tbbs eSoft

Understand How it Works

The Bread Board System (TBBS)

Version 2.2M (Multi-Line)

Understanding How It Works

Manual by Alan Bryant

Table of Contents

Chapter 1: Understanding How TBBS Works

Chapter 2: Understanding the Logon Process

Chapter 3: Usage Control, Accounting and Tracking

Chapter 4: How Other Stuff Works

Thank You For Using TBBS

All of us here at eSoft want to take this opportunity to thank you for selecting The Bread Board System – TBBS – for your online information management needs. We wish to assure you we will do everything we can to make your TBBS experience a successful one.

The purpose of this guide is to assist you in understanding our product in the shortest possible time. TBBS is complex, meaning that it is composed of many easily understood parts (as opposed to complicated, which would imply that it is difficult to use). This guide will serve to explain "the big picture" of our product, which is critically important to successfully using TBBS.

Please take the time to read through this guide before you install the software, and before you even open the large Reference Manual. Every minute you spend reading this material will save you at least 10 minutes in the long run!

How to Use the TBBS Manual

The TBBS documentation is composed of the following parts

- **Understanding How TBBS Works (This Book)**
  If you are new to TBBS you should begin by reading this book. It contains basic, easily understood information on the concepts and operation of TBBS which you must understand to use TBBS effectively. Even users of previous versions will learn from it.

- **Reference Manual**
  A complete, detailed reference for system installation, setup and operation. This manual contains all of the detailed information you will need when you are setting up or using specific portions of TBBS.

- **In Case of Trouble ...**
  This brochure details how to get technical assistance for TBBS.

What is "The Bread Board System"?

Contrary to appearances, TBBS is not so-named because it's a BBS (Bulletin Board System) software product. TBBS is an acronym for "The Bread Board System." But what does **that** mean?

In the earlier days of electronics, engineers used empty circuit boards to prototype designs for their projects. They would place electronic parts on the board, wiring and soldering them together. This process was known as "bread-boarding" and the empty circuit board was the "bread board" itself.

In 1980 when Phil Becker (an electrical engineer himself) was designing what is now TBBS, he selected the name *The Bread Board System* because he designed the software to allow the same sort of extreme flexibility an engineer enjoyed with a bread board.

With TBBS you have a component based system – like the engineer and his bread board. Once you have learned how to use TBBS, you can easily and quickly assemble a custom system from a series of parts, putting them together as your own unique needs dictate.

You Are in Control

The Bread Board System lets you prototype by combination – to put the parts together any way you wish – changing parts or adding new parts easily as your needs evolve. Its template approach allows you to build a custom system from pre-fabricated parts without any programming. This accessible flexibility is what makes TBBS the most powerful product of its type available. That's why it is named after the concept of bread boarding.

It can be helpful to your understanding of TBBS to think of it as a building toy kit. There are many different parts, each with a different function. You can use as many of each part as you wish (you have been given a virtually unlimited number of each one), and you can put the parts together any way you wish.

Most TBBS System Designers are familiar with only a small fraction of the parts in the kit, and use only some of those. TBBS is designed this way on purpose, so that the parts you need are there when you need them. You *bread board* your own unique system, using only those parts that suit your needs.

It's important to remember that with TBBS, you – the System Designer – are in control of your system. What TBBS does, what it looks like, and who can use it are entirely controlled by your input. However, because TBBS offers you so much flexibility it can be intimidating at first. If you learn the concepts explained in this manual, and spend a little time using TBBS, you will find it is quite easy to use. And once you master even a small portion of its capability you will find yourself designing multiuser information systems you never would have thought you could attempt alone.

Where to Begin

The first step to installing and setting up your new TBBS is to read this guide. It's designed to get you up to speed quickly and painlessly on the broad concepts of the TBBS software. *This background is essential to successfully using the software.*

We know you're excited to begin, but take the time to read this guide completely, even if you don't read anything else!

This guide will introduce you to all the functionality of TBBS. By the end, you will have an understanding of the following:

- What TBBS is capable of doing
- How TBBS handles access control
- How users log on to your TBBS
- How menus are built and operate
- How accounting and tracking works
- How you gather information from your users
- How to upload and download files
- How electronic mail is handled by TBBS
- How the message system works
- How you use TBBS to build your personalized system

Once you have the basic understanding of these concepts provided by this manual you will be able to begin installing, configuring and using your new TBBS.

The Structure of TBBS

Interaction with TBBS begins with logon – the act of someone connecting with your system, and logging on as a user. You have a great deal of control over this aspect of TBBS, as you'll discover later. You can limit access to your system in a variety of ways, controlling who can access it, through what methods, and for how long. Once a user is actually logged on, he can interact with the system itself, taking advantage of features you provide.

Every TBBS system is constructed as a series of menus. The menus are like building blocks you use to build your own unique system. You are the builder and the architect. The menus are of your own design, and the structure you end up with can be as large or small, as simple or complex as you wish.

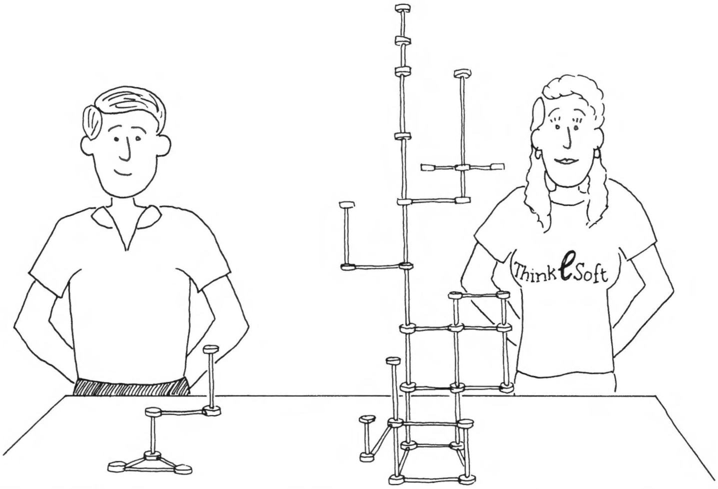
Each menu is a series of smaller components called *menu entries*. There is a tremendous variety of these components, each designed to perform a specific function. These parts are pre-fabricated, so you can select the pieces that meet your needs best, integrating them wherever you want. Because there is a virtually unlimited supply of each component, you can use as much or as little of each as you want. If your application demands a lot of one piece and none of another, that's fine. Take only the pieces you need and leave the rest (they won't go anywhere, and are available if you need them later).

Each menu you build from the components "self-customizes" to each person who uses your finished system. Through a series of *access controls* you have the ability to decide who has access to anything on your system (and who doesn't). It's easy to provide things for a user or group of users, and restrict people from other things as you wish. With TBBS, you – the System Designer – are in control of your system . . . what it does, what it looks like, and who can use it.

What Does All This Mean to You?

So far, you've learned that TBBS is constructed of menus that you design and assemble from pre-fabricated components. You can use as much or as little of each component as you want, and you control access to each component. But what does all of this mean to you?

By building menus yourself, you're creating your own custom system. The system you create will be different from systems created by anyone else. Your system will meet your needs in the way you want them met. It's this concept which makes TBBS so powerful.



Since menus adapt to each user automatically, you can actually create multiple custom systems in one. The one single system you create may well appear differently to different users. Your TBBS can address the needs of many different people – whatever your own situation dictates.

Because you're doing the building, and because you have a virtually unlimited number of parts at your disposal, you can build a structure as small as you want, or as large and sophisticated as you wish – the choice is yours!

How Access Control Works

As you've learned, your system can automatically adapt to each user individually. TBBS wouldn't be very appealing if it gave everyone an "open door" to your computer system. For this reason, TBBS provides simple but powerful methods to control access, not only to the system itself, but also to the services you provide on the system.

Access control applies to two areas of the system – at logon (when people call into the system) and in the menu structures you build.

Logon Access Control

Access to your TBBS is controlled by a system of user identifiers and passwords. Each user on your system will have a unique identifier, usually the user's name (but there are other options), along with a password. The entire ability (or inability) of a caller to access your TBBS revolves around these two items, so as simple as they sound, they are very important.

TBBS maintains a list of all users in a *userlog*. If a particular caller is unknown to TBBS, you can have the system automatically create a userlog account and authorize them with a set of default access control settings, or you can choose to disallow unknown callers from accessing your system at all.

Menu Access Control

To allow you to tailor access to your specific needs, there are two access control sub-systems in TBBS. One of these systems is *inclusive*, meaning you can provide different levels of access with each level *including* the lower levels. The other system is *mutually exclusive*, allowing you to assign a particular user (or user group) access to any of several items in any combination you wish.

Each user on your system has their own unique access authorization settings in their userlog account. These may or may not be the same as those of other users.

Your TBBS system is constructed of menus, and each menu is made up of individual menu entries. You can control access to each and every menu entry on your system by setting access control on each one. These menu entry access settings are compared against a caller's userlog access authorization settings to determine whether or not a particular user is qualified to access any one system function.

Menu Templates

The single most powerful feature of TBBS is its *template-based menu structure.*You've learned that you can control access not only to the system itself, but to each individual function the system provides. You've also learned that TBBS doesn't lock you into a particular menu structure, because you build the menus yourself. The template menu structure is what makes this possible.

As the System Designer, when you create TBBS menus you are creating *menu templates*. You can create just one menu template that contains all the functions provided by your system, or you can create hundreds of templates in a sophisticated tree structure.

Menu templates describe what users are able to see and access on your system. However, only some parts of any menu template you build may be used when the menu is customized for a particular user, because the user may not qualify for everything in that menu template.

The menu templates you design and build are stored on disk for use by TBBS while it's running and are therefore called *disk templates*. When a user accesses a menu TBBS first reads the disk template into memory where it becomes a *memory template*. TBBS then builds a custom version of the menu for this user (called the *user template*) from the memory template.

This three-part process may sound somewhat complicated at first. It's actually very simple, however, because TBBS does it automatically for you. You just need to design and build the initial disk template – the general pattern – then TBBS automatically customizes each menu for each user, based on that user's access authorization settings (in the userlog).

