determine 300 or 1200 baud on a dual speed modem.

Some older US and European modems also answer the phone immediately when DTR is asserted. They require the computer to do the logic required to answer the telephone, determine if a carrier is present, time out if no carrier is there, and hang up the phone. In TBBS, this method of modem handling is called European Modem Handling and will use the RI signal from the modem to determine if the telephone is ringing.

These signals are expected to behave in roughly the fashion that is standard on the Bell 103 and 212A modems. If they depart drastically, then TBBS and the modem will not work together. The two most common reasons for a modem being incompatible with TBBS are that the HS signal is unimplemented on a 300/1200 baud modem or that the DTR signal is unimplemented or does not work according to the Bell standard.

The Normal Modem Handling Method

When the CEDIT system option "European Modem Handling" is set to NO, then TBBS handles a modem as follows:

When the system resets, TBBS will set DTR low and expect the modem to hang up and refuse to answer any incoming calls. During this time, TBBS expects that carrier detect will not be asserted by the modem.

After the system has reset, TBBS will raise DTR and expect that the modem will now answer any incoming calls. It further expects

that the modem will time out a call which does not have a carrier present and automatically hang it up, without ever raising carrier detect (CD). Thus TBBS remains blissfully ignorant of the non-data call.

When the modem actually answers a call which has a carrier, it is expected to raise the carrier detect line. TBBS will debounce this line for 1 second to guard against modems which "glitch" CD during other transitions.

When a call completes, times out, or is terminated for any reason, TBBS lowers DTR for 5 seconds. It assumes that the modem will take this as a request to hang up, and that the CD line is low at the end of this interval.

300/1200 Baud

In the normal modem handling method, TBBS will handle a dual baud rate modem. To enable this feature, you set the "300/1200 Baud detect Enabled" option in CEDIT to YES. Then when TBBS detects a carrier to start a new call, it will examine the HS signal from the modem to determine whether it is a 300 or 1200 baud call. Now here comes a small problem. A bell 212A compatible modem provides the HS signal on pin 12 of the DB-25 connector. Upon studying the RS-232 interface of most microcomputers, you will find that pin 12 is rarely hooked up. Therefore, a special cable will be required to connect pin 12 of the modem to a pin on the computer which can read the signal. Depending on the computer interface, TBBS uses one of two pins on the computer. If it is implemented, the RI pin is used. This requires that pin 12 of the modem be wired to pin 22 on the computer. If the RI signal is not implemented on the computer, then the CTS signal is used. This requires a special cable which connects pin 12 of the modem to pin 5 of the computer.

The IBM PC and the TRS-80 use pin 22 for this signal input to TBBS. For CP/M systems, it depends on the computer. The Kaypro and Epson computers use pin 5.

You may choose to make your own modified cable, or eSoft, Inc. can provide a cable through your dealer. If you wish to purchase a special cable directly, you may do so from Ebert Personal Computers, Inc. 4122 S. Parker Rd. Aurora, Co 80014. Phone orders from EPC may be placed by calling 303-693-8400.

European Modem Handling

When handling older style American or European Modems, TBBS must have access to the RI signal. It handles these modems as follows:

When the system resets, DTR is lowered and TBBS expects that this will hang up the phone and lower the CD line.

After the system resets, TBBS will monitor the RI line to look for ringing. When it sees ringing, TBBS will raise the DTR signal and expect that the modem will go off hook and answer the phone.

Then TBBS will monitor the CD line for 30 seconds. If a carrier is found, then TBBS will proceed to the logon. If 30 seconds elapse and the CD line does not come true, TBBS will lower the DTR signal and expect te modem to hang up.

When a call completes, times out, or is terminated for any reason, TBBS lowers DTR for 5 seconds. It assumes that the modem will take this as a request to hang up, and that the CD line is low at the end of this interval.

Modem Switch Settings

From the preceding discussion of how TBBS handles a modem, you can see some guidelines for setting any modem option switches. The first rule is that if your modem has switches to force CD or DTR to fixed levels, instead of passing these signals through, they must be set for normal operation. Otherwise, TBBS cannot control the modem as indicated above.

Since TBBS expects a modem to act in a fairly dumb fashion, you should also suppress any "smart" features. This includes such things as echo in command mode, special escape sequences, result codes etc.

As an example, the Hayes modems should have their switches set as follows:

SW1-UP, SW2-UP, SW3-UP, SW4-DN, SW5-UP, SW6-UP, SW7-UP

The escape code should be overridden using the Modem Init string in CEDIT.

CONFIG.CTL File Format

The CONFIG.CTL file contains the values for all TBBS configuration parameters and options created or selected with CEDIT. Just as each TBBS system is different, so is each CONFIG.CTL file.

The file is comprised of a set of type or "marker" bytes by which TBBS reads the file, accessing information in value bytes related to each marker byte.

There are ten marker bytes. Seven of them are for entries of fixed length, and are used only once in CONFIG.CTL.

Of the other three, two are used for the variable-length Message Board and Terminal definitions, and are used repeatedly.

The tenth marker byte is also for a variable length entry, the Logon Greeting Text, and is used only once.

Each of the three marker bytes for variable length information are followed by a second byte, indicating the character length (in bytes) of that entry.

Because of these variables, marker bytes may appear in any order and at any location within CONFIG.CTL. Bytes of identical values may also appear in the file, but TBBS is able to discern them as part of the information for some other entry.

The operation of many of the value bytes depends upon the status of an individual bit (1=set, 0=not set) within the byte. Bytes used this way at present have some bits reserved for possible use in future versions of TBBS.

Our listing will show the following:

1. Function of the marker byte.
2. Value (in hexadecimal) of the marker byte.
3. The number of bytes related (fixed or range) to the marker byte.
4. Operation of the value bytes, in succession following the marker byte.

