Centipede

C= BBS Software by BUGSOFT

For the Commodore 128 computer

OWNER'S MANUAL

By Adam Fanello

Software Registration # 3

Centipede BBS

CONTRACT FOR LICENSE OF BUGSOFT SOFTWARE PRODUCTS

IMPORTANT - READ THIS BEFORE DOING ANYTHING ELSE: This Contract for License of Software is a legal agreement between you (either an individual or a single entity) and Bugsoft for the product identified above, including software and associated documentation. By installing, copying, or otherwise using the software, you are agreeing to be bound by the terms of this Contract. If you do not agree to the terms of this Contract, then do not use the product and promptly return the complete package to your place of purchase for a refund.

1. GENERAL: Bugsoft ("LICENSOR") agrees that the following terms and conditions apply to the provision of software ("Software") and documentation ("Documentation") included in this package. Software shall also include all other software delivered to Customer by LICENSOR and shall be subject to the restrictions herein. Some Software or Documentation may be owned in whole or in part by a third party. Documentation shall normally include all user manuals and supplements which by their titles refer to or apply to the Software. The protections included in this license (including all reservations of rights and limitations of liabilities) extend to Software and Documentation provided under this Contract by both LICENSOR and any third-party provider.

   This Contract becomes effective immediately upon use of the Software and continues until terminated as provided in the TERMINATION Article. Customer agrees that a facsimile copy of this agreement shall be considered equivalent to the original for purposes of validity and enforcement of this agreement.

2. LICENSE: LICENSOR hereby grants, and Customer accepts, a single non-exclusive license to use Software and Documentation subject to all the terms and conditions of this Contract. The Software is licensed solely for Customer's internal use. No rights to sublicense or market the Software or Documentation are granted. All rights not specifically granted to Customer by this license shall remain in LICENSOR.

   The Software may be used by Customer on up to three computers, provided that only one *system* is available to outside calls. A single *system* may be composed of one computer, or two multiplexed computers. The Software is licensed as a single product. Its component parts may not be separated for use. Customer may make one copy of the Software for backup, archiving, or security, but all copyright or proprietary notices in the original must be included in such copy. Customer shall not clone, reverse assemble, or reverse compile any binary part of the Software or adopt *any* part of the Software as its own. Customer is granted the right to produce and release (for free or fee) add-ons to the Software, provided that any such add-on does not violate any other part of this Contract.

3. UPGRADES: If the Software is an upgrade from another product, you may use or transfer the Software only in conjunction with that upgraded product, unless you destroy the upgraded product.

4. CONFIDENTIALITY AND NON-DISCLOSURE: Customer acknowledges that the Software and Documentation are proprietary products of and shall remain the property of LICENSOR or its suppliers. Customer will not disclose or otherwise make available to any third party any Software, Documentation, or information contained therein, in any form, except to its employees and users for purposes limited to and specifically related to Customer's use of the Software in accordance with this Contract. Customer shall take appropriate action by instruction or signed agreements with such employees and users to satisfy Customer's obligations under this Contract. If for any reason Customer gains access to LICENSOR's manuals containing any confidential or proprietary marking, or LICENSOR's software source code to which Customer does not have a right of access under a written agreement between Customer and LICENSOR, Customer agrees to not examine, use, copy, or keep such items, but shall return them promptly to LICENSOR. Customer's obligations of confidentiality and nondisclosure shall apply to all forms of software received.

   Provisions of this Article shall survive any termination of this Contract.

5. COPYRIGHTS AND TRADEMARKS: The Software and Documentation are copyrighted and are protected by United States copyright laws and international treaty provisions. Customer will not remove any copyright notice, and agrees to prevent any unauthorized copying of the Software or Documentation. All trademarks mentioned in the Software or Documentation are trademarks of their respective companies.

6. WARRANTY: LICENSOR warrants that when it delivers the Software, the Software will conform in all

material respects to LICENSOR's published specifications when operated on and with the equipment specified in the applicable Documentation. LICENSOR reserves the right to correct manuals due to typographical or clerical error. In the event of any breach of this warranty, provided notice of the breach of is given in writing to LICENSOR within thirty (30) days after the delivery of the Software, LICENSOR will, at its option, repair or replace the Software or terminate the Contract and refund any charges paid by Customer. This warranty is given by LICENSOR and not by any of its third-party suppliers.

Neither LICENSOR nor any of its third-party suppliers warrants or guarantees the results from use of the Software.