How TBBS Customizes Menus

TBBS creates a custom menu display for each user. This display is based on their access authorization settings (from the userlog) and each menu template entry's access control settings. Because a particular user may have access to some menu entries and not others, showing the user entries he or she is not qualified for doesn't make much sense. Therefore, TBBS creates a custom menu display for the user each time a menu is accessed. You should understand this process so you can obtain the displays you want, but you don't need to be too concerned about it because TBBS handles it for you automatically.

When a user accesses a menu on your system, TBBS reads the menu's template from disk and places it in memory where it becomes a memory template. Until the user accesses another menu on your system, TBBS will use this memory template to build a user template each time the user returns from a function. It is at this point that the access control system comes into play.

STOP HERE FOR ACCESS KEYS BEFORE PROCEEDING TO SYSTEM!
TO BULLETINS
DOOR UNLOCKED COME ON IN!
TO PRODUCT ORDERING
TO FILE AREAS
TO MESSAGE AREAS

Once the memory template is established, TBBS employs its access control system to dynamically construct a user template. For each entry in the menu, TBBS compares the entry's access control settings against the user's access authorization settings. If the user qualifies for an entry, it's added to the user template. If not, the entry is skipped and is not added to the user template.

In this way, the original menu template you designed is custom tailored to each user, based on their specific userlog access authorization settings. Once the user template is constructed, the entries on that menu template are processed by TBBS, allowing the user to interact with the system. We'll discuss this interaction in more detail later.

It may seem at first that this would be a lengthy, time-consuming process. As detailed as it is, however, TBBS does it for you automatically in the blink of an eye. The user will notice no delay while this customization occurs – it's as though the system was already designed specifically – just for him.

Menu Entries Described

Each menu template you build contains one or more *menu entries*, like individual choices on a list. Individual user access to each entry is controlled by you, so that the presentation of your system is customized for each user.

It's now time to examine these entries in detail to understand how you use them. Each menu template entry is composed of four separate parts:

- Access Control
- Method of Access
- Display
- Function.

Access Control

Access control is a method of screening which users can access any one particular menu entry. Control can be either open (anyone can see and access this entry) or restricted (the user's access control settings must qualify in order to see and access the entry).

Method of Access

This is how the menu entry is activated. It can be by an active keypress (the user sees the entry, and presses a key to activate its function) or can be "auto-executing" which means it happens automatically when the menu template is loaded by TBBS for the user.

Display

This is the part of the menu entry shown to the user. After the menu loads and is customized for the user, the display portion of each qualifying entry is shown to the user. The user then selects which one of the entries he or she wishes to activate.

Function

As the name suggests, this is the actual function or operation that a particular menu entry performs. In TBBS, a function is a fairly low-level operation, such as entering an electronic mail message, downloading a file from a particular directory on the disk, or displaying the contents of a text file to the user. Functions are the "brass tacks" of what the system *does*.

In TBBS, functions are called *menu types*. Menu types are all referenced by number. The complete list of menu types is shown in Reference Manual Chapter 6, Menus.

You should now understand that you use menu entries to build each menu on your system, and that each menu entry you build is composed of four parts. You should also understand that these entries together represent a menu template, and that menu templates build your custom TBBS.

The TBBS Access Control System

As you've already learned, TBBS provides access control in two primary areas, at logon and inside the system itself. We'll discuss the logon aspect of access control in Chapter 2 of this guide.

The single most important area of access control is within the system itself. Because "the system" is composed of the menu structure you create, this area of access control is an important aspect of your task as a TBBS System Designer.

The access control system is actually two independent systems in one – privilege level, and access flags. Both work together to provide you with a high degree of control over your system, and which of its functions any particular user can access.

Privilege Level

Privilege levels are an *inclusive* access control, and are hierarchical in nature. Privilege levels are simply a number between Ø (zero) and 255. The higher the number, the higher the access. The higher levels are *inclusive* of the lower levels, meaning that access is being added to what was already available to the user.

Access Flags

Access flags are a *mutually exclusive* access control. There are 32 of these flags, and they are best thought of as "on and off" switches. The flags can be used alone or in combination, allowing you to selectively *include* or *exclude* particular users or user groups.

The privilege level and access flags are independent from one another, but work in conjunction. On an entry-by-entry basis, both access control methods work to automatically and dynamically customize the system for each user. It's the access control system which makes this automatic customization possible.

Methods of Access

You've learned that functions on your system are presented as entries on menus, an that you build these menus to customize your TBBS. There are two ways that these menu functions can be activated.

Active Keypress

This is by far the most common method of access. A list of menu entries is displayed to the user from the user template. (Remember that a user template is a menu that TBBS has automatically customized for a particular user.)

As the menu entries are displayed to the user, he or she may at any time make a valid selection by pressing a single key. You as the System Designer decide what keys activate which entries. Keys are usually letters, numbers or printable punctuation. The user simply presses the key that corresponds to the entry he or she wishes to activate, and TBBS immediately performs the function selected.

Auto-Execute

Auto-executing menu entries are used in special situations where you want a TBBS system function to be performed automatically each time a menu is loaded and processed by TBBS. Auto-executing entries are useful in many situations. They are like any other menu entry except that they are not presented to the user as choices on a menu (because no active keypress is required to activate them).

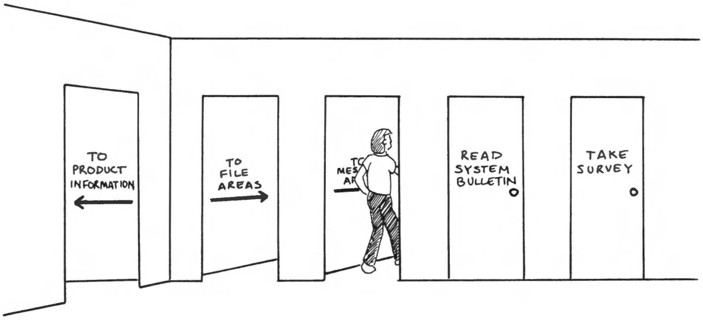
See **Menu Processing** at the end of this chapter for a detailed explanation of exactly when an auto-execute menu entry will be activated.

Functions (Menu Command Types)

You've learned that the building block components of a TBBS system are called menu entries. You put various menu entries together to form menu templates, and those menu templates describe the system itself.

Most of the capabilities people associate with a bulletin board system are really just menu entries under TBBS. Menu entries enable such activities as "enter a message in this area", "display this text file to the user" and "enter this file download area" for example.

There are two categories of menu entries. *Navigational entries*, which move the user from one menu to another, and *functional entries* which perform a specific operation. Both navigational and functional entries are implemented in the menu template in exactly the same manner – their only difference is the action they perform.



Navigational entries are like a corridor which takes you someplace else in the system, while functional entries are a final destination to a particular operation (function).

Navigational Entries

Navigational entries, as the name suggests, allow users to navigate your template menu structure. Navigational entries take users from the current menu to other menus – forward or backward. Often you may group several menus to form a tree structure so TBBS also provides a navigational entry to go directly to the top-most menu of the tree.

Functional Entries

Functional entries are the raw operations that TBBS is capable of performing. They include reading and entering messages, downloading files, answering a questionnaire, and so on. Most users will perceive these functional entries as the heart of your system, because they're the actual "nuts and bolts" of what they do on your system.

Menu Processing

You've learned that TBBS customizes the menu display to each user on your system. You also know that the access control system uses a memory template to build the custom user template.

There are four steps involved in the processing of a memory template into a customized user template. Let's look at these steps in more detail.

Step 1 Privilege and access flags required by the menu are compared to the current settings of the user, filtering out unqualified users on an entry-by-entry basis until the user template is constructed.

Step 2 Any auto-executing menu entries from the **user** template are taken in order, and marked as "executed" in the **memory** template. After each auto-executing entry, the flow returns to Step 1, where the user template is reconstructed from the memory template over again. This is because some types of menu entries can change the current user authorizations, and may affect which entries the user is or is not qualified for. The user template is rebuilt each time to keep authorizations in synch.

Note, however, that auto-executing menu entries occur once, and are marked as "executed" in the **memory** template. So even though the **user** template is rebuilt each time, auto-executing entries occur once – until the **memory** template is rebuilt from disk, which occurs only if the user activates a navigational menu entry.

Step 3 If auto-executing entries (if any) don't change the menu via a navigational entry then the menu title and text of the various menu entries is displayed to the user, and TBBS waits for the user to press a key.

Step 4 Any selected command is executed. If the user selected a functional entry, the function is executed, and flow is returned to Step 1. If the user selected a navigational entry, then the new disk template is loaded, and the flow returns to Step 1.

This process occurs over and over again, until the user logs off the system. This repeating process is called a *user session*. The things that occur during the user session *are* your system.

As complex as the process appears at first, you can see quickly that it's really very simple at its core. It's a defined process that simply repeats itself many times during the user session. It's your efforts as the System Designer that control what operations happen during the process, but the process itself is static, and is what happens in every TBBS installation.

Understanding the Logon Process

In the previous chapter we studied the importance of menus to TBBS. You've learned that menus are the heart of your system. But before any user sees a menu, they take part in the logon process – the act of connecting to and logging onto your system. This chapter will describe that logon process.

Logon involves users (callers to your system) connecting to your system through any one of a variety of methods. The user enters their unique user ID and password, and after a series of authorization and access checks, is allowed to interact with your system.

The first part of the process is the physical connection. With TBBS, you have a great deal of flexibility with connections. The most common type of connection is a modem, connected to a regular telephone line, and allowing people to use their modem to dial your system and connect. There are other types of connections however, which allow you to address special connectivity needs. Regardless of the physical connection, once the call is answered by TBBS the user makes the transition from connection to logon handling.

TBBS begins the logon process by displaying an opening banner You design the appearance and content of this banner.

ALL GUESTS MUST CHECK-IN WITH THE ...D!
YOU CAN PASS, BUT BE SURE TO READ THE INFORMATION FILES...
ID

After the banner is displayed TBBS prompts the user to input their unique user ID. Normally this ID is the user's name but there are other options. TBBS requests this information so that it can decide what portions of the system (if any) the user has access to, and for how long. TBBS also maintains a password for each user so that it can verify the person logging on with a particular user ID.