Fixed-length marker bytes

Func	TBBS Marker Byte	No. Of Value Byte(s)	Definition of Value Byte(s) or Bits
Chat Mode Control	01	1	00 = Chat mode on at all privilege and A1/A2 settings. (Overrides Menu Control File specifications.) 01 = Chat mode on only when privilege and A1/A2 settings qualify by Menu Control File specifications. 02 = Chat mode disallowed for all users.
New User Defaults	03	4	(1st Byte) = Privilege level assigned to user not in USERLOG.BBS. (2nd Byte) = A1 setting for new user. (3rd Byte) = A2 setting for new user. (4th Byte) = New user time limit (0=no time limit).
SYSOP Password	05	8	If the password is less than eight characters, it is left-justified with spaces. This password is required to logon as SYSOP remotely. If it is typed in as the answer to the First Name? prompt, the caller is logged on as SYSOP. If USERLOG.BBS has a different password in Entry 0, the caller will be asked to supply it also.
DAYFILE Routing and Options	07	1	Bit 7 = Don't log Menu File loads. Bit 6 = Don't log Data File reads. Bit 5 = Reserved Bit 4 = Reserved Bit 3 = Logoff after write to MSG TO SYSOP board. Bit 2 = Don't ask for msg reply except on EMAIL type board. Bits 1-0 = DAYFILE.LST routing: 0 0 = No Dayfile kept. 0 1 = Goes to printer only. 1 0 = Goes to disk file only. 1 1 = Goes to printer and disk file.
Various System Options	08	2	Bit 7 = SIGNON1.TXT to be displayed. Bit 6 = SIGNON2.TXT to be displayed. Bit 5 = NEWUSER.TXT to be displayed. Bit 4 = CTRL-S requires CTRL-Q.

Func	TBBS Marker Byte	No. Of Value Byte(s)	Definition of Value Byte(s) cr Bits
Various System Options (cont)			Bit 3 = Auto-blank screen is ON. Bit 2 = Don't count local logons. Bit 1 = SIGNON2.TXT is conditional. Bit 0 = 1200 Baud disabled. Bit 7 = TRSWATCH in use. Bit 6 = TRSWATCH uses alternate ports. Bit 5 = Use European modem handling. Bit 4 = No duplicate names in USERLOG. Bit 3 = Reserved: Always 0. Bit 2 = Local CRT display rate, 0-7. Bit 1 = " " " " Bit 0 = " " " "
Various System Options	09	1	Bit 7 = Reserved Bit 6 = Reserved Bit 5 = Security echo (*) on password entry. Bit 4 = Reboot after all logoffs Bit 3 = Don't ask pause on multiple message read. Bit 2 = Don't ask delete message, except on EMAIL type board. Bit 1 = Set chat command to say SYSOP will break in. Bit 0 = Select logon method: 1 = First Name, Last Name 0 = Password first
No Input Timeout	0A	1	Time in minutes allowed without input or output to user's terminal. TBBS will give one-minute warning; then, if still no input in one minute, the modem carrier is terminated.
Various System Options	0B	4	Bit 7 = Upload disk drive number. Bit 6 = " " " " Bit 5 = " " " " Bit 4 = " " " " Bit 3 = No new users allowed. Bit 2 = Request password if new user. Bit 1 = Wait msg on Userlog search. Bit 0 = Wait msg on Msg base search. Bit 7 = Don't Display Combined Boards.

Func	TBBS Marker Byte	No. Of Value Byte(s)	Definition of Value Byte(s) or Bits
Various System Options (cont)			Bit 6 = No Dayfile Divider Bar. Bit 5 = Log Failed Password attempts. Bit 4 = Log changed passwords. Bit 3-2 = Reply chain if no pause: 0 0 = Ask on each chain 0 1 = Always follow chain 1 0 = Never follow chain 1 1 = Illegal. Bits 7-0 = Local Auto Blank Timeout in minutes (0=none). Bits 7-4 = Dayfile Drive Number. Bits 3-0 = Dayfile User Area (CP/M).

Variable-Length Marker Bytes

(NOTE: Each of the three variable-length marker bytes will be followed by a length byte indicating the number of bytes or characters which follow it for the particular entry. Our description shows the range for each length byte.)

User Terminal Definition	06	8-18	The byte following the marker byte will show the length of the terminal name. Following the name are are five bytes defining the terminal's characteristics as follows: (1st Byte) = Number of nulls required. (2nd Byte) = Terminal options Bit 7 = Reserved Bit 6 = Line feeds required. Bit 5 = Reserved Bit 4 = Reserved Bit 3 = Upper case only. Bit 2 = Reserved Bit 1 = Reserved Bit 0 = Reserved (3rd Byte) = First clear screen code (4th Byte) = Second clear screen code (5th Byte) = Third clear screen code (6th Byte) = Reserved.

Variable Length Marker Bytes (cont.)

(NOTE: Each of the three variable-length marker bytes will be followed by a length byte indicating the number of bytes or characters which follow it for the particular entry. Our description shows the range for each length byte.)

Func	TBBS Marker Byte	No. Of Value Byte(s)	Definition of Value Byte(s) or Bits
Message Board	02	7-22	The byte following the marker byte will show the length of the Message Board name. Following the name are five bytes defining the board's options: (1st Byte) = Write privilege. User must have atleast this level to write to board. (2nd Byte) = Read privilege. User must have at least this level to read the board. (3rd Byte) = A1 Flag setting. User must match any set to use board. (4th Byte) = A2 Flag setting. User must match any set to use board. (5th Byte) = Option Flags Bit 7-6 = Board type - 0 0 = Standard Message Board 0 1 = EMAIL Board 1 0 = Reserved, illegal now. 1 1 = Reserved, illegal now. Bit 5 = Reserved Bit 4-3 = Scan includes: 0 0 = no From, To or Date. 0 1 = Date -- no From or To. 1 0 = Date and From -- no To. 1 1 = Date, From and To.
Logon Greeting Text.	04	0-255	The message may be up to 255 characters long; the length byte following the marker byte will show the length.
Modem Init	0C	0-63	The modem initialization string is terminated with a zero byte.

MENUXXXX.CTL File Format

Each MENUXXXX.CTL file defines the commands which may be used when that menu is displayed, and the parameters of how they may be used. Each entry in the file defines one command, although the same command type may be used as many times in the same MENUXXXX.CTL file or other files, as desired.

Your TBBS may have as many MENUXXXX.CTL files as you want. Each file may not be longer than five 256-byte sectors. Although each will have different command types for your system, they are all constructed similarly.