7. **IMPLIED WARRANTIES, DISCLAIMER, INDEMNIFICATION:** EXCEPT AS OTHERWISE REQUIRED BY LAW, THE EXPRESS WARRANTY IN THE WARRANTY ARTICLE OF THIS CONTRACT IS LICENSOR'S EXCLUSIVE WARRANTY AND IS IN LIEU OF ALL IMPLIED WARRANTIES, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND THE REMEDIES STATED THEREIN ARE THE EXCLUSIVE REMEDIES FOR ANY BREACH OF WARRANTY. LICENSOR WILL NOT BE LIABLE IN ANY EVENT FOR ANY CONSEQUENTIAL, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES ARISING OUT OF OR IN CONNECTION WITH THIS CONTRACT, THE PERFORMANCE OF THE SOFTWARE OR ITS USE IN ANY AND ALL CASES. CUSTOMER AGREES TO ASSUME ALL RISK AND LIABILITY FOR DAMAGES RESULTING FROM THE FAILURE OF THE SOFTWARE. LICENSOR'S MAXIMUM LIABILITY IN CONNECTION WITH OR ARISING OUT OF THIS CONTRACT SHALL BE THE AMOUNT PAID BY THE CUSTOMER TO LICENSOR FOR THE SOFTWARE.

CUSTOMER IS RESPONSIBLE FOR THE SELECTION OF SOFTWARE TO ACHIEVE ITS INTENDED RESULTS, USE OF SOFTWARE, AND THE RESULTS OBTAINED THEREFROM. CUSTOMER AGREES TO INDEMNIFY AND HOLD LICENSOR AND ITS SUPPLIERS HARMLESS WITH RESPECT TO ALL CLAIMS BY THIRD PARTIES ARISING OUT OF CUSTOMER'S USE OF THE RESULTS OF OPERATION OF THE SOFTWARE.

8. **TERMINATION:** This Contract or the license of any Software may be terminated as follows:

   (A) by the Customer at any time; or
   (B) by either
      (i) nonpayment by Customer of any amount due under applicable invoices for the Software, which nonpayment continues for a period of ten (10) days; or
      (ii) nonperformance by Customer of any other material term or condition of this Contract or of any other contract between Customer and LICENSOR related to the Software or the equipment for which it is licensed.

Upon termination of this Contract or the license of any of the Software, Customer shall promptly return to LICENSOR all copies of the Software and Documentation involved or certify in writing to LICENSOR that all copies have been destroyed, including all copies which Customer has modified and/or merged into other computer program material.

Termination of this Agreement shall not relieve Customer of any of its obligations under any invoiced license fees.

9. **APPLICABLE LAW:** This Contract is governed by the laws of the State of California, without giving force and effect to its choice of law provisions.

Table Of Contents

1 - Introduction

1.1 - Welcome

Welcome to the world of Centipede. You now hold in your hands the most powerful, flexible, and extensible bulletin board system ever created for a Commodore 8-bit computer. This is not a one time claim; Bugsoft is committed to retaining this title through continued development.

1.2 - Heritage

Centipede traces its roots back to 1985, and the release of Color 64 BBS by Greg Phountz. Color 64 spent five years receiving updates from the author and *mods* (changes made directly into the base program code) and games from other authors. In 1990, the author of Centipede decided that he wanted to run his own BBS. Liking Color 64's flexibility and large selection of mods and games, but wanting to use the full power of the Commodore 128, he wrote Version 128; an upgrade to Color 64 v7.3x that ran the system in native C128 mode. This was followed by years of updates, mods, and games for V128 that lasted until 1995, when the lack of availability of Color 64 and requests from customers led to the announcement of Centipede. Unlike V128, which was simply an upgrade, Centipede is an entirely new system. Some low level programming from V128 have been reused, and long time Color 64/V128 users will certainly see similarities between Color 64/V128 and Centipede; a result of years of influence. Much of this similarity on the program code level was done purposely in order to facilitate usage of Color 64/V128 on-line games and mods with Centipede. One thing people accustom to Color 64/V128 programming will not find in Centipede is ten years worth of updates, mods, duplicated code, and hacks. What these people will find is a clean system with all modern features built in and facilities for easily, and cleanly, adding new features in the future.