The user IDs and passwords are kept in a list called a userlog. Attached to each user's ID is a variety of information – terminal settings, access authorization settings, limitations on number of calls and call duration, the width and length of the user's screen, and much more. You maintain the userlog with a userlog editor, which allows you to add or delete users, authorize users, or change any user's parameters.

When a user logs on, TBBS will attempt to locate the user ID in its userlog. If a user cannot be located in the userlog, TBBS will handle the situation whichever way you choose; you can have the system disallow the caller from access entirely, disconnecting unknown callers. Or, you can have TBBS automatically add them to the userlog, and provide them with default access authorization settings which you configure.

Once the user is logged onto the system, there is other information that is displayed to them. You have a great deal of control over this part of the presentation. It includes options to list any messages waiting for the user, various user statistics, and bulletins or news files.

After the logon handling presentation is complete, the user exits the logon process, and the menu process has begun. TBBS loads its "top menu" and the user's interaction with your system continues until logoff (when the user disconnects from the system and terminates the connection).

Although the logon process seems simple, you have a variety of options to control and tailor the process.

The Logon Process Explained

The logon process begins when a user makes a physical connection to TBBS. It continues by identifying the user and deciding to permit them on the system (or not) and displaying various information. During logon the user is taken through a variety of configurable presentations. Let's look at this process in more detail.

Physical Connection

Users can connect with a TBBS system in a variety of ways. By far the most common method of connection is through a dial-up modem connected to a standard voice telephone line. This method allows almost anyone with a modem and a telephone to connect to your system, and provides maximum accessibility to your users.

Several other connection methods are available, however, which exist to meet special connectivity needs. Of these other methods, a hard-wired connection is probably the most common. This allows you to connect another computer or data terminal to the system via a standard serial cable and null modem adapter. X.25 packet switch network connections are possible (through a PAD) as well as ISDN telephone connections and data PBX switched connections. All of these physical connection methods use an RS-232 serial interface. TBBS requires all connections to come in to the system through an RS-232 serial port.

Logging On

Once a physical connection is established and TBBS becomes aware of a caller's presence logon handling begins. If your system is configured to handle multiple languages, then the very first step TBBS will take is to prompt the user to select their desired language from a list of those you have configured. Once selected, this language remains active for the user for the remainder of their call.

Afterward (or initially if your system uses a single language only) TBBS will display the banner. The banner can be a text file you've created, or you can use CEDIT (the TBBS configuration editor) to configure a brief logon greeting. You can also use both together if you wish.

After the banner display, the user is prompted for their unique user ID. As the System Designer, you have several options here. Normally users enter their own name as a user ID. You can have them enter it as a single name, or as first and last name separately. You can also choose not to use names at all, but instead to use a generic "user code" which can contain virtually any printable characters.

Once the user has entered their user ID, TBBS will attempt to locate that ID in its list of users – called a *userlog*. If the ID is found in the list, then TBBS will prompt the user for a password (if any) to confirm the identity of the caller.

You have two options in configuring TBBS to handle users that cannot be located in the userlog. If you choose, you can have TBBS display a brief text file to the user, then disconnect them from the system. In this case, the user has no access to your system if his user ID is not located in the userlog. Your other option is to have TBBS automatically add the user's ID to the userlog, and authorize them with a default set of access authorization settings. If you select this option, then you can also select the particular access authorization settings you want to use as the default. You can also configure a variety of other options, such as time limits, for these new callers.

Userlog Settings

TBBS maintains a variety of information about each user in its userlog. Each user ID has its own account or *userlog record*. In each record, TBBS keeps the user's ID, password, calling location, terminal settings, access authorization settings, limits and dozens of other settings and options.

In cases where you allow new users to access your system (a new user is one TBBS couldn't locate in the userlog and automatically authorized), the user will be asked for some of the userlog record information, such as their terminal settings. As the System Designer, you also have the option to default much of this information so the user is freed from entering it. If you don't allow new users, then you must add new users to the userlog yourself, and enter the various settings and options in their userlog record.

Adding, deleting or modifying existing users is all accomplished with the userlog editor – ULEDIT. With ULEDIT, you can view and modify any information about any of the users on your system.

Passwords

One of the most important parts of the logon process is password verification. This is how TBBS confirms the identity of the caller. A caller's user ID is available to other users on the system, and is the user's unique identifying label. To keep others from logging on and using the ID of another user, however, password verification is needed. This step is optional, but is highly recommended for all but very specialized applications.

If you allow TBBS to automatically authorize unknown callers, then you can also have TBBS request a password for future logons.

Users logging onto the system who are known to TBBS (their ID is found in the userlog) will be prompted to enter their password after their user ID is entered and located in the userlog.

Logon Display

Once a user is signed onto the system successfully, you have a variety of options which custom tailor the display information. This information includes news or bulletin files (text files you create), new user information (shown to unknown callers TBBS has automatically authorized, if you have that option enabled), user statistics (how many times the user has called, how many messages they've read, and so on), and finally you can have TBBS list any messages waiting for the user and optionally allow the user to read and respond to them.

Much of the logon display information is optional and can be enabled or disabled by you, the System Designer. Some of the displays are text file based, so you can create the files and their contents to your exact format and needs.

Passing Off to the Menu System

Once the logon process is completed and the various displays are finished, the menu process begins, and control passes from the logon process to the menu system. The top-most menu on the system, which TBBS calls menu ØØØØ, is loaded and displayed to the caller. All control from this point on is within the menu structure, and the logon process is complete.

What Happens Before Logon?

You've learned that the logon process involves several steps, and that you have control over most of these steps. Let's take a look at the options you have to customize the logon presentation, beginning with the steps prior to the user logon. To make this process easier to understand, let's follow an actual caller – Steve – through a sample logon.

Connection

The very first step is that Steve connects with your system. In our example, Steve is calling you on a modem line. On your end of the connection is a standard, dial-up modem connected to a regular telephone line. Steve also has a modem connected to his computer which he has used to call your system.

Other Possibilities

Users could also connect through a packet switch network such as BT Tymnet or SprintNet if you were a subscriber to those services and had a PAD connected to your TBBS. State-of-the-art ISDN connections are also possible. For users located on the premises, connections can be made through dedicated RS-232 serial lines directly to another computer or a data terminal. Many PBX telephone systems provide serial data connections, which can also be used to connect to your TBBS.

With TBBS, you have many options for physical connections into the system, even though dial-up modems are by far the most commonly employed method.

Speed Restriction

With TBBS, it is possible to restrict access to certain lines by connect speed (baud rate). If you have chosen to limit speeds on Steve's line, speed checking is the first operation TBBS performs after the connection.

The speed at which Steve is calling is permitted by the system on his line, so Steve's call progresses to the next step.

Other Possibilities

If you choose to, you can selectively restrict users calling at particular data speeds from connecting to any one particular line (or group of lines) either permanently or during certain hours of the day. This optional configuration is performed on a line-by-line basis, in four speed groups:

- 300 bps (covering 110, 150, 300 and 600 bps callers)
- 1200 bps (covering 1200 bps callers only)
- 2400 bps (covering 2400 bps callers only)
- 9600 bps (covering 4800, 9600, 19200 and 38400 bps callers)

Access to any particular line in any of the four speed groups is configured on a time range basis. You can select the start time and stop time during which calls for each speed group are allowed or you can allow them always or never.

If a user connects at a speed that has been restricted by you, then a text file will be displayed and the caller will be disconnected from the system (no access will be given). You create and compose these text files. More information about restricting speeds can be found in Chapter 3.

Language Request

The next step in the logon process is the language request. By default, TBBS uses one language (English). It is possible to configure TBBS to use more than one language which you have translated and configured, in which case TBBS will query the user to find out which language they wish to use for the duration of their call. In our example, TBBS uses just one language, so the language request step is skipped, and Steve moves to the next step.

Other Possibilities

TBBS features multi-lingual capability for situations where you have callers who speak a variety of languages. Even in cases where only one language is required, TBBS allows you to translate and use a language other than English (which is the default, and is supplied with TBBS).

Language text is used in several places within TBBS: internal text, menus, and system text files. You may customize each text string TBBS uses through the EDLANG language file editor..

When multiple languages are configured, TBBS will query the user about which language they want to use. All languages you have configured will be listed, and the user chooses one. (If only one language is configured, this step is skipped.) This selection will impact several areas of the system, and at each point in the logon process flow we'll mention where language selection has an impact on your options.

CEDIT Logon Greeting

The next step in the logon process is the CEDIT Logon Greeting. This is a small amount of text that you can optionally configure in CEDIT (the TBBS configuration editor). If you have created this text and placed it in CEDIT, Steve will see it displayed to his screen.

Other Possibilities

You may also omit the CEDIT greeting text if you wish, simply by not configuring it in CEDIT, or by removing any existing text from the Logon Greeting option in CEDIT.

Multi-Lingual Impact

CEDIT text is not language specific. If you want to have language specific greetings, use a banner file instead.

Banner Text

The banner text is the next item to be displayed. This is an optional step. If you choose to configure the system to display a banner, you must also create the file to be displayed. The TBBS default is to use a banner text file, named SIGNON1.TXT, which is displayed to users connected on all lines. Steve sees the contents of this file displayed to his screen, and moves to the next step.

Other Possibilities

By default TBBS uses a single banner text file on all lines, named SIGNON1.TXT. You can change the name of the file, or choose that it not be displayed at all. You may optionally configure unique banner files for one or more lines, and you may selectively enable or disable the banner on one or more lines.

Multi-Lingual Impact

Because the banner file is a text file, you may follow the unique multilingual text file naming convention (see Chapter 4) for any banner files if your system is configured for multiple languages. This means that each banner file you configure will be customized for the language selected by the user.

Summary

That completes the steps taken before the caller actually logs onto the system. You've learned that callers connect to the system, and can optionally be speed restricted. You've also learned that the user can select their preferred language (if you have implemented multi-lingual capability), and that up to two greetings can be displayed. Above all, you've learned that with TBBS you have a great deal of flexibility and control over nearly every element of the system.