If the menu has a title text, it will be the first entry in the file. It will begin with a 01 byte, followed by the text (up to 255 characters) and be terminated by a 00 byte.

Each file will then have all the commands for that menu stored in sequential order, beginning with a one-byte value showing the length of that particular entry. The first six bytes of the entry, in order, are defined as follows:

(1st Byte)	Length of the file entry for that command in bytes, including the length byte.
(2nd Byte)	Privilege level of the command, which a user must match or top to see or use the command.
(3rd Byte)	A1 setting user must match to see or use the command.
(4th Byte)	A2 setting user must match to see or use the command.
(5th Byte)	Type of command, as entered with MEDIT.COM.
(6th Byte)	Character by which the user may select this command. Upper/lower case are synonymous. Any character produced by the computer's keyboard (including control characters) may be used, except the carriage return, linefeed and ?, which are reserved for system use.

Following the six value bytes will be the entry line text, which may be up to 200 characters. The line text will be wordwrapped to the user's terminal, and may contain carriage returns or linefeeds, as entered with MEDIT.

The line text will be terminated by a 00 byte. If the command type requires Opt Data field information, it will follow the 00 byte and be included in the count of bytes shown by the length byte.

Below is the logical sector display of a typical MENUXXXX.CTL file, with examples of the above. The logical sector size of this type of file is 256 bytes.

Typical MENUXXXX.CTL Logical Sector

```
     0 1  2 3  4 5  6 7  8 9  A B  C D  E F
00   0154 4242 5320 2331 2050 5249 5641 5445 .TBBS.#1.PRIVATE
10   204D 4149 4C0D 3D3D 3D3D 3D3D 3D3D 3D3D .MAIL.==========
20   3D3D 3D3D 3D3D 3D3D 3D3D 0D00>1F19 0000 ==========......
30   0853 533E 6361 6E20 596F 7572 2045 2D4D .SS>can.Your.E-M
40   4149 4C00 5052 4956 4154 451F 1900 0006 AIL.PRIVATE.....
50   5252 3E65 6164 2059 6F75 7220 452D 4D41 RR>ead.Your.E-MA
60   494C 0050 5249 5641 5445 1B19 0000 074C IL.PRIVATE.....L
70   4C3E 6561 7665 2045 2D4D 4149 4C00 5052 L>eave.E-MAIL.PR
80   4956 4154 4520 1900 0009 4444 3E65 6C65 IVATE.....DD>ele
90   7465 204F 6C64 2045 2D4D 4149 4C00 5052 te.Old.E-MAIL.PR
A0   4956 4154 451E 0000 0005 5454 3E65 726D IVATE.....TT>erm
B0   696E 6174 6520 2F48 616E 6720 5570 0054 inate./Hang.Up.T
C0   4552 4D19 1900 000C 4747 3E6F 2054 6F20 ERM.....GG>o.To.
D0   5072 696D 6520 4D65 6E75 0030 0000 0000 Prime.Menu.0....
E0   0000 0000 0000 0000 0000 0000 0000 0000 ................
F0   0000 0000 0000 0000 0000 0000 0000 0000 ................
```

When listed by MEDIT, the top of the above MENUXXXX.CTL file would like this:

```
TBBS #1 PRIVATE MAIL
====================
S>can Your E-Mail
```

The menu begins with the title text, indicated by a 01 byte and terminated by a 00 byte.

The menu command entries begin at relative byte 2C above (marked >, following the menu text title), with the first byte (1F) showing the length, in bytes (31), of the command entry in the file. Then, in succession, the bytes are:

19 - One byte value for privilege level of command type (25)
00 - One byte value for A1 Flags (none set)
00 - One byte value for A2 Flags (none set)

08 - One byte value for type of menu command (Type 8 - SCAN)
53 - One byte for menu command key stroke entry (Key = S)

- Display text line, here, in 17 bytes (maximum length: 200 characters)

00 - One zero byte to indicate end of text

- Opt Data field entry (presence and length according to command type -- here, for EMAIL Message Board PRIVATE).

The succession of bytes for each command entry is the same for all MENUXXXX.CTL files.

The above is an example, not a model! Your files will have different entries.

Typical USERLOG.BBS Disk Sector

```
     0 1  2 3  4 5  6 7  8 9  A B  C D  E F
00   4A4F 484E 534F 4E20 434F 4E4F 5645 5200 JOHNSON.CONOVER.
10   5049 5554 452C 204F 4849 4F00 0000 0000 PIUTE,.OHIO.....
20   0000 0000 0000 0000 2194 0200 0042 5345 ........!....BSE
30   4C35 5245 564F 4E4F 4320 1901 0000 5048 L5REVONOC.....PH
40   0D0D 0008 0000 5C01 101E 041C 5314 1B04 ............S...
50   1553 153B 0409 5315 2004 0553 1209 0311 .S.;..S....S....
60   5313 1B03 0E53 120D 0303 5315 0E03 0153 S....S....S....S
70   1330 0000 0000 0000 0000 0000 0000 0000 .0..............
```

TBBS stores the file information for a user in one logical sector of the file USERLOG.BBS. A logical sector for this file is 128 bytes long. The information is stored in each logical sector as follows:

BYTES	INFO	LENGTH IN BYTES
0-49	First/Last Name, Location (00's between)	50
50	Logon Password (Left justified)	8
58	This user's Privilege	1
59	This user's A1 Flags	1
60	This user's A2 Flags	1
61	Terminal Nulls Needed	1
62	Configuration Flags:	1
	Bit 7 - Reserved Bit 3 - Upper Case Only	
	Bit 6 - Line Feeds Bit 2 - Reserved	
	Bit 5 - Reserved Bit 1 - Reserved	
	Bit 4 - Reserved Bit 0 - Reserved	
63	Terminal Column Width	1
64	Clear Screen Code	3
67	No. of Logons	2
69	User Expert Level	1
70	Number of Last Message Read	2
72	Review Flags as follows:	1
	Bit 7 - Password Compromised Bit 3 - Reserved	
	Bit 6 - Marked for Deletion Bit 2 - Reserved	
	Bit 5 - Registered User Bit 1 - Reserved	
	Bit 4 - Reviewed By SYSOP Bit 0 - Reserved	
73	Time out limit	1
74	Last ten logons, five bytes per entry	50
124	SYFGS - 25 bits, 7 bits unused as follows:	4
	Left to right, from bit 7 of byte 124, authorizing a user as SYSOP of the respective message board, 1 to 25. The user -- if all other authorizations for that board are met -- has read/delete power for all messages, public or private, on that board. The SYSOP entry of USERLOG.BBS uses these bits for system purposes: They must NOT be changed with ULEDIT in the SYSOP record!	