1.3 - Using This Manual

This manual serves as both a step-by-step overview of setting up a Centipede BBS, and as a reference manual. Before you begin, you may find it useful to skim through this entire manual to familiarize yourself with it, as well as Centipede. Then, next time you have a free day, clear your desk, cuddle up with this manual, some scratch paper, a pencil and your Commodore 128, and get started!

The first thing to do is to run the first file on the first disk of your Centipede package. This program (for the C128 in 80 column mode) will display the addendum to this manual; covering all changes and additions made since this writing.

This manual is split into chapters, which are subdivided into sections, which are subdivided into subsections. For instance, chapter 4 covers configuring your BBS, section 4.1 covers the bbs-setup program, and subsection 4.1.1 covers the file locations selection part of bbs-setup. The following is a brief overview of what you will find in each chapter:

- Chapter one is your introduction to Centipede, and a few general things you will need to know. Read it completely.
- Chapter two is the planning stage of your BBS. Read all applicable sections thoroughly and take notes.

- In chapter three, you will be instructed on how to select and install the Centipede files onto your storage units.
- Chapter four will guide you through configuring Centipede to your specific hardware and preferences, as well as converting any existing BBS support files you may have from a Color 64 or V128 BBS. This chapter also serves as reference for making changes in the future.
- With chapter five, you will boot your BBS for the first time. From hereon, this manual becomes a reference guide. The remainder of chapter five explains the maintenance and configuration tools that are available from within the BBS.
- Chapter six explains the concept of networking, and gives a brief overview of two of the networks available for Centipede at the time of this writing.
- For the skilled programmers, chapter seven describes the process of converting Color 64/V128 games for use with Centipede - using the Cocoon program.
- In chapter eight, we wrap things up with information about contacts, updates, and other concluding remarks.
- Appendix A contains a complete listing of all files used by Centipede, and how to use them. This information is invaluable for customizing your BBS, as well as for programmers to make use of existing routines.
- Appendix B covers the powerful Message Code Interpreter (MCI) that is built into Centipede.
- The flexible Centipede scripting language is documented in Appendix C.
- The manual concludes with Appendix D and a listing of keyboard commands that can be used throughout Centipede.

1.4 - Terminology

There is a whole language of terminology used on bulletin board systems. Most of these terms should be clear to you from prior experience with computing, or from context. A point that can cause confusion, however, is synonymous terms. Throughout this manual and Centipede, many terms are used synonymously. I will attempt to describe these synonymous terms here.

Member, Caller, and User:

For the most part, these words refer to the same person. A member is someone who has an account on your BBS. A caller is one who calls the BBS, and a user is someone who uses the BBS.

Posting and Message:

A posting or message is simply a piece of written text. A posting generally refers to such a text that is made for public display. Message is a more generic term.

Upload/Download, U/D, and File Transfer area:

These are all terms for the same thing: an area of the BBS where members can publically exchange files.

DOS mode, DOS prompt, and File Maintenance:

This is an area of the BBS where files can be viewed, scratched, and renamed. Directories can be listed, and numerous file maintenance utilities run from a line prompt. DOS is a generic term meaning Disk Operating System.

Add-on, Games, and Doors:

An add-on is some sort of system that can be added to the base Centipede BBS. A game is a

specific type of add-on. Networks are also add-ons. On some other BBS programs, these same sort of systems are called doors.

Check marks, carets, and up-arrows:

Color 64/V128 systems use the check mark character to start many of their file names. This naming convention has largely been dropped with Centipede, but you will run across it in a few places; particularly when dealing with games converted from Color 64/V128. To type this check mark, press `SHIFT-@`. Since the check mark is not part of the ASCII character set, some editors cannot produce it. In these cases, it has been common convention to substitute a caret (^) symbol, which appears as an up-arrow on the Commodore, for the check mark. Whenever you see documentation which shows a caret or up-arrow as the first character of the file name, you will need to substitute a check mark in its place.

1.5 - Running Programs

In several places through this manual, you are instructed to run programs. For instance, to boot up the BBS, you will be told to run the program "bbs". There are several ways to do this, some of which are dependent on the hardware you using. The following steps form a general outline that will work on any system:

- Make sure your Commodore 128 is turned on, in C128 mode, 80 columns, and not running any other programs. It is often a good idea to reset the computer between programs.
- If necessary, switch to the disk location (partition, subdirectory) of the program you wish to run.
- Type in the BASIC command: `run"program name",u8`. Replace `program name` with the name of the program to run, and the `8` with the device unit number of where the program is stored. Actually, the device number defaults to 8, so the `,u8` is optional in this case.