Now, let's proceed to the next area of the logon handling process, identifying the user and allowing or disallowing access.

What Happens During Logon?

Logon is the process where the user is identified and logged into the system and begins interaction with the system itself. It is composed of several smaller steps, which we'll now take a look at.

Logon Prompting

The next prompt Steve sees is a request to enter his name. The system Steve uses prompts for his full name as a single entry. Steve types in his full name, and presses Enter. He is then taken to the next step in the logon process.

Other Possibilities

You can have TBBS prompt for a user ID in any one of three ways:

- first and last name (as separate items)
- full name (as a single item)
- user ID (which can be any alphanumeric identifier).

The method you select changes only the way TBBS prompts the user – all user IDs are physically stored in the userlog file in exactly the same manner regardless of format.

Multi-Lingual Impact

The exact prompt shown to the caller for their name – regardless of which one of the three prompts is used – can be configured as you wish for each language using the EDLANG language editor.

Userlog Lookup

TBBS next attempts to locate the ID the user typed in by looking it up in the userlog file. Steve can be located because he's an existing user of the system, so he's taken to the Location Verification step.

Other Possibilities

You have two options for the handling of users that cannot be located in the userlog:

1) You may choose to disallow new users. If you do this a file named NEWUSER.TXT is displayed and then TBBS disconnects them. No access to your system is allowed to unknown callers when TBBS is configured in this way.

2) You may choose to allow unknown callers. If you do TBBS will add them to the userlog, and authorize them with a set of default access authorization settings you define. These options are covered in more detail in the **New User Handling** step.

Location Verification

This step has two different modes of operation – one for callers who have been located in the userlog, and one for those who have not.

For callers that are located in the userlog you can have the system optionally prompt the user to verify their calling location stored in their userlog record. If you have configured TBBS to do this then the system will display the calling location on record, and confirm with the caller that it is in fact correct. If the user responds with a “N” TBBS will go back to the logon step and request the user’s ID over again. Steve answers “Y” when asked if he’s calling from Denver, Colorado, so he proceeds to the password verification step.

Other Possibilities

If you allow unknown callers to be automatically added and authorized, and the caller cannot be located in the userlog, TBBS will prompt the user to enter their calling location. By default, TBBS asks for the user’s city and state; the exact prompt can be changed using EDLANG. The user then proceeds to **new user handling.**

Multi-Lingual Impact The exact prompts shown during location verification can be modified to be language specific by using the EDLANG program.

New User Handling

Because our example user, Steve, was located in the userlog by TBBS, this step is skipped for him. But if the caller was not found in the userlog, and you have configured the system to allow such callers, this step is taken. *If you have configured TBBS to disallow unknown callers, this step is never taken because the user has already been disconnected.* The new user handling step is actually composed of several smaller steps.

Userlog Addition

TBBS will first add the new caller to the userlog by creating a new record for him. This allows TBBS to locate this user the next time they call the system.

User Authorization

TBBS will then authorize the user for access based on the default access authorization settings you have configured. In TBBS, these are called the *new user defaults* and they will be given to all new users of your system automatically when they call. The settings include privilege level, access flags, and call limits.

Prompt for User Profile Settings

Using CEDIT, you may optionally configure TBBS to assign a new user various other non-authorization related settings. These include terminal settings, screen width and length settings, and usewr preferences such as ANSI and Graphics. For those settings you have not chosen to default, TBBS will prompt the user with a series of questions and enter their responses in the userlog record.

After TBBS creates a userlog record, the user then advances to the password verification step.

Multi-Lingual Impact All of the prompts can be changed to be language-specific by using the EDLANG program.

Password Verification

This step has two modes of operation: one for users that were located in the userlog, and one for those that were not be located and were automatically added to the userlog.

Known User

Our example user, Steve, is a known user because he was located in the userlog. As such, TBBS prompts Steve for his password. Steve has three tries to get it right – after the third attempt, TBBS will disconnect. After this step Steve moves to the Limits Check step (if he gets his password entered correctly),.

Unknown User

If the user was not located in the userlog, and was added to the system and authorized, you can optionally have TBBS prompt them to enter a password. If you have configured the system in this manner, TBBS will query the user for the password they desire. A brief instruction screen will be presented to the user. (The instruction screen can be edited using EDLANG, along with the exact prompt shown to the user.) After entering the password, TBBS can optionally confirm the password by asking the user to enter it again.

Other Possibilities

Password prompting is optional. If you had added Steve to the userlog, but left the password portion of his userlog record blank, TBBS would skip the password verification step.

Using CEDIT, you can select two options for display of password entry – as-is (where the letters of the password are displayed as they are typed by the user) or secured (where asterisks (*) are displayed as the letters are typed). The secured method prevents passwords from being obtained by someone “looking over the shoulder” of the user as they call the system, and is the recommended method for additional security.

Multi-Lingual Impact

The exact text of the password verification and entry prompts can be made language-specific by using the EDLANG program.

What Happens After Logon?

Limits Check

The next step which takes place is the limits check. As always with TBBS, you have a great deal of control over access to your system. At this point, TBBS checks four separate settings to determine whether a particular user is authorized:

- Does the caller have adequate Billing Class Ø time?
- Is the user's access expired by calendar date?
- Has the user exceeded his number of calls per day limit?
- Has the user exceeded his allowed time per day limit?

The Class Ø check is performed if you gave the user a pre-authorized time limit. This optional feature (set with ULEDIT) allows you to pre-authorize a particular user for a total number of hours and minutes. When that amount of time is exhausted (or the date has passed), he can no longer access the system.

If you set an expiration date for a user and that date has passed TBBS treats the user as if he had been given a Class Ø time limit and used it all.

If any of these checks reveal that the user has exhausted a preset limit, the file NOAUTH.TXT will be displayed to the caller, after which they will be disconnected and disallowed from accessing the system until their time or limits are reset.

Other Possibilities

> You can choose not to use pre-authorized time limits, and/or allow unlimited calls per day or unlimited time per day.

Multi-Lingual Impact

Because NOAUTH.TXT is a text file, you may follow the unique multi-lingual naming convention for the file (if you have configured the system for multiple languages). This means that you can optionally create separate NOAUTH.TXT files for each language the system is configured to support.

A3 Flag Line Bumping Check

The next check to take place is that of "A3 bumping." As you'll learn later, it's possible to restrict users from access to certain lines, unless they are marked as "registered" by you. Steve is marked as registered so he continues to the next step of the logon process.

Multi-Lingual Impact

Because the A3BUMP.TXT file is plain text, it uses the standard TBBS naming convention for text files in multi-lingual installations.

User Status Messages

The next step in the logon process is the displaying of any CEDIT status messages. These messages are all optional, and are configured using CEDIT's Logon Options screen. On our example system Steve sees the following messages in order:

```
TBBS Welcomes STEVE
Calling From DENVER, CO
Your last time on was 07/10/91 19:28
You have read through message 4748
Current last message is 4756
You have called this system 29 times before
You are caller number 6906
```

The first two lines cannot be omitted – the welcome line, and the caller location display. The exact text of these messages can be edited using EDLANG. The other messages can be selectively enabled or disabled with CEDIT. In order, they are:

- Date and time of last logon
- Number of last message read
- Number of current last message
- Number of calls user has made to the system
- Number of calls made to the system total by all users
- Display of current active combined board list for user

Multi-Lingual Impact

The wording of the text displayed for each status message can be modified using EDLANG to be language-specific.

News File Display

The next step of the logon process is the display of the news file if one is present. By default, this file is named SIGNON2.TXT, which we will use in our example. We have this news file configured to display each time the user calls, so our sample user, Steve, sees this file displayed to his screen, then he advances to the next step.

Other Possibilities

The news file can be named differently than SIGNON2.TXT if you wish. You can also use one main news file, or customize news files on a line-by-line basis. You can also have the news file displayed once to each caller (until it's changed) or each time the user calls.

Multi-Lingual Impact

Because the news file is a standard text file, you can use TBBS' standard text file naming convention to create a news file for each specific language you have configured.

New User Information Display

If the user was just automatically added to the userlog and authorized by TBBS for access, then this step will be taken. Since Steve is not a new user, he skips to the next step of the logon process. If the user is new to the system, the file NEWUSER.TXT is displayed if it exists and if it is enabled via CEDIT.

Other Possibilities

It is the TBBS default to use a single new user info file on all lines, named NEWUSER.TXT. You can change the name of the file, or opt that it not be displayed at all. You may optionally configure unique new user info files for any one or more lines, and you may selectively enable or disable the new user info file on any one or more lines.

Multi-Lingual Impact

Because the new user info file is a standard text file, you may use TBBS' standard text file naming convention to customize a file for each language configured on your system.

List Waiting Messages

The next step is the listing of waiting messages. We have configured our example system to list all messages waiting for the user. Steve will see messages addressed to him that he has not read automatically listed to his screen. The listing includes the date and time of the message, who it is to and from, the subject of the message, the message number, and the name of the board on which the message is posted. (This information is known collectively as the "message header." NOTE: Some parts of the message header display are optional.)

After listing his waiting messages, Steve moves to the next step.

Other Possibilities

> You can choose not to have waiting messages listed if you wish. This may be done either system-wide, or on a line-by-line basis.

Read Waiting Messages

NOTE: If you do not allow the listing of waiting messages (see previous step) then this step will **not** be taken!

After Steve's messages are listed, he is given the option of reading and replying to the messages immediately. Steve chooses to read his messages now, after which he moves to the next step.

Other Possibilities

> You can choose not to allow the immediate reading of waiting messages if you wish. This may be done either system-wide, or on a line-by-line basis.

Transfer User to Menu System

This is the last step of the logon process – releasing the user to the control of the menu system. If the user successfully arrives here, he has been given access, and is free to use the system under the constraints and limits you have selected.

The passing of control from the logon process to the menu system involves the loading of menu ØØØØ, the top level menu for the system. The user is then released from logon process control.

Multi-Lingual Impact

In multi-lingual installations, the name of the top level menu is language dependent to allow the menu structure to be language-specific. The name of the top menu is ØØLn, where “n” is the is letter which corresponds to the language selected by the user. (Language “A” uses menu ØØØØ.)