Typical MSGHDR.BBS Disk Sector

```
     0 1  2 3  4 5  6 7  8 9  A B  C D  E F
00 >01E3 0508 5307 3211 5359 534F 5000 444F ....S.2.SYSOP.DO
10  4C46 2050 4849 4C49 5053 454E 0052 4947 LF.PHILIPSEN.RIG
20  4854 204F 4E2E 2E2E 0000 0000 0000 0000 HT.ON..........
30  0000 0000 0000 0000 0000 0000 0000 0000 ................
40  0000 0000 0000 0001 2020 2020 2020 2020 ................
50  0000 35E6 0301<E905 0853 0A23 3B44 4156 ..5......S.#;DAV
60  4520 4841 4748 4553 0041 4C4C 0043 4F4D E.HAGHES.ALL.COM
70  5045 5449 5449 4F4E 0000 0000 0000 0000 PETITION........
80  0000 0000 0000 0000 0000 0000 0000 0000 ................
90  0000 0000 0000 0000 0000 0000 2B44 4444 ............+DDD
A0  2020 2020 2000 003A 0D04>021C 050B 5308 .......:......S.
B0  1113 4B41 5448 5920 5348 4152 5259 0042 ..KATHY.SHARRY.B
C0  5249 414E 2044 4F42 494E 534F 4E00 4849 RIAN.DOBINSON.HI
D0  0000 0000 0000 0000 0000 0000 0000 0000 ................
E0  0000 0000 0000 0000 0000 0000 0000 0000 ................
F0  0002 2020 2020 2020 2020 0000 3E5B 0300 ............>[..
```

Header information for messages is stored three headers to a logical 256 byte sector, 85 bytes per header. There is one spare byte in each logical sector. Hence, above, headers start at bytes 00,55 and AA -- the bytes pointed to by > and <.

Within each 85 bytes, this information is stored, starting with the first byte of the header:

BYTES	INFO	LENGTH IN BYTES	START POS.
0-1	Message Number	2	00, 55, AA
2-7	Date and Time	6	02, 57, AC
8-69	From<00>To<00>Subject	62	08, 5D, B2
70-71	Times Message Read	2	46, 9B, F0
72-79	Message Password (left justified)	8	48, 9D, F2
80	Message Flags:	1	50, A5, FA
	Bit 7 - Message has been received.		
	Bit 6 - Message is marked deleted.		
	Bit 5 - Message is private.		
	Bit 4 - No wordwrap on display. (Set only by zapping file.)		
	Bit 3 - Message Has Replies		
	Bit 2 - Message is a Reply (bytes 68 & 69 are number of msg this is a reply to)		
81-82	Relative Sector in MSG.BBS file	2	51, A6, FB
83	Byte within Sector of MSG.BBS file	1	53, A8, FD
84	# of Board on which msg is posted	1	54, A9, FE

All message text is stored sequentially in MSG.BBS separated by a zero (00) byte.

Typical XXXXXXXX.QAF Disk Sector(s)

These abbreviated portions of two disk sectors show how TBBS stores data for .QAF files. The first byte of each file shows the type of .QAF file. A 01 byte is a Survey file, and a 02 byte is a Voting file. An 03 or an 06 indicate Survey or voting files with A1/A2 set/clear flags present. In this case Bytes 1 and 2 of the file are the A1/A2 flags to set, and Bytes 3 and 4 of the file are the A1/A2 flags to clear after running the Q & A. The logical sector size is 256 bytes on .QAF files.

```
     0 1  2 3  4 5  6 7  8 9  A B  C D  E F
00   0154 6869 7320 6973 2061 2054 4242 5320 .This.is.a.TBBS.
10   6578 616D 706C 6520 6F66 202F 5141 463A example.of..QAF:
20   0D0D>0551 7565 7354 696F 6E20 312E>0151 ...Question.1..Q
30   7565 7374 696F 6E20 322E>0051 7565 7374 uestion.2..Quest
40   696F 6E20 332E>0056 6572 6966 790D>0541 ion.3..Verify..A
50   7265 2020 7468 6579 2063 6F72 7265 6374 re..they.correct
60   2028 592F 4E21 3F0D>0354 6861 6E6B 2079 .(Y/N)?..Thank.y
70   6F75 210D>0500 0000 0000 0000 0000 0000 ou!.............

00   0249 6E20 7768 6174 2079 6561 7220 646F .In.what.year.do
10   2079 6F75 2074 6869 6E6B 2074 6865 2066 .you.think.the.f
20   6972 7374 204E 464C 2D55 5346 4C20 666F irst.NFL-USFL.fo
30   6F74 6261 6C6C 2067 616D 6524 7769 6C6C otball.game.will
40   2062 6520 706C 6179 6564 3F0D 0D31 2E20 .be.played?..1..
50   3139 3833 0D32 2E20 3139 3834 0D33 2E20 1983.2..1984.3..
60   3139 3835 0D34 2E20 3139 3836 0D35 2E20 1985.4..1986.5..
70   3139 3930 0D36 2E20 4E6507665 720D 0D56 1990.6..Never..V
80   6F74 6520 666F 7220 6F6E 6520 6279 206B ote.for.one.by.k
90   6579 696E 6720 6E75 6D62 6572 3A0D>0006 eying.number:...
A0  >0300>0800>0A00>0100>0100>1B00 0000 0000 ................
```

The top sector is a list of questions: The bytes marked by ">" are the terminator bytes for each question, telling TBBS what action to take when it displays that text line. The answers to a survey .QAF are stored in ASCII in a separate .SVY file, with the same file name. The .SVY file is appended to each time answers are written to the .QAF file -- it is not overwritten. Each time TBBS writes to the .SVY file, it begins the addition with the USERLOG.BBS name of the user, and the date/time of the user's logon.