Armed with this knowledge, insert your the first side of your first Centipede distribution disk, and run the program "run me". This program will display any updated information since the writing of this manual.

2 - Planning for Your BBS

Many of us like to rip open a new piece of software and start fudging our way through it, just to see what it does. You are free to do so with Centipede, but be prepared to delete all of that hacking and come back here for a proper planning session. You see, a BBS system is not like those other little games or utilities you are accustomed to throwing in the disk drive and typing LOAD"*",8,1 on. Centipede is a very large software package - more complex than even many operating systems. After all, when was the last time you dealt with any software package on an 8-bit system that required at least a megabyte of space? You can't just throw that much code around and expect it to work.

This chapter is meant to be read through step-by-step, while actually planning your BBS. You should have some paper and a pencil handy to take down notes on what you decide.

2.1 - What Are You Offering?

The first thing to decide, is what your BBS is all about! You probably already have a general idea. Now it needs to be fine tuned. Some key questions are:

- Why are you running this BBS? Is it a support board for a group or product? Will you be stressing a social atmosphere (messages), a point of exchange (file transfers), or a fun place to waste time (games)? Something else?
- Is there a theme? A theme gives your BBS a unique look, and provides you with a basis for design decisions. Pick a theme that fits your or your group's interests and go with it!
- How much space do you have? Don't think that you will become *the* place to exchange files or to run a zillion message networks, if you are running off an REU and a fleet of floppy disks. It won't work.

2.1.1 - Message Boards and Networks

Bulletin Board Systems earned their names by being compared to cork bulletin boards used to post messages in schools and workplaces. While BBSs have expanded to offering many other features since that time, the message boards continue as a central feature. Here, any member (with proper access) may post a new message on the board or post a reply to another message.

Message networks expand the bulletin board idea for automatic posting of messages on other BBSs. As such, any message posted on one BBS will automatically appear on every other BBS in the network. Please see section 6.1 for more information about networking.

Now, keeping your BBS theme, space available, and the networks you plan to run in mind, it is time to plan your message board structure. Centipede allows for up to fifteen separate message boards on your BBS, although it starts using hexadecimal to number the boards after the ninth one, so you may want to keep the total under nine. Each message board has the following properties:

Name:
Your board needs a purpose for existence. The name should *very* briefly state this purpose in a way that clearly separates it from your other message boards. On a small BBS, you may wish to run only one message board.

Categories:
Each board must then be separates into categories. Up to twenty-six categories may be used. Each category needs a name and a clear purpose within the confines of the purpose of the board. You may also define a minimum access level required for each category, which functions over and above the message board access level. (See *Access* below.)

Threading:
A board may be threaded or unthreaded. A threaded board will have all replies to a post displayed together. In the bulletin board analogy, it is like posting replies right under the last reply. An unthreaded board will display messages strictly in the order in which they were created. In the analogy, this would be like always posting new messages and replies in the next free spot on the board, with no relation to the original post or replies.

Access:
Access can be set either by one of the nine general access levels, or as private. For a private message board, only members whom you have given explicit access will be able to see the message board.

Size of the Board:
A message board can only hold so many postings, before it becomes full and old postings must be removed. You must decide how many postings you want to keep on each message board. Enough postings must be kept for readers to be able to refer back to a detail and for members who don't call daily to still be able to read when they do call. However, a member who goes away on vacation is probably not interested in returning to read a thousand new messages. Balance is needed. Additionally, each posting requires its own file on disk. Many disk storage units have limitations on the number of files they can hold in a directory. Don't fret over this property too much though - it can be adjusted at any time.

Network Association:
For public messaging networks, message board categories must be associated with network categories. It is generally a good idea to give each network its own message board, in order to be clear to the readers where a message came from, and where a reply will go. However, Centipede does not require this. You are free to mix local and different network categories on a single message board.

2.1.2 - File Transfer Areas

The next major area of any BBS is the file transfer area. The file transfer area (sometimes called Upload/Download or U/D area) allows for members to exchange files by *uploading* to the BBS, and *downloading* from it.

Centipede allows you to group your file transfer directories (similar to disk directories) into anywhere from one to fifteen categories, although it starts using hexadecimal to number the categories after the ninth one, so you may want to keep the total under nine. Each category has the following properties:

Name:
The category name should briefly indicate the types of directories that fall under it. On a small BBS, you may wish to run only one file transfer category. In this case, Centipede will skip any category selection and go directly to directory selections.