Usage Control, Accounting and Tracking

So far in this guide, we've been referring to you as the System Designer. But once your system is designed and implemented, you take over the role of System Operator – the manager and caretaker of your new TBBS system. (System Operators are often called "Sysops" for short.)

With TBBS' control, accounting and tracking capabilities, you can regulate, account for and record system usage on a user-by-user basis, using a variety of approaches. You can control call and file transfer volume, and can manage the length of incoming calls to maximize system availability. You can also control and account for usage of the whole system, or individual portions thereof. You can also record many areas of system usage, both globally through audit trails and on a user-by-user basis.

All good business managers need information to effectively carry out their decision making and supervisory tasks. Therefore, in order for you to be an effective manager of your TBBS, you too need information about its operation and usage. This information takes the form of usage accounting and tracking data, and TBBS provides several means to capture this information.

Tracking data generated by TBBS also helps in your role as system manager and designer. For example, flaws in the design of your system may be reported by some of the tracking mechanisms, dictating a change in your design.

With control, accounting and tracking, as everywhere within TBBS, you have a number of options open to you.

The Options for Control, Accounting and Tracking

The TBBS control, accounting and tracking options offer an extensive range of capabilities. Not all controls will be of interest to you, and some of the accounting data may not be very useful in your situation. Some of the various tracking capabilities are built-in and are "on" all the time, while others must be explicitly enabled by you before they are active.

Control

The primary means of usage control is implemented on a user-by-user basis. Each individual user can be independently controlled through a series of limits. These limits include the number of minutes the user may consume per call and per day, the number of calls the user can make each day, and the number of bytes they may download each day. The four limit options can be individually set to a particular value, or left in an unlimited state.

Users may optionally be assigned an expiration date. If the user accesses the system beyond the designated date, they will be shown a message you create, then disallowed from accessing the system at all.

TBBS also provides a system of controls for regulating access time on a global (system-wide) or area (corresponding to a particular section of your system) basis. Using this method, you can pre-authorize a user for a particular amount of time, and once this time is exhausted the user can no longer access the specific area(s) you have controlled (or possibly the system itself).

Also available is a method of designating a group of your users who will be limited to using the system only on a particular line or group of lines. Finally, you can limit access on specific lines to particular data rates during particular time periods if you wish.

Accounting

Whether or not you actually charge a fee for access to your system, accounting data can provide important feedback on the operation of your system. As part of each user's userlog record, TBBS always maintains the user's number of calls (total and daily), connect time (total and daily), downloaded bytes (total and daily), and uploaded bytes (total).

TBBS can also be configured to maintain time usage counters (hours and minutes) on a specific area basis (corresponding to a particular section of your system) in up to 9 categories called *billing classes*.

Tracking

Tracking refers to the maintenance of audit trails. Audit trails allow you trace through various aspects of system usage.

The primary means of maintaining an audit trail in TBBS is through the use of a log file. All major system events are logged, and you may configure several other logging options to track activity you are interested in. The log can be disabled, or if enabled, can be routed to a printer, a disk file, or both simultaneously.

Real-time conferences (inter-user chat areas) can be optionally logged to a disk file as well, providing a running sequential audit trail of all conversations that take place in each defined conference area.

TBBS maintains a record of the last 10 calls (date and time) made by each user of the system. This data is actually part of each user's userlog record, and is kept automatically by default.

Finally, the last 127 callers to the system are also tracked and can be displayed online through a special menu entry command type.

Usage Control in Detail

You have learned the basic areas in which you have control over system usage. You have also learned how TBBS keeps track of system usage and what audit trail facilities are available to you. Now, let's explore these options in detail, starting with usage control.

User Limits

On a user-by-user basis, as part of each userlog record, you can control four simple but powerful limits. These limits are by far the most commonly used methods of controlling system usage, and are employed on nearly every TBBS installation.

Because they're stored in the userlog, you use ULEDIT (the userlog editor) to view or modify these settings. See Reference Manual Chapter 11 for details.

The first limit you may set is minutes per call. This controls how many minutes the user may stay online during any one particular call to the system. When the time limit is up, the user is disconnected. A warning message is displayed to the user at 2 minutes prior, then again at 1 minute prior to being disconnected.

The second limit is the number of minutes per day, which controls the total number of minutes (across all calls) that the user may consume each day. The counter is reset on the user's first call after midnight (the first call of a new day).

The third limit you may set is the number of bytes that the user may download per day. When the user attempts to download a file which will exceed this limit, a warning message is displayed. (You can have particular download commands override this limit automatically if you wish – controlling which files are restricted by this limit.) The counter is reset on the user's first call after midnight. The fourth and final limit is the number of calls the user may make to the system in any one day. When the user attempts to make a call into the system which would exceed this limit, a warning message is displayed, and the user is disconnected. (This counter is reset on the user's first call after midnight.)

Expiration Date

TBBS provides you the option of setting expiration dates for each caller if you wish. The TBBS default is that no users have expiration dates set. If you choose to set an expiration date for a particular user, that user will be able to use the system normally (within the constraints set by other usage controls) but access to the system will be prohibited after the set date. If a user calls the system after that date, a text file named NOCLASSØ.TXT will be displayed (if it exists). Once the expiration date has past, you will need to reset it before the user can have access again.

Other Possibilities

> Expiration date handling can also produce access flag changes in lieu of prohibiting access. If configured in this manner, NOCLASSØ.TXT will not be displayed – the access flags of the user are simply modified as you configure, and the user remains authorized for access.

Multi-Lingual Impact

Because NOCLASSØ.TXT is a text file which you create, you can use TBBS' multi-lingual naming convention to customize this file for supported languages.

Regulating Access Time

TBBS provides a mechanism known as "billing classes" to control (and account for) system usage on a system-wide or section-by-section basis. Billing classes are numbered Ø through 9 and billing class Ø is the global billing class – meaning that it applies everywhere on the system. Billing classes 1 through 9 are Sysop-configurable, and are assigned by you on a menu-by-menu basis to indicate areas you wish to track or control.

If you pre-authorize a user with billing class Ø time (in ULEDIT) then the user will only be able to spend the number of hours and minutes on the system that you have authorized.

If you pre-authorize time within other billing classes (1 through 9), the user will only be able to spend that amount of time within any menu which is assigned that billing class. NOTE: Time is counted from the moment of entry to a menu until the user leaves that menu.

If a user exhausts their time within any billing class a text file named NOCLASSn.TXT will be displayed (where "n" is the billing class). The user will not be able to access any menu with that billing class assigned to it, nor any function inside that menu until you re-authorize them.

Other Options

Billing class time expiration can also be handled an alternate way.

Using CEDIT, you can configure billing class expiration to modify access flags to provide restricted access for the user. If you configure billing classes in this manner, then the NOCLASSn.TXT text file will not be displayed. Instead, the designated access flag changes will take place, and the associated billing class limit will be reset. The user's ability to navigate and use the system after this change is designed and configured by you.

If billing class Ø expires, however, the user will still be logged off the system. When the user calls back, their access flags will be changed as you have configured, and the user will be allowed to use the system normally, within the constraints of other usage limits you have put in place.

Billing classes also have accounting capabilities which are discussed later.

Multi-Lingual Impact

Because the various NOCLASSn.TXT files are text files, you can apply the standard multi-lingual text file naming convention to customize the file for each configured language.

Line Group Usage Control

TBBS provides you with a method of splitting your user base into two groups, and limiting the lines on which one of the groups can access your system. The primary application of this feature is to limit new users or other "reduced access" groups to certain lines. This feature is called "A3 Flag Line Bump" due to the way it operates.

This capability revolves around the "full access" flag setting in each user's userlog record. This flag defines which of the two groups a particular user belongs to – those who are limited to access only some lines of your system or those who are not.

The second key part of this functionality are the A3 flags and their relationship with incoming lines. Within CEDIT, you can optionally have a line or line group set selected A3 flags. When a user calls in on a line configured this way, his A3 access flags are changed to what you configured for that line. (NOTE: This feature does not have to be used for usage control exclusively; you may use it to control what menu entries are available on what lines, for example.)

Users who have their **full access** flag set in the userlog will be able to access all lines without intervention by this usage control method. Users who **do not** have their full access flag turned on may be "bumped" from the system depending on whether or not their A3 flags match the setting of the A3 bump flags you set using CEDIT.

For example, let's assume you wanted unregistered users to be able to access line 1 only, and be bumped from all other lines. A3 flag 1 (A3[1]) is available, so we'll use this flag to indicate restricted lines. Within CEDIT, all lines (except line 1) will be configured to set the A3[1] flag on. Also with CEDIT, you configure the A3 bump setting to bump users with A3[1] on. Only users with the full access flag set can then use lines other than line 1. If an unregistered user attempts to call on any line other than line 1, they will be shown the text file A3BUMP.TXT and then will be disconnected from the system and not allowed access.

Multi-Lingual Impact

Because A3BUMP.TXT is a text file which you create, you can use TBBS' multi-lingual naming convention to customize this file for supported languages.

Limiting Data Rate Access

The final means of controlling usage of your TBBS is through the limiting of calls to particular data rates, during particular times, on particular lines.

For example, you could selectively restrict 1200 bps callers from dialing into lines 2, 3 or 4 from 11:00am to 3:00pm. You can control which data rates are restricted, on which lines, and during which hours. The default is to allow all speeds on all lines at all hours. This control is implemented through CEDIT, as a part of each line definition. Speed may be limited by four speed groups as follows:

- 300 bps (covering 110, 150, 300 and 600 bps callers)
- 1200 bps (covering 1200 bps callers only)
- 2400 bps (covering 2400 bps callers only)
- 9600 bps (covering 4800, 9600, 19200 and 38400 bps callers)

If a user calls the system with a restricted combination of line, date rate, and time, a text file will be displayed, and the caller disconnected and not allowed to access the system. The files themselves are customized for each independent data rate – not data rate group. The files are named as follows:

BADØ11Ø.TXT	(for 110 bps calls)
BADØ15Ø.TXT	(for 150 bps calls)
BADØ3ØØ.TXT	(for 300 bps calls)
BADØ6ØØ.TXT	(for 600 bps calls)
BAD12ØØ.TXT	(for 1200 bps calls)
BAD24ØØ.TXT	(for 2400 bps calls)
BAD48ØØ.TXT	(for 4800 bps calls)
BAD96ØØ.TXT	(for 9600 bps calls)
BADØ19K.TXT	(for 19200 bps calls)
BADØ38K.TXT	(for 38400 bps calls)

Multi-Lingual Impact

The various BADxxxx files are standard text files. As such, you can apply the standard TBBS text file naming conventions to create a unique version of each file for each language configured.