The bottom sector is a voting file. Note that the question ends at relative sector location 9E with a 00 byte. The next byte -- a 06 at location 9F -- is the number of choices the question offers. That is the range of entries TBBS will accept; in this case, 1 through 6. Using the 06 as a guide, TBBS creates a runtime file in the next six succeeding pairs of bytes, in which to store the vote totals.

Word Wrap Algorithm

TBBS "word processes" much of its output, so that it looks like it was generated for the column-size screen (or printer) of your users' computer terminals. Considering systems now commonly display from 22 columns to 80 columns (and printers go to 132), this gives TBBS a very natural display for any user. The rules (algorithm) it follows are these:

Words are never broken unless they are too long to fit on one line. The algorithm defines a word as a group of characters surrounded on both ends by a space. The system will break the line between words at all times, if possible.

Actual carriage returns in the text file or message are displayed as either true carriage returns (line breaks to next line), or as a single space. How they are handled depends on the character before and after the return:

a. If the character before the carriage return is NOT a letter or number, and NOT a line ending character, then the carriage return is an actual line break. Line ending characters are:

   . , ; % : " ') or right-hand brackets

b. If the character before the carriage return IS one of the line ending characters, then the character AFTER the carriage return is examined. If it is NOT a letter, number or a line starting character, then the carriage return is an actual line break. These are line starting characters:

   (" ' or left hand brackets

c. Leading spaces on a new line are not printed.

So, how can you prepare formatted text? Or tabular rows of numbers? As example, how do you deal with wordwrap when writing a file for the Display File from Selection List Group of menu commands?

The fastest, easiest rule of thumb to remember is to start each line of specially-formatted text with one blank space. It fully follows the rules above. However, you can be fancier than that -- each of the characters below, used to start a new line following a carriage return, will force a line break:

* @ + $ = # > <) : & .

Getting Started FAST on TRS-80 I/III/4

This Chapter of the TBBS Operator's Manual is intended to get your system online and operating at a minimal level within a few hours using the sample system provided. However, if you have not yet read the introduction (Chapter 1), please do so now. Also, make copies of all files on your TBBS master disks, and store the masters. This is very important, because some of your TBBS files will be changed by installation or running of the sample system, and you do not want to ever change your master files!

Note: If you are upgrading from version 1.1 or 1.2 to 1.3, proceed directly to page 11-13, upgrade installation, if this is a new system installation, proceed with the next paragraph.

These installation instructions assume your system has a minimum of two single sided, double density 40 track drives. You do not need this exact configuration, however, just enough disk space to hold your DOS system along with the TBBS programs and the sample system files. If you have a hard disk, place all files you are instructed to copy onto your system from the release disk onto your hard disk instead.

We will begin here with a "cookbook" installation of the sample system. The Manual Installation instructions which follow the cookbook installation are intended to guide you to a working TBBS system even if you have a non-standard configuration, and will also explain some of the following steps in more detail.

The instructions also assume you have a typical 300 or 300/1200 baud Bell 103 or Bell 212A compatible auto-answer modem wired to your computer's RS-232 serial port. If you intend to use a 300/1200 baud modem, you will need to prepare or purchase a special cable before it will operate correctly. See the section in chapter 10 on "1200 baud with TBBS" for full details on operating dual speed. The cookbook installation will leave you with an operating 300 baud only system.

TBBS is distributed with a sample system which is explained in detail in chapter 2. Its top level is a simple, single-level system with one general user message board, an information section, a bulletin section, a download section, and product listing and ordering. This is similar in structure to many non-TBBS systems with which you may be familiar.

The sample system also includes a section for electronic mail (EMAIL) for two-way private messages by sender-recipient names. In addition it includes a section with four special interest groups (SIGS) each with its own message center. This section also demonstrates the combined board feature of TBBS by providing

a combined message section for all SIGS.

TBBS is the most powerful personal telecommunications system available today. This sample system barely taps the potential of TBBS. It is intended to get you started, so you may learn-by-doing to develop your own full, unique TBBS system.

TBBS will operate on either NEWDOS/80 V2.0 or 2.5, DOSPLUS V3.5 or LDOS 5.1.3 or 5.1.4. In the next few pages, we will provide startup instructions for each, separated by DOS type. Please turn now to the appropriate section for your DOS.

TBBS Startup: NEWDOS/80

1. Create a double density system disk for drive 0 and format a double density data disk for drive 1. The data disk should have no user files at all. Consult your NEWDOS/80 manual if you are uncertain of these procedures.

2. Place your newly-created system disk in drive 0 and enter the command:

   PDRIVE,0,5,TI=A,TD=A,TC=35,SPT=10,TSR=3,GPL=2,DDSL=17,DDGA=2,A

   This allows your master disks to be read using PDRIVE 5.

3. TBBS will require certain NEWSDOS/80 system options to be properly set. Enter the command:

   SYSTEM,0,AX=255,AY=N,BD=Y

   Since you have just created a new system disk, you should check other SYSTEM options to assure they are as you want them.

5. Next you should enter the command:

   AUTO TBBSDVR N

   to set the boot command to start up TBBS.

4. Next, you need to copy the following files from your release disks to your system and data disks. It does not matter which files go on which disk, as TBBS will execute correctly in any case. After you gain experience with TBBS, you will find that placing certain files on certain disks can improve performance and speed of you system, but do not worry about that just yet.

   All /CTL files
   All /BBS files
   All /TXT files except HARDDISK.TXT and MODS.TXT
   ORDER/QAF
   CEDIT/CMD
   MEDIT/CMD
   QAEDIT/CMD
   MFSQZ/CMD
   ULEDIT/CMD
   LDTBBS/JCL

An example of the NEWDOS/80 bulk copy command you might use

for this is:

```
COPY :0 :1,,NFMT,CBF,/CTL,SPDN=5
```

This command would allow you to insert a release disk in drive 0 when prompted, and copy all of the /CTL files on it to drive 1. It would read a release disk because PDRIVE 5 was set correctly above, and the SPDN=5 option told the copy to use PDRIVE 5 instead of the normal PDRIVE for drive 0. In this way you can share a drive between two disks which have different PDRIVE settings.