Directories:
Each category must be separated into directories. Up to twenty-six directories may be used, each with the following attributes:

Name:	The name of the category.
Location:	The disk location where the files are stored. Each file transfer directory must be in its own disk directory (location).
Direction:	A directory may be upload only, download only, or both.
Access Level:	General access level needed to see the directory. A more restrictive category access requirement will override this setting.
Files:	Each directory may contain up to 300 downloadable files; subject to storage unit directory limitations.

Access:
Access can be set either by one of the nine general access levels, or as private. For a private file transfer category, only members whom you have given explicit access will be able to see the category

and its directories. This will override a lower directory access level.

There are several ways of setting up file transfer area options. First, if you only have a small amount of storage space, you may opt out of using categories all together. This will leave you with only directories to deal with, and is recommended if you plan to have twenty-six or fewer directories on your system.

For larger transfer areas, a category system is necessary. I'll cover two common methods used for selecting category/directory configurations: platform/type and type/platform. You are free to mix these methods or come up with your own.

Platform/type uses the computer platform (computer type required) were applicable as the category. Program type (game, utility, etc.) is the directory. The advantage of this configuration is that a user of a particular platform need not change categories much. An example configuration is:

- Non-Executables
  - Text Files
  - Documentation
  - Hi-Res Pictures
  - Sounds & Music
- Commodore 64
  - Games
  - Utilities
  - Productivity
  - SID Music

- Commodore 128
  - Games
  - Utilities
  - Productivity
- MS-DOS/Windows
  - Games
  - Utilities
  - Productivity
- Amiga
  - Games
  - Utilities
  - Productivity

Alternatively, the type/platform method places the program or file type as the category, and the platform as directories. This is particularly useful if your support for other platforms is limited, and you would rather break down the file types more. An example configuration is:

- Text Based Files
  - Humor
  - Documentation
  - Technical
- Audio/Visual Files
  - C64 Programs
  - C128 Programs
  - Non-Commodore Programs
  - Pictures
  - Digitized Sound
  - SID Music

- Utilities
  - C64 Utilities
  - C128 Utilities
  - Non-Commodore Utilities
- Games
  - Arcade
  - Educational
  - Sports
  - Strategy
  - Non-Commodore Games

A final note on the file transfers area involves a thing called *credit*. You may choose to encourage fair usage of your file transfer area by requiring *credits* to be used when downloading. Centipede has two credit awarding systems built-in, of which you may use both, either, or neither.

The first place a member may earn credits is upon uploading of a file. For each disk block uploaded to the BBS, the caller will receive a certain proportional number of credits to use for downloading. The number of credits received for each block uploaded may be any integer value. (I.e., One-for-one, double credit, triple, etc.)

The problem with the first award system is that there is no check of upload *quality*. Someone can

upload his grocery list multiple times and receive credits. The second way of earning credit then, is when another member *downloads* a file. The person who uploaded the file is then awarded credit based on how many times the file is downloaded. The more popular a file is, the more credits the uploader will earn.

Alternatively, of course, you may choose to simply disable the requirement of credits for downloading. This will effectively disable the credit system and allow any member (or any member of a minimum access level) to download any number of files within their daily time limit.

2.1.3 - On-Line Games

There are a multitude of on-line games available for Centipede. New ones pop up all the time. Centipede builds upon one of the largest selections of on-line games for Commodore computers: Color 64. Any game made for Color 64 or Version 128 can be easily converted to Centipede. Many were converted prior to the initial release of Centipede - a few of which are on your Centipede disks. If you have a small amount of BASIC programming knowledge, you may even write your own or convert a previously unconverted Color 64/V128 game to Centipede with the help of the Cocoon conversion utility. (See chapter 7.)

Games can be added to or removed from your BBS at any time. As such, little planning is required.

2.2 - General Access Levels

General access levels define a set of abilities and limitations shared by members at each of the nine levels. In Centipede, level one is the most restrictive while level nine gives full access to the system. Beyond that general guide line, the abilities and limitations of each level is completely configurable. Following is a brief list of areas and features that are tied into access levels. Details on each will be discussed in the appropriate areas later in this manual.