Usage Accounting in Detail

As you've learned, usage accounting allows you to gain an understanding of system usage on a user-by-user basis. Whether you choose to actually charge a fee for the use of your system or not, accounting for usage lets you know who is using your system, for how long, and to what extent.

Basic User Statistics

For every user on the system, TBBS maintains a set of very basic but critically important accounting information. These four accounting tools are part of each userlog record, and are automatically maintained. They are:

Number of Calls

A count of the number of calls the user has made to the system is kept for both total calls (calls since the user was added to the system) and in the last day. The calls per day counter is reset to 1 with the user's first call after midnight.

Connect Time

An incrementing timer is kept. which records the user's time spent connected to the system. Time is recorded for both total usage (since the user was added to the system) and during the last day he was on. The last day recording is reset to Ø (zero) with the user's first call after midnight.

Downloaded Bytes

You may often wish to know how actively a user has been downloading files from your system. For this reason, a byte counter is maintained for downloads, for both total downloads (since the user was added to the system) and in the last day. The last day counter is reset to Ø (zero) with the user's first call after midnight.

Uploaded Bytes

Another important aspect of file transfer activity is uploading. TBBS also maintains a count of the total bytes uploaded by the user. This counter reflects all uploads made by the user since he was added to the system.

Specific Area Usage Counters

You may wish to count usage on an area-by-area basis, rather than everywhere on the system. For this, TBBS provides "billing classes." In addition to simply controlling access (which you learned about before), billing classes also provide usage accounting capabilities.

You can set billing classes on a menu-by-menu basis. Time spent in a menu or any functional entry within it will be accrued in that corresponding billing class's time counters. Time is recorded in hours and minutes used.

If you choose to implement billing classes by assigning them to menus, you can view the accrued time data via ULEDIT for each user on the system. You can also modify the data if you choose while using ULEDIT.

NOTE: With billing class Ø (the global billing class), time will not be accrued unless you use pre-authorization usage control (discussed previously). This is because global time is already being recorded as a part of each caller's userlog record. For billing classes 1 through 9, time recording will automatically be in effect if the user access a menu with billing class 1 through 9 assigned to it.

Usage Tracking in Detail

Usage tracking is the maintenance of audit trails. Audit trails provide running accounts of system activity, allowing you to "backtrace" through the things that have occurred on your system. Usage tracking may not be directly accessed often, but can be critical in locating and repairing situations of minor system failure or abuse of the system on the part of users.

Log File (Dayfile)

The primary audit trail within TBBS is called the "dayfile." The dayfile is simply a running log of system activity. Among the information it records is: user logon/logoff, file transfers, menu accesses, message entry and DOS errors that have been encountered during the running of the system.

The various user information allows you to track the activity of users on your system. You can know at a glance where they've been, and what they've done.

TBBS logs all encountered DOS errors to the dayfile as well. This allows you to track such problems as missing files, missing menus, and a variety of hardware or software related problems.

The dayfile is optional. If you choose to maintain it, you can have the output routed to a disk file, a printer connected to the system, or both simultaneously. You can also selectively enable and disable certain aspects of dayfile recording. All dayfile configuration is accomplished through CEDIT.

Real-Time Conference Recording

If you implement real-time, user-to-user conferencing (chat areas) on your system, then you have another usage tracking option – that of recording the conversations within the chat areas.

If you choose to record conference activity, a separate audit trail will be maintained for each conference you record. (You may record some and not others if you wish.)

As conferences are defined using CEDIT, numbers are given to each. These numbers are used in forming the name of the audit trail which will be recorded on disk as a text file. Conference number 1, for example, will be recorded in CONFØ1.REC. Conference 2 will be recorded in CONFØ2.REC, and so on.

Private conversations and user pages will not be recorded as part of the conference audit trail.

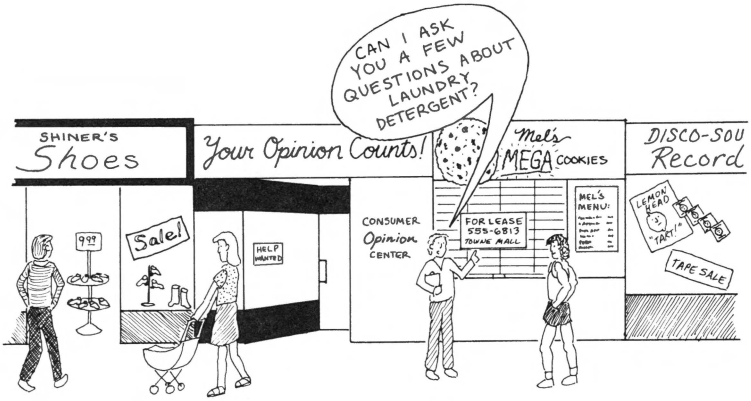
Last 10 Call Recording

TBBS automatically maintains a list of the last ten (10) calls each user has made to the system. For each call, TBBS records the date and time that the call began. This list can be useful in determining the level of activity for any caller on the system.

This information is stored as part of each caller's userlog record, and hence can be viewed with the userlog editor – ULEDIT.

An Introduction to Questionnaires

Bulletin boards are often used to gather information from your callers, and TBBS provides an easy yet remarkably powerful way of doing just that. Whether you want to simply query new callers to your system, implement online product ordering, or allow your users to cast a ballot on the issues of the day, TBBS' questionnaires are the place to go.



Questionnaires provide two basic types of user queries. Survey functions allow you to ask questions and store the responses to disk. Voting functions allow you to have ballot questions, where user responses to multiple choice questions are tallied.

If you wish, you can intermix surveys and voting into a single questionnaire. Questionnaires have no limit on the number of questions or the total size of the questionnaire itself.

TBBS provides a special language for constructing questionnaires. Don't let the term "language" scare you – even non-programmers can successfully create working questionnaires with just a little effort.

Questionnaires are used to gather information from your callers, and provide voting functions. You write questionnaires with a special, easy-to-use language. Now let's look at questionnaires in more detail.

Questionnaires in Detail

There are two basic questionnaire functions – survey and voting. You are free to use them separately, or together within a single questionnaire.

Survey style questionnaires are those where you wish to gather (and usually save) data entered by the user. A survey is used for new user questionnaires, online order taking, user profiles – any type of gathering activity which demands free-form user input.

TBBS provides a great deal of power with surveys. Users can enter strings (free-form text), numbers, yes/no type responses, credit card numbers, dates, and social security numbers. Credit card numbers are automatically format-checked and check-digit verified for accuracy and matching against accepted credit card issuers. Social security numbers are format-checked. The information gathered can be selectively saved to disk, or discarded.

Voting type questionnaires are those where you want to gather cumulative information from your user base. By providing questions with set multiple-choice responses, your users can select their preferred response to each query. The responses are saved and accumulated automatically, and may be viewed online using a special menu entry or offline with a special utility. Ballots can be secret if you wish (where even you as the System Operator won't know how a particular user voted).

As we've mentioned previously, both types of questionnaires can be combined freely. For example, you can easily construct one that has a few free-form questions which are saved to disk, followed by multiple choice voting questions. One possible application would be online product ordering which includes voting functions on the quality of service or product selection you offer.

How Questionnaires Are Constructed

Questionnaires are implemented in TBBS using the Question and Answer Language – QAL. The term "language" shouldn't alarm you. Although QAL features the power of a computer programming language, it's really very simple to learn and use. Because questionnaires can be as simple or complex as you wish, you probably won't need to learn all of the elements of QAL in order to be productive with questionnaire design and implementation.

The Question and Answer Language – QAL – provides all of the tools to construct questionnaires. It provides a set of defined elements that you assemble into a questionnaire with an ASCII text editor. This text questionnaire is processed by the QAL compiler, and the result is a binary version of the questionnaire that it used by TBBS itself.

Source text questionnaires have the file extension .QAL, and binary questionnaires have the file extension .QAF. To provide user access to a questionnaire you create a menu entry which references the .QAF file. A Type 32 menu command runs the questionnaire while a Type 33 menu command will summarize the vote totals in a questionnaire.

QAL includes a variety of ways to control the flow of a questionnaire. For example, assume you wanted to implement online product ordering through a questionnaire. You could ask a user whether they are paying by C.O.D. or credit card. Obviously it wouldn't make sense to ask a C.O.D. customer for their credit card number. QAL allows you to change the behavior or flow of the questionnaire depending on user response or other input.

Full-screen presentations are also possible with TBBS' questionnaire functionality. You can construct questionnaires as "fill-in-the-blank" forms for users with ANSI or VT-100 terminal emulation. The same questionnaires will automatically be presented in line-oriented, prompt/response form for callers without such terminal emulation.

A complete discussion of QAL – including examples – can be found in Reference Manual Chapter 8, QAL.

The TBBS Message System

One of the most common uses of a bulletin board system is for electronic communication. TBBS provides a variety of methods for your users to send electronic messages to one another. In fact, TBBS offers the widest selection of message handling options of any bulletin board software.

With TBBS' message features, you can allow your users to communicate privately, user-to-user. You can also implement public message discussion areas. You can even provide topic based message facilities, previously unavailable outside of sophisticated conferencing systems. All three of these types of message facilities can be used on your system, or (as with all TBBS capabilities) you can select only the ones you need. If you choose, you can opt to provide no message functions at all.

TBBS features powerful message entry and retrieval tools that are among the fastest, most intuitive available. Most callers can effectively use the message facilities with little or no explanation or training. The raw speed of message retrieval is especially useful for new callers, as they can quickly read through or scan message conferences with ease.