5. Next you need to copy the TBBS run time system on to your system disk. If you are running NEWDOS/80 2.0 (floppy disks only) then copy the following files onto your system disk:

   TBBSDVR/CMD
   TBBS/CMD
   TBBS/OV1
   TBBS/OV2

   If you are running NEWDOS/80 2.5 (Hard disk system) then copy the following files to your system disk and name them as follows:

TBBSDVHD/CMD	named as TBBSDVR/CMD
TBBSHD/CMD	named as TBBS/CMD
TBBSHD/OV1	named as TBBS/OV1
TBBSHD/OV2	named as TBBS/OV2

   You will also need to print the file HARDDISK.TXT and install the MODS listed there in the other /CMD files.

6. Proceed to the After Installation section on Page 11-11.

TBBS Startup: DOSPLUS 3.5

1. Create a double density system disk for drive 0 and format a double density data disk for drive 1.

2. Next, you need to configure your newly generated system. In this process, you will need to reserve space for the TBBS driver program. It needs the uppermost memory locations in your system. Therefore it is important that you do the configuration correctly, or your TBBS system will malfunction.

   IMPORTANT!! You must begin with no /CFG file active. Enter the command:

   SYSTEM

   The HIGH$ value and the TOP$ value should both be FFFFH. If they are not, then you cannot proceed until you remove the drivers or filters which are currently installed. We will re-install any critical filters and drivers after the next step.

3. Next enter the command:

   SYSTEM (HIGH=F95FH)

   Then enter the command:

   SYSTEM

   The HIGH$ value should now read F95FH. If it does not, do not proceed until you correct it.

4. Now you may proceed to do any drive configuration your system requires such as multi-sided, multi-density, 80 track, or hard disk. and install any filters you wish. End this process by entering the command:

   SYSTEM TBBS/CFG

   to create the configuration file. See your DOSPLUS manual for configuration details.

5. Enter the command:

   AUTO TBBS/CFG;TBBSDVR D

6. Next, you need to copy the following files from your release disks to your system and data disks. It does not matter

which files go on which disk, as TBBS will execute correctly in any case. After you gain experience with TBBS, you will find that placing certain files on certain disks can improve performance and speed of you system, but do not worry about that just yet.

All /CTL files
All /BBS files
All /TXT files except HARDDISK.TXT and MODS.TXT
ORDER/QAF
CEDIT/CMD
MEDIT/CMD
QAEDIT/CMD
MFSQZ/CMD
ULEDIT/CMD
LDTBBS/JCL
TBBSDVR/CMD
TBBS/CMD
TBBS/OV1
TBBS/OV2

7. Proceed to the After Installation section on Page 11-11.

TBBS Startup: LDOS

1. Make a fresh system disk for drive 0 and format a double density data disk at the maximum capacity (tracks and sides) of your drive 1.

   NOTE: The system disk must be bootable! Thus, if you have a Model I the disk must be single density, unless you have obtained the program SOLE from MISOSYS which allows creation of a double density LDOS system disk which will boot on a Model I.

   IMPORTANT!!! You MUST assure no filters or drivers are loaded by LDOS until your TBBS installation is complete. Do this by holding down both the CLEAR and ENTER keys at reboot, as noted below.

2. Place your new system disk in drive 0. Re-boot, while holding down the CLEAR and ENTER keys, until the LDOS Ready prompt appears. Then, enter the command:

   MEMORY

   The system must respond with the message:

   High = X'FFFF'

   or you may not proceed. If a different number prints, then you have inadvertently loaded a driver or filter. It must be removed before proceeding. Then enter the command:

   MEMORY (HIGH=X'F95F')

   This reserves the high memory location for TBBSDVR/CMD, which is not relocatable.

3. Next, enter the commands:

   SYSTEM (DATE=OFF)
   SYSTEM (TIME=OFF)

4. Now you may establish your desired SYSTEM (DRIVE=), DRIVER, and Filter settings (such as PDUBL for a Model I, or *KI filters etc). Refer to your LDOS manual for this information. When you have your system configured as you want, enter the command:

   SYSTEM (SYSGEN)

   This will save your configuration for reboot.

5. Enter the command:

   AUTO TBBSDVR L

6. Next, you need to copy the following files from your release disks to your system and data disks. It does not matter which files go on which disk, as TBBS will execute correctly in any case. After you gain experience with TBBS, you will find that placing certain files on certain disks can improve performance and speed of you system, but do not worry about that just yet.

   All /CTL files
   All /BBS files
   All /TXT files except HARDDISK.TXT and MODS.TXT
   ORDER/QAF
   CEDIT/CMD
   MEDIT/CMD
   QAEDIT/CMD
   MFSQZ/CMD
   ULEDIT/CMD
   LDTBBS/JCL
   TBBSDVR/CMD
   TBBS/CMD
   TBBS/OV1
   TBBS/OV2

7. Proceed to the After Installation section, Page 11-11.

Non-Standard TBBS Installation

If you are using these instructions, we must assume your system cannot accept any of the three standard DOS installations. So, our instructions here will be generalized. You will need to place online various programs and files from your TBBS master disks. (The programs and files are listed and explained briefly in Chapter 1).

1. First, prepare a system disk and a data disk. Use your DOS manual as a guide. During this process, you must reserve the memory from F95FH through FFFFH for the TBBSDVR/CMD program. If it is not possible to reserve this memory because you are using NEWDOS/80 V2.5 or because some driver or filter you need to use will not relocate, then you can reserve the memory from F15FH through F7FFH instead and use the "HD" series of TBBS run time programs as described below. This should only be done as a last resort, since it will cause your system to have a smaller message capacity.

   Note: to reserve this memory, you must first boot up your system with no drivers or filters installed. The High Memory setting must be FFFFH. Then enter the command to set the High Memory to F95FH to reserve the memory. This must be done BEFORE you proceed with any other configuration.