All menu commands
Message boards
Message categories
Each message board reader command
Post public message
Send local e-mail
Send network e-mail
File transfer categories
File transfer directories
Unreleased file access
Exemption from credit & download limits
Daily & per call time limits
Exemption from per call limits
Days until inactivity purging
New member level
Expired member level

In addition to the access levels, each member may be given individual access to a message board or file transfer categories. Daily time limits may also be adjusted on an individual basis.

There is nearly an unlimited way of defining your access levels. It is best to first think of a general guide line of what each level can do, by defining *who* will be given each level. *One* possible guide line is as follows:

1. Crippled: These members will be able to do next to nothing. They will be able to read and reply to e-mail, but only initiate mail to the SysOp. No message board, file transfer, or game access. This is a possible starting level for an unverified new member.
2. Looker: These members will have the added benefit of being able to look at messages or files, but not add anything of their own. This is also a good place for a new member.
3. Restricted: These members will be able to post messages, upload files, and play games. Local e-mail may be allowed, but not network. They are restricted by the credit system for downloads and uploads that must be released by a Co-SysOp.

4. Open:	These members have open access to most of the general system, including all e-mail, networking, games, and file transfer areas. The credit and upload release restrictions may or may not stay in place. This is your typical member.
5. Privileged:	These members have access to all non-SysOp features with little restrictions. Place valued members here.
6. V.I.P.:	These members have special V.I.P. treatment. Guest SysOps of other systems or close friends often reside here.
7. Area-Op:	These members are low level Co-SysOps. They are usually in charge of one particular area, such as releasing uploaded files or patrolling message boards.
8. Co-SysOp:	These members are your primary Co-SysOps. They have access to most SysOp functions, possibly including Account Maintenance but excluding direct disk access (DOS prompt). The Account Maintenance utility will not allow a caller to edit their own account or give someone higher access than themselves.
9. SysOp:	This person is you, and perhaps one fully trusted partner. Level nine members can do anything. It's that simple.

2.3 - What Storage Units Are You Using?

It is now time to start looking at your physical system. In this section, we will go over how to use different storage units (disk drives, hard drives, etc.) with Centipede. I assume that you already know the basics of how to use your specific hardware, as well as the Commodore 128 computer. If this is not the case, please review the relevant manuals before beginning.

Centipede is likely to be the most hardware demanding system you will ever run across for a Commodore 8-bit computer. The bare minimum storage capacity for a stripped down system is 512 kB. A more practical minimum is 1024 kB, with half of that needing to be on high speed device(s). The following subsections detail not only which devices meet the 'high speed' requirement, but also how to use each one with Centipede.

First, we need an introduction to the Centipede concept of disk *locations*. Different hardware units utilize the terms drives, partitions, subdirectories, and several other concepts for indicated a specific file directory on disk. In Centipede, this is all generalized into the term *location*. There are four attributes to defining a location: device number, prefix, command, and secondary command.

The device number should be familiar with you. Every Commodore disk drive unit has a device number between eight and fifteen. Some units allow for higher device numbers than fifteen, but Centipede does not support this. If you have more than eight units, it is time to trade in for a higher capacity device and start saving on the electric bill! For most BBSs, these devices are on 24 hours a day, so this *can* be a real concern.

The prefix is automatically attached to the file name of any reference to a file in the location. In the original Commodore design, this prefix could either be "0:" or "1:", indicating drive 0 or drive 1 on a dual-floppy unit. Modern additions now allow for partition numbers, subdirectories, and LUs to be selected via the prefix. The prefix will always conclude with a colon.

The command is a disk channel command sent to the storage unit when the location is to be selected. On simple single directory units (such as floppy disks), this is generally "i0", which tells the device to take note of what disk it currently holds. Modern units can select partitions, subdirectories, and so on, by way of this command.

The secondary command is an optional field used when a single disk channel command is not enough. For instance, the first command may select a partition, while the second command selects the subdirectory.

2.3.1 - Floppy Drives

This is the bottom line storage unit for your Centipede system. Every system should have a least one floppy, even if solely for transferring files from floppy disk to another storage unit. While it is possible to run a slow system with only a series of floppy drives, it is by no means recommended. You will need to have at least one of the other storage units listed in the following subsections.

The 1541 and 1571 floppy disk drives are only capable of holding a single directory on a disk[1]. As such, the disk location information is rather straightforward: Prefix is always "0:" and command is "i0". A device number of eight to eleven may be physically selected by jumper, solder cut, or dip switches. Refer to your disk drive manual to learn how to change the device number.