TBBS stores all of your messages in a two-file message database system. This design allows specific messages or message groups to be located by TBBS and begin displaying to the caller in the blink of an eye.

The message system has a total capacity of 60,000 individual messages. Depending on the type of message areas you implement, you can have a virtually unlimited number of individual message areas – groups of messages that pertain to the same basic topic.

TBBS also provides an array of special features for its message system. Users can enclose a file with a message (analogous to giving a computer diskette to an associate with an attached memo). The recipient can read the message, then download the accompanying file – the message and file are in an "electronic envelope" if you will.

The paper mail analogy is also duplicated with carbon copy and mailing list features, as well as return receipts and message forwarding capabilities.

The TBBS Message System in Detail

TBBS has a robust message system with a powerful array of features which allow you to present customized electronic communication on your system. You've learned why the message features are fast and plentiful. Now, let's take a look at the TBBS Message System in detail.

Introduction

As with all of TBBS, the message handling features are accessed by the user as menu functions. As such, you can use the access control system to limit access to message features. If you wish, you can even disable message features entirely by not adding the access commands to your menu structure.

You can also implement access control with each message board, separate from access control at the menu level. This type of control allows, for example, users to read messages on a particular message board but not to post replies to them. (Message boards are introduced later.)

The TBBS message system stores messages in two separate files. One of the files (MSGHDR.BBS) contains information about each message; the other (MSG.BBS) contains the text of the messages stored on the system. This database layout allows very rapid, easy message retrieval.

Messages are stored in distinct areas, called *message boards*. All messages in all boards are stored in the two message files but all messages in one board are logically grouped together for purposes of organization and access. Message boards each have a name you supply, which sets forth the general focus of all messages on that board. Each board is defined to have one of three message handling types which we'll discuss momentarily.

Messages can be entered in two ways: directly, or as a reply. If a user wishes to post a new message or to start a new discussion, he can simply enter the message with TBBS' message editor. The editor is easy-to-use and most users can be productive with it without any special training or instruction.

Messages are retrieved through the TBBS message reader. The reader is simple and designed for quick message access. Users can reply to any messages they read (if they are so qualified), which will temporarily take them to TBBS' message editor. After saving the reply, they return to message reading.

Messages are normally retrieved in sequential, chronological order. In this manner, messages will appear in the order they were written by users. You and your users also have the option of retrieving messages in reverse order, or you can retrieve specific messages a variety of different ways.

TBBS features advanced subject chaining (sometimes called threading). A subject chain begins with a single message, entered by a user. Each reply to that original message causes the reply to be added to the chain. (Replies to the replies are also added to the chain.) Each message in the chain is an additional thought or response on the original subject, therefore the messages together form a flowing, dynamic conversation. While reading messages, you and your users can choose to follow a message chain and see all entries on that subject in thought sequence, then afterward return to reading remaining messages in sequential order, independent of the chain.

TBBS provides three message handling methods, each with their own purpose and range of features and benefits. They are: electronic mail (e-mail), standard and topic. They are referred to as *board types* since they are associated with a message board as it is defined. Message boards are defined by you in CEDIT.

Electronic Mail

One of the most common paradigms in electronic communication is that of electronic mail. There is a common misconception that all electronic messages are "electronic mail" and in a sense, the description is accurate. However, TBBS uses the original meaning of the term – private, person-to-person communication electronically.

A message board defined as electronic mail type has special message retrieval tools designed around the needs normally associated with private, person-to-person communication. These tools are different than they are for the other handling types (standard and topic).

System Operators (such as yourself) are authorized to read electronic mail messages, whether you are a party to them or not. No other users can read electronic mail messages unless they sent the message or are the recipient of the message.

Standard Messages

Another common paradigm for message handling is that of public BBS message conferences. When you define a message board as standard type, you will be allowing public conversations to occur on that board. Users who are authorized may enter, read and reply to messages posted there. You can optionally configure a standard board to allow private messages to be mixed with public ones, but the default will always be that the messages are public.

As public messages, any users authorized to access the message board will be able to read any messages posted there, regardless of who entered a message or who it's to. Private messages on a standard board work as electronic mail does – only the sender, recipient and the System Operator can read the message.

Topic Messages

Although standard type message boards allow subject chains, occasionally you need additional power in topic handling. The topic message board type provides outstanding flexibility and power in topic handling. There is also a limit of 63 message boards if only electronic mail and standard type boards are used; topic boards allow you to implement a virtually unlimited number of distinct message areas.

Topic boards allow the sort of power that previously has only been available on UNIX-style message conferencing systems. Online services such as McGraw-Hill's BIX and conference software such as Caucus and Participate use this type of message handling.

Conceptually, topic boards allow users to dynamically "join" topic areas within the board, and read messages within that topic independently from other topics.

Topic boards also allow the Sysop (or other designated users) to function as message conference moderators. These moderators can dynamically add and remove topics to and from the board, and can move messages or message groups among topics – all without reconfiguring the system.

The topic interface is hierarchical in nature, like a tree. Each topic has a root (the actual topic board) and sub-topics branching out from the root. There can be messages stored at each level of the tree. Users can join and resign sub-topics on the board at will. When a particular topic is accessed, any topics the user has joined are available to him for reading. Full subject chaining is also available within each topic.

Topic boards can be accessed one of three ways: as a topic/sub-topic hierarchy with full topic navigation (as described above); as a simulation of several standard type boards combined; or finally, one particular topic or sub-topic can be accessed in isolation like a standard board type.

Special Features

A variety of special message features are optionally available to you and your callers. Most of these special features can be selectively enabled or disabled through the setting of privilege level requirements in CEDIT.

File enclosures ("electronic envelopes") is one of the primary special features. This allows users to upload a file (such as a data file or program) to the system along with a message. Whoever can access the message can then download the file enclosed with it, allowing enclosures in public or private messages. When the host message is deleted, the file enclosed with it is also automatically deleted.

TBBS also provides return receipt capabilities, much like those of traditional paper mail. When a user desires a return receipt, he informs the system prior to saving a message. When the message is retrieved by the recipient, TBBS will automatically send a message back to the sender letting him know the message was read.

Reading of messages is normally accomplished on a board-by-board basis (or topic-by-topic within topic boards). TBBS also provides combined read capabilities, which allow a user to read messages from any area he is authorized, in a single operation. The user can select or deselect which message boards are included in the combined read. (Electronic mail type boards are not included in combined reads, except for the System Operator.) The user's list of selected boards can optionally be saved as part of the user's profile. (In topic areas, selected boards are always automatically saved.)

TBBS also provides true message forwarding. This allows you or your users to pass any one of their messages to another recipient. Message forwarding also allows the movement of a message from one message board to another if desired.

Of course, TBBS allows the deletion of messages too. Users can delete their own messages during a message reading operation. A message delete function can also be provided as a separate menu option. System Operators are able to delete any messages from the system, even if they are not a party to the message.

Finally, TBBS allows two types of multiple mailings: carbon copies, and distribution lists.

Carbon copies allow the sending of a message to a small list of users. The carbon copy list is entered as part of the message, much like the "cc:" line of a written letter. When the message is saved, a copy is delivered to each person on the cc: list. Carbon copy messages reference the original recipient – much like a paper carbon copy letter is still addressed to its original recipient – and they are specially formatted when displayed to the user.

You can also use distribution mailing lists, and send a message to every user on the list. Each message is an "original" and no reference is made to the fact that the message was sent as part of a mailing list. Distributions lists are prepared in advance as text files, which reside on the disk of the TBBS machine.

An Introduction to File Transfer and Display

Bulletin board systems are often used to provide access to access to file libraries. They are also an excellent tool for presenting a variety of text information to callers. For both of these related tasks, TBBS provides an array of presentation options that let you fine tune your system to your exact data distribution needs.

File libraries serve as central-site data storage banks. They allow users to search through file directories (which you organize in a variety of ways) and download selected files to their own system. These files can be programs, information, text – any data which can be stored on disk. Your users can also upload files to the system to be added to a file library.

Data is not limited to IBM PC compatible files – any computer data, regardless of the computer it comes from – can be stored and made available through TBBS. IBM, Macintosh and Amiga PCs; UNIX and VAX mini-computers; IBM mainframes; all of these computers and more can upload data to and download data from a TBBS installation through appropriate asynchronous communications links.

Data can be uploaded and download as single files or in batches (groups of files). TBBS supports a wide range of file transfer protocols which handle the file exchange between computer systems.

Text information display is another important element of TBBS. Information, such as bulletins, technical data – anything in the form of text – can be organized, displayed, indexed and searched via TBBS.

If you have a need to transfer or display information and data to remote sites, TBBS has the tools you need to customize the presentation to your exact requirements.

File Transfer and Display in Detail

TBBS provides several ways to display files to your callers. A variety of approaches to organizing file libraries is also available to you. Let's look at these capabilities in more detail.

File Display

TBBS' file display options allow you to show the contents of any standard text file to callers. You have a variety of ways to organize these files for access and display.

You can use simple, single-file displays, which are frequently used for help or general information displays. There are also two methods of providing multiple file groupings, which are useful for product information, system bulletins – any situation where you need to group several related text files together for access under a single menu entry.

TBBS provides a powerful keyword search text database function as well. This allows users to enter one or more keywords, for which TBBS will search a text index file. Upon locating any matches, the system will display the line of the index file on which the match occurs. You have the option of providing a four-character identifier on an index line which will allow the caller to retrieve a file of extended information for that index entry. Keyword search databases are excellent for such things as online troubleshooting guides or interactive help systems.

File Transfer

TBBS provides state-of-the-art tools for organizing files into file libraries, for easy access by callers. TBBS also provides several options for direct access to specific files by name and access to specific disk directories.

With all file transfer functions within the system, TBBS provides a variety of protocols. From the ubiquitous XMODEM and KERMIT to advanced ZMODEM-90™ there is a protocol supported by TBBS that can be used by virtually any communications software or platform.