2. Next copy the following files from the release disks to your system and data disks.

   These files MUST be online for operation of the TBBS startup system:

USERLOG/BBS	INFOM/TXT	INFO4/TXT	PRODM/TXT
CALLERS/BBS	INFO1/TXT	BULLM/TXT	DOWNM/TXT
MSGHDR/BBS	INFO2/TXT	BULL1/TXT	ORDER/QAF
MSG/BBS	INFO3/TXT	BULL2/TXT	MENU0000/CTL
MENUTERM/CTL	MENUMAIL/CTL	MENUTSIG/CTL	MENUSTRS/CTL
MENUSAPL/CTL	MENUSATA/CTL	MENUSIBM/CTL	MENUSIGS/CTL
* MENUREMS/CTL	LDTBBS/JCL		

   They may be on any drive -- drive location is not important to TBBS.

   These files MAY be online. (If you plan remote editing.)

CEDIT/CMD	ULEDIT/CMD	MEDIT/CMD
MFSQZ/CMD	QAEDIT/CMD	VOTESUM/CMD

   * If the above /CMD files are not online, you will not be able to access them with the remote editing menu, MENUREMS/CTL.

You should place them on a "system maintenance" disk, for your host terminal use.

3. Lastly you must install the TBBS run time system files. If you have reserved the memory from F95FH through FFFFH (the normal condition) then copy the following files from your release disk:

   TBBS/CMD TBBS/OV1 TBBS/OV2 TBBSDVR/CMD

   If you are using NEWDOS/80 2.5 or for other reasons have reserved the memory from F15FH through F7FFH, then copy the following files from your release disk and rename them on your system disks as follows:

   TBBSHD/CMD (renamed as TBBS/CMD)
   TBBSHD/OV1 (renamed as TBBS/OV1)
   TBBSHD/OV2 (renamed as TBBS/OV2)
   TBBSDVHD/CMD (renamed as TBBSDVR/CMD)

4. To complete your Manual Installation, enter the command below appropriate for your DOS, on the Drive 0 system disk.

   AUTO TBBSDVR N for NEWDOS/80
   AUTO TBBSDVR D for DOSPLUS
   AUTO TBBSDVR L for LDOS

5. Proceed to After Installation, Page 11-11.

After Installation

Your TBBS system is shipped with a default CONFIG/CTL file and a USERLOG/BBS with a default SYSOP entry as Entry 0. You must now update these.

Refer to Chapter 8 (Using ULEDIT) for general instructions on editing the USERLOG/BBS, and change your SYSOP password. You will want to read Using ULEDIT at length, later, but for now, you need only to M (Modify) the SYSOP password entry for your own use.

Refer to Chapter 3 (Using CEDIT) for general instructions on Changing System Options for your system. You will need, now, to read this section in some detail, because there are many System Options to be edited with CEDIT, and some of them are affected by your particular hardware configuration. Making the changes will require checking of each present default value.

At this time you should also print out the file MODS/TXT. This file contains a list of field modifications you can install into TBBS for certain non-standard equipment. The file also explains how to install these modifications.

Host (SYSOP) Terminal Operation

Once you have completed the above steps, your system is ready to go online! Boot your system disk. Key the left arrow and right arrow simultaneously, and TBBS will respond, "SYSOP Logging On(Y/N)?" Key Y, and you're on your way!

The left arrow-right arrow logon combination is only functional at your host terminal. Answering Y identifies you as SYSOP, giving you full system access. When you're ready to look at your system from the user viewpoint, key N from the left arrow-right arrow logon, and TBBS will handle you as a regular user. (Should your screen display operate a little too fast, or too slow, use CEDIT to change the System Option for Local Display Rate.)

Pressing left arrow-right arrow while a user is online will enter the "chat" mode. You exit the "chat" mode by keying right arrow alone.

Keying up arrow-down arrow while a user is online will display the user's name and calling location on your local screen. The caller will not be aware of anything happening when you do this.

You may exit TBBS (when it is idle) to DOS, leaving TBBSDVR active, by keying the up-arrow. From DOS ready, entering TBBS

will recall the system.

Keystrokes from the host terminal will actuate the system just as if they were from the user's terminal. Using this capability, you may exercise all functions of TBBS (except XMODEM file transfers) from your terminal.

You now have only a very minimal TBBS system -- just sufficient to help you start learning the full power of TBBS. We suggest you now spend some time now practicing with the system. Later, you will need to return to the manual at Chapter 2, and work through the entire manual.

All files you have used thus far may be retained as you build your system, by using the various TBBS editors. You may add to, and re-arrange the files as you want -- but do not change your master disk files!

As you develop your TBBS system, don't forget about one of your most valuable resources. That is the TBBS SYSOP support board at 303-699-8222. As a registered owner, you will be able to ask questions of Phil Becker (the TBBS author) and of other SYSOPs. This sharing of information among TBBS owners, and the support received on this system, makes it one of your best ways to get questions about TBBS installation and use answered.

Upgrading TBBS V.1.1 or 1.2 to V.1.3

To upgrade your present TBBS system:

1. Copy all /CMD files from your version 1.3 master disks to appropriate locations on your present system. Also copy the files TBBS/OV1 and TBBS/OV2 onto your system. Note: If you are using NEWDOS/80 Version 2.5, then copy and rename the following TBBS run time files:

   TBBSHD/CMD (renamed to TBBS/CMD)
   TBBSHD/OV1 (renamed to TBBS/OV1)
   TBBSHD/OV2 (renamed to TBBS/OV2)
   TBBSDVHD/CMD (renamed to TBBSDVR/CMD)

2. If you are using LDOS or DOSPLUS, you will have to reconfigure your system to the new high memory value of F95FH.

3. Using CEDIT, load your existing CONFIG/CTL file. Review the file, particularly the new System Options, and set them as you wish for your system. There are several new options. These will all be given default values when you load your CONFIG/CTL file, but you should review them for the desired settings. See Chapter 3 for the new CEDIT instructions. You will also have to re-specify your upload drive setting, since it may now be up to drive 9 (for hard disk systems).