Generally, the 1581 floppy drive works much like the earlier Commodore model floppy disk drives. The 1581 does contain a partitioning option though. Each partition has its own directory space with a capacity of holding 296 files. As such, partitioning can be useful for storing multiple message boards on a single 1581 disk. A 1581 disk split into three even sized partitions can then easily hold three message boards of 290 postings each. (Leave yourself some elbow room.) Refer to your manual and/or 1581 Test/Demo disk for information on creating partitions. In Centipede, the location information will still use the prefix of "0:", now with a command of "/partition-name" and a second command of "i0".

The 1542 and other dual-drive units work just like their single drive counterparts, but use the "1:" prefix and "i1" command for referring to drive #1.

Centipede can also be used with IEEE drive units. In order to use one of these, you will need a C128 mode IEEE interface, such as the QuickSilver IEEE interface. The location information works the same as other floppy drives.

For details on configuring CMD floppy drive units and their location information, please refer to subsection 2.3.3. The CMD FD units can be effectively treated as slightly slower, smaller capacity versions of the CMD HD. It is even possible to use a CMD FD unit as your primary storage unit, if you do not mind some short delays.

The following table lists the capacity and directory space (number of files that can be held) for several common floppy drives.

DEVICE	1541	1571	1581	CMD FD-2000
CAPACITY	164 KB	330 KB	784 KB	1590 KB
DIRECTORY SPACE	144	144	296	unlimited

2.3.2 - RAM Expansion Unit (REU)

The Commodore (and clone) RAM Expansion Unit (REU) adds memory to the Commodore 128 computer in the form of external memory. This memory is not readily available to programs however. A convenient way for a program to use an REU is as a RAM-based disk drive. This is accomplished via a special program called RAMDOS.

Centipede is fully compatible with RAMDOS, and a copy of the RAMDOS 128 v4.5 package is included on your Centipede disks. Please refer to the documentation within the RAMDOS package for information on how to use it. Use the default option of memory page 14. Any device number between 8 and 15 may be used, so long as it does not conflict with any other devices.

---

[1]Actually, the 1571 can have two directories by treating the two sides of the disk as if they were two single-sided disks. This is a rarely used but occasionally useful feature that is available for those who wish to dig up the details.

Because RAMDOS uses RAM for storage, it is extremely fast. As such, it easily meets the speed requirements for your 'high speed' device for Centipede, provided that your REU is of at least 512 KB in size. The smaller 128 KB and 256 KB REUs are insufficient.

The down side of REUs is that they do not maintain data in the event of a power loss. If you lose power to your computer (including simply turning it off), the contents of your RAM disk will be lost. Because of this limitation, *only* program files should be stored on your RAM disk. All program files should be added to and/or modified on a special floppy disk (or disks) which can be used to reload the RAM disk when needed. Additionally, any time you reset your computer the RAMDOS system will need to be reloaded. Done properly though, the contents of your RAM disk will not need to be reloaded. (Once again, refer to the documentation that is included in the RAMDOS package.)

The location information for a RAMDOS REU is the device number you chose, prefix "0:", and command "IO"; just like a floppy disk.

2.3.3 - CMD Hard Drive and RAMLink

CMD's storage units are the most advanced storage systems ever built for Commodore computers. They have a level of flexibility, compatibility, and reliability that is unequaled by any other device. It goes without saying that they make excellent choices for running Centipede.

The CMD RAMLink is a high speed RAM disk. Unlike an REU with RAMDOS, there is no driver software to load (it's in ROM), and its contents are not lost when the computer is reset or turned off (it has its own power source). The RAMLink can still lose power in a back out though, so make frequent backups and do not store important information that changes frequently on a RAMLink. You may wish to consider getting a battery backup for your RAMLink. Storage capacity ranges from one to sixteen megabytes.

The CMD HD is a high capacity hard drive system. Using a normal serial connection, the CMD HD runs at a speed that may be acceptable if you don't mind short delays. You may wish to combine your CMD HD with a Jiffy-DOS chip (also from CMD) or a RAMLink in order to achieve much higher speeds. Refer to your CMD HD manual for more information. Storage capacity ranges from 20 MB to 4 GB.

Both of these devices (as well as the CMD FD units) use the same partitioning and subdirectory structuring. Unless you have reason to do otherwise, it is recommended that you use native mode partitions, set for the highest size possible. On the RAMLink and FD units, the largest size is the entire unit. On the HD, each partition may be up to 16 MB. You can then split the partitions into subdirectories for different uses. The reason for this recommendation is that free space is common to all subdirectories, but not to partitions. Obviously you have to take the situation into account. You can run a very large Centipede system (excluding the file transfer areas) on five megabytes. You need not allocate a full 16 MB partition. If you have a large hard drive, give the Centipede system two or three times of what you think you will need (so you won't have to worry about it anymore) and then create 16 MB partitions for your file transfer areas.

Finally, if you are running both a RAMLink and a HD, create a partition on your HD the same size as your RAMLink. This will make for easy backups using the *mcopy* program that comes with both units.

There are two ways of designating file locations on CMD equipment. One is to send explicit change partition and change subdirectory commands. The other is to use the full path to a file as the prefix. There is no proven advantage of one over the other, so take your pick.

Using explicit change commands, set the prefix to "0:", which tells the CMD to use the *current* partition. The first command is then "cp *x*", where *x* is the partition number. The second command is "cd*//name*" where *name* is the subdirectory name. Double slashes are needed to tell the unit to begin searching from the root directory. If you are using nested subdirectories (not recommended), use the syntax "cd*//name1/name2...*" If you are not using partitions or not using subdirectories (which is possible with some systems), then only use the one needed command and leave the secondary command blank.

Using the full path prefix method, the prefix will include the partition number and subdirectory name. For instance, to locate the directory in partition number 12, in the subdirectory *games*, use the

prefix "12/games/:". The prefix always ends with a final slash and a colon. The command can then be set to a simple "i0", and the secondary command is blank.

A mix of these two methods is also possible. For example, you may use the *cp* command to choose a partition, and then put the subdirectory in the prefix. Refer to your CMD manual for a detailed explanation and more examples on how to select partitions and subdirectories.

2.3.4 - ICT Data Chief and Mini Chief

The ICT hard drive units were the first hard drives units designed for the Commodore 64 and 128. In essence, these units pretend to be a chain of 1541 and 1571 disk drives. The hard disk is partitioned to emulate these drives. There is also a *chain* mode used to link multiple partitions together - effectively creating larger partitions. The Mini Chief is a 1571 with a hard drive added internally. On this unit, the actual floppy drive is partition zero. The Data Chief only contains a hard drive.

Centipede supports the ICT units with a limitation: chain mode is only supported for file transfer areas. For everything else, single partitions must be used. You should format your partitions alternatively as 1541 or 1571 emulating, depending on the storage capacity needed. Chained partitions need to be all formatted as 1541 partitions.

To define a location on an ICT, use the standard prefix of "0:". The command will be "*hx*", where *x* is the partition number to select. The second command is blank. For file transfer areas, you may chain together a group of partitions by using the *hm4* command as described in your manual. As an example, "hm4 11 30" will chain partitions 11 through 30 together.

2.3.5 - Lt. Kernal Hard Drive

The powerful Lt. Kernal hard drive system is also supported by Centipede. This is a high speed device that plugs into the cartridge port of your C128. An additional C128 Adaptor Board is required to use the Lt. Kernal in C128 mode. Lt. Kernal capacities range from 20 MB to 165 MB.

You may create up to ten user defined partitions, called *logical units* (LU). Each may hold as much as 32 MB and is split into sixteen *users*. These user numbers create the effect of subdirectories. In actuality, all files in an LU are stored in the same physical directory. A system with many postings on the message boards and many games could easily have a *few thousand* files in the physical LU directory, which can create a noticeable slow down. In this case, you may wish to split the files into multiple LUs to speed up access time. Please note that as far as Centipede is concerned, different user numbers are different directories. You should treat them as such.

When defining the LUs on your Lt. Kernal, you need to give careful consideration as to the size and number of LUs you use. You may wish to give one LU to the main Centipede files, another to the message boards, and then all others for the file transfer areas. While having only a few, large, LUs provides the most flexibility in usage of free space, you are limited to only sixteen directories per LU. Give these limitations careful consideration when planning. Fortunately, you do have the Lt. Kernal utilities `autocopy` and `automove` on your side. These utilities make it easy to move directories from one LU to another, should one fill prematurely.

A location on an Lt. Kernal is defined by its LU and user numbers. A single disk channel command named *ldlu* selects both of these values. The '*l*' is the command name and always remains the letter L. '*d*' stands for device. This is the device number of your Lt. Kernal - usually 8. The second '*l*' stands for