Simple file transfers can be offered in two forms: single file or group. Single file transfers reference a specific file by name. Such files are accessed directly from a menu entry, and the user is not required to enter or know the file's name. Group transfers reference a DOS directory (list of files) and allow upload and/or download of any files from that directory under caller control. File libraries provide sophisticated methods to organize groups of files. There are several choices for how these libraries are displayed to the user, with optional file date, size and description. Separate libraries can be created, and you can group together smaller libraries for access from a single point.

With file libraries, users have several file location tools available including searches by name or date and examination of the contents of file archives. Users may download selected files either individually or in batches.

More Information

A complete discussion of TBBS' file transfer and display capabilities – including examples – can be found in Reference Manual Chapter 4.

Using TBBS in a Multi-Lingual Environment

With dozens of prominent languages in the world, you may need to service callers who don't understand TBBS' default language – English. Callers in Canada may demand French in addition to English. If your system is in Europe, French, Spanish, German or other languages may be necessary. TBBS can provide these languages and more to your callers – simultaneously. A user on line 2 could be interacting with the system in English while a user on line 7 is using Italian.

TBBS' multi-lingual capability extends to all internal text messages and external text files. These two sources of text encompass all user interface text in TBBS. Internal (non-menu) hot keys are also definable by language. This would allow, for example, a French-speaking user to respond with an O for "oui" instead of a Y for "yes" in response to a yes/no question.

TBBS' multi-lingual capabilities also include the routing of a user to independent menu structures based on the selected language. In this manner, and in combination with language-specific text messages and files, you can make available virtually independent systems based on the preferred language of the user.

NOTE: Because TBBS does not actually translate languages dynamically, areas of your system such as messages always remain in the language in which they are entered.

How Multi-Lingual Works

Multi-lingual capability is enabled simply by configuring more than one language. If you only need one language ¬ even if it's not English – then you don't need to enable multi-lingual capabilities. (We'll discuss single-language systems with languages other than English later.) TBBS can support up to 9 different languages on a single system, lettered A through I.

When a user calls the system, the very first step of the logon process is the selection of the user's preferred language. All languages defined are displayed, and the user selects their preference.

The first area of multi-lingual support involves *language modules* which each contain the entire set of TBBS internal text messages and associated hot keys. TBBS is supplied with one language module – English. If you want additional languages on your system, you will need to translate the English module into the target language of your choice. (NOTE: eSoft may provide additional language modules over time. We also anticipate that some user-supplied language modules will be posted on our Support Board. Contact eSoft through the Support Board regarding the availability of additional language modules.)

The second area of multi-lingual support involves the various text files on the system which are invoked by TBBS itself. Examples of such files are help files, banner files, new user information files and so on. It **does not** include text files you create which are referenced through a menu entry. TBBS employs a standard naming convention which allows you to translate and associate these special text files in the various languages you'll be using.

The final area of multi-lingual support is your menu structure. When multi-lingual capabilities are enabled, TBBS will use an alternate method of referencing the top level menu depending on which language the user selected. You can then control the user's navigation capabilities from that point, ensuring that they receive menus written in their selected language.

It's important to note that multi-lingual capabilities control the user interface only – how the user interacts with your system. Computer technology is not yet advanced to the point where accurate, contextually correct language translation can be done easily "on-the-fly." For this reason, TBBS does not translate language itself, and messages posted on the system as well as text files associated with menu entries will be displayed in whatever language they were written in, regardless of what language the user selected at logon.

If you want to use a single language on your system other than English, then you do not need to use TBBS' multi-lingual capabilities. Instead, you may simply translate the language module and the various text files, and you're finished. The text of your various menus is written in whatever language you prefer.

Multi-Lingual in Detail

So far, you've learned about language modules, and how multi-lingual support includes text files referenced by TBBS. You've also learned that the top level menu changes depending on which language a user selects. Now, let's take a look at how all these pieces are put together.

Single Languages Other Than English

If you only need to have one language on your system, but do not wish to use English, then your task is relatively simple. It begins with editing the standard language module supplied with TBBS to use the language of your choice for each internal text message. This process is discussed in more detail in the section "Language Modules and EDLANG" later in this section.

You will also need to make sure that external text files (such as help files and banners) are edited and translated into your desired language. Since you create the menus on your system, they will of course have text in whatever language you use when you create them.

Configuring Multi-Lingual Capability

If you want more than one language on your system, then you will begin by configuring the languages themselves. This process begins by creating a LANGUAGE.BBS file which contains a list of the languages your system will have available. Afterward, you add a /L switch to the TBBS command line, which will instruct the software to operate in multi-lingual mode and use the LANGUAGE.BBS file. Refer to Reference Manual Chapter 1, Installation, for details on creating the LANGUAGE.BBS file.

Once this configuration step is complete, any users who call the system will first be asked their preferred language before any other operations take place.

For the duration of this section, let's assume you have configured three languages, as follows:

Language	**Language File**
English	STANDARD.LNG
Deutsch (German)	GERMAN.LNG
Español (Spanish)	SPANISH.LNG

(The .LNG files could be named however you wish.)

In this example, when users connect with the system, the first action to take place is the following prompt:

```
<A> English
<B> Deutsch
<C> Español
?
```

Which one they select designates the language module which TBBS uses for all internal text and command keys for the duration of this call. It also designates the language letter, which is used by TBBS for the duration of the user's call in handling the various elements of multi-lingual capabilities.

Language Modules and EDLANG

TBBS is supplied with one language module in English, named STANDARD.LNG. This file contains all internal text messages. Examples of such messages include:

Enter your FULL Name?	
Is this correct?	During logon;
From:	
To:	In message handling.

EDLANG is the program which is used to edit this (or other) language module files. If you wish to use one or more languages in addition to English, you can copy STANDARD.LNG to another file (GERMAN.LNG, for example) then work with the new copy of the file during translation.

Message by message, you would translate the text into the new language:

From:	Von:
To:	Zu:
Subject:	Thema:
<D>elete	<S>treichen

You would also need to translate the various hot keys where applicable. In our previous example, the <D>elete command has an associated hot key (letter D) that would no longer be appropriate if the command is now named "streichen." Determining which hot keys are appropriate and ensuring they do not conflict with one another is part of the EDLANG language editing procedure you'll undertake to create language modules other than English.

External Text Files

There are a number of text files which TBBS uses (some optionally) in place of internal text messages. Help, banner, restricted data rate, and new user information files are all examples of such files. These are different from text files (such as bulletins, general information files, and so on) which you create and reference from within a menu.

External text files must also be translated in a multi-lingual environment. TBBS uses a defined naming convention for these files. All of the files have the same base name, but the file extension varies as follows:

.TXT	Default (or Language A) plain text.
.ANS	Default (or Language A) ANSI version of text file.
.TLn	Language-specific (Language B-H) text.
.ALn	Language-specific (Language B-H) ANSI version.

The "n" in .TLn and .ALn represents the language letter the user selected at logon. (B is the second defined language, C the third defined language, and so on.) NOTE: Language "A" uses the default .TXT and .ANS files.

NOTE: ANSI versions of the files are optional. You can place simple ANSI escape sequences within .TXT and .TLn files if you wish, and TBBS will automatically strip the ANSI from the files when displaying them to non-ANSI capable callers. If the .ANS or .ALn files are present, they will be shown to ANSI capable callers in lieu of the .TXT and .TLn versions, which will be shown only to non-ANSI callers. Use .ANS and .ALn files where you wish to keep ANSI and non-ANSI versions separate, or in situations where you have sophisticated ANSI screens to present (with animation, for example). See Reference Manual Chapter 2 for additional information on ANSI.

TBBS will always locate which file to present in the following sequence:

1. Multi-lingual environment **and** language other than "A" selected?
   YES, go to 2.
   NO, go to 4.

2. ANSI caller **and** .ALn file exists?
   YES, display .ALn file, EXIT.
   NO, go to 3.

3. Does .TLn file exist?
   YES, display .TLn file, EXIT.
   NO, go to 6.

4. ANSI caller **and** .ANS file exists?
   YES, display .ANS file, EXIT.
   NO, go to 5.

5. Does .TXT file exist?
   YES, display .TXT file, EXIT.
   NO, go to 6.

6. Is there an internal message to substitute?
   YES, display internal message, EXIT.
   NO, ignore missing file and do nothing.

Some of the external text files have an internal substitute message (such as help files, which will substitute "Help file not available") while others do not (such as banner files, which will simply move on and display nothing if the file is missing).

The default files (.TXT and/or .ANS) are used when only one language is defined on the system, or when language "A" is selected in multi-lingual environments. The language-specific files (.TLB through .TLI and/or .ALB through .ALI) are used when languages B through I are selected by the user.

Multi-Lingual Menus

The final area where multi-lingual capabilities have an impact is with your menu structure. Since menus are at the core of your system, it's important to have menus communicate with your users in their preferred language as well.

When you configure multi-lingual capability, TBBS automatically changes the name of the top level menu to match the selected language. Instead of using menu ØØØØ (the default), TBBS will use menu ØØLn instead, where "n" the language letter the user selected.

For example, if the user selected language B, then the top level menu TBBS will use for him is menu ØØLB. It is your responsibility to ensure that the proper top level menu exists for each defined language. You will also design the menu flow from that point onward, letting the user navigate to language-specific menus (named any way you see fit) as required.

The only exception is language “A” which uses the standard top level menu – menu ØØØØ.

Other Considerations

We should remind you again that TBBS is not capable of dynamically translating among languages. For this reason, the following areas are not directly affected by multi-lingual functions, and will be displayed in whatever language was used in their creation:

Messages Electronic mail and public message conferences will be displayed as-is.

Text You Create Text files, such as bulletins, news files, instructional files and so on which are implemented as menu entries will displayed as-is. You are responsible for ensuring that the files are written in the correct language for the menu the user has accessed.

CEDIT Text This includes message board and conference names, logon and logoff greetings, and so on.

Of course, TBBS itself and its various utilities are programmed in English, and remain that way independent of multi-lingual capabilities. You, the System Designer and Operator, are the only person affected by this however.

eSoft

eSoft, Incorporated

15200 E. Girard Ave.

Suite 2550

Aurora, CO 80014

(303) 699-6565