3. Write the CONFIG/CTL file back to disk. This completes the conversion. You will want to review the manual, however, for new functions added to the various TBBS editor programs.

4. If you are converting from Version 1.1, you will need to change your AUTO command to the one appropriate for you DOS. See manual installation above. If you remain on NEWDOS/80, the Version 1.3 AUTO command is:

   AUTO TBBSDVR N

Subject.....Page No.
==========
A1/A2 Flags, Setting..... 3-16, 8-3
ASCII .TXT Files..... 4-29, 4-31, 4-34
Aborted Uploads..... 4-37
Authorization Required Message..... 4-26
Auto Answer Message..... 3-11
Auto Blank Host Console..... 3-9
Auto Delete Message..... 3-11

BREAK Key, Control..... 10-5
Baud Rate Detect..... 3-10, 10-8
Binary-to-ASCII Conversion..... 4-29
Bug Reporting..... 9-4

Change User Password..... 4-40
Character Width, RS232..... 10-1
Chat Procedures..... 3-7, 3-8
Clear Screen Code..... 3-23
Co-SYSOP Flags (SYFGS)..... 8-3
Co-SYSOP Message Board..... 6-1
Combinable Message Boards..... 3-20
Configure User Terminal..... 4-41
Count Local Logons..... 3-17

DOS Remote Operation..... 4-42
Day File (DAYFILE.LST)..... 3-12, 3-13
Defining Message Boards..... 3-19
Defining User Terminals..... 3-22
Delete Messages Routine..... 4-25
Deleting Message Boards..... 3-21
Deleting Users From System..... 8-3
Disk Space Management..... 6-18
Display Partitioned File Text..... 4-34
Display Book Chapters..... 4-34
Display Selection List File..... 4-31
Downloading BASIC Programs..... 4-29
Duplicate User Names..... 3-16

EMAIL Message Boards..... 4-27
Editing Message Headers..... 7-1
Editing USERLOG.BBS..... 8-1
European Modem Handling..... 3-10, 10-9

FCB Contents Dump..... 9-2
File Format: CONFIG.CTL..... 10-10
File Format: MENUXXXX.CTL..... 10-15
File Format: MSGHDR.BBS-MSG.BBS..... 10-19
File Format: USERLOG.BBS..... 10-20

INDEX

Subject.....Page No.
==========
File Format: XXXXXXXX.QAF..... 10-20
File Transfer Conversions..... 4-29
Filespec extension (.ext) conventions..... 1-2
Functions of CEDIT..... 3-1

GOSUB-GOTO-RETURN Flow Control..... 6-4
General System Function Commands..... 4-39
Getting Started FAST!..... 11-1
Greeting Text..... 3-2
Greeting Text - Extended..... 3-18
Groups of Menu Commands..... 4-13

High Memory Use..... 10-5
Host (SYSOP) Terminal Operation..... 11-5

Introduction to TBBS..... 1-1

Leave Message Routine..... 4-23
Line Feed, Terminal..... 3-23
List of Programs and Files..... 1-3
Listing Menu Entries..... 4-8
Local Display Rate..... 3-8
Logon Methods..... 3-14

MEDIT Operations Review..... 4-6
MODEM Protocol 4-38, 4-33
MSG TO SYSOP Logoff..... 3-12
Manual Organization..... 1-1
Menu Branching Commands..... 4-19
Menu Construction..... 4-1
Menu Stack Overflow..... 4-20
Menu Title Text..... 4-7
Menu User Exit..... 4-20
Message Board Authorizations..... 4-26
Message Board Commands..... 4-21
Message Board Privileges..... 3-19, 4-21
Message Passwords..... 4-23
Messages To/From TEXT File..... 7-5
Messages, Block Mode..... 4-24

NEWUSER.TXT File..... 3-18
Naming Menu Control Files..... 4-1
New User Default Settings..... 3-16
New User Info File..... 3-18
News File At Logon (SIGNON2.TXT)..... 3-18
Nulls, Terminal..... 3-23

Subject.....Page No.
==========
Offensive Messages..... 6-6
Online Files..... 1-2
Opt Data Field (General)..... 4-5
Overwriting Files On Upload..... 4-37

Password Entry..... 3-15, 4-40
Password Security Echo (*)..... 3-15
Password, Change User..... 4-40
Password, SYSOP..... 3-16
Pause on Message Reads..... 3-11

QAF File Functions..... 5-1
QAF Terminator Bytes..... 5-2

RS232 Standard..... 10-1
Read Messages Routine..... 4-22
Reboot on Logoffs..... 3-9
Remote SYSOP Editing..... 6-3
Review of CEDIT Operations..... 3-3

SIGNON1.TXT Greeting File..... 3-18
SIGNON2.TXT News File..... 3-18
SYSOP Password..... 3-16
SYSOP Remote Menu Commands..... 4-41
Scan Messages Routine..... 4-24
Searching USERLOG.BBS..... 8-3
Select Message Boards..... 4-40
Selection List Families..... 4-31
Selection List Keys..... 4-31
Single File Display Commands..... 4-29
Squeezing Message Base..... 7-1
Standard Message Board..... 3-11
Survey QAF Files..... 5-2
System Error Statements..... 9-1
System Options, List..... 3-4
System Utilities Menu..... 4-42

TBBS HQ. Support..... ii, 9-4
TBBSDVR (RS232 Driver)..... 10-3
Tokenized BASIC Conversion..... 4-29
Table of Contents..... iii
Terminal Definitions, List..... 3-22
Types Of Menu Commands..... 4-10

Upload Drive Select..... 3-9

INDEX

Subject.....Page No.
==========

Upload File Commands..... 4-36
Upload Protocols..... 4-38
User Inactivity Time Limit..... 3-9
User Log Display..... 4-40
User Menu Display Modes..... 4-40
User Re-Logon..... 4-41
User Time Display..... 4-39
Using CEDIT..... 3-1
Using MEDIT..... 4-1
Using MFSQZ..... 7-1
Using QAEDIT..... 5-1
Using ULEDIT..... 8-1
Using VOTESUM..... 5-6

Voting QAF Files..... 5-4

Word Wrap Algorithm..... 10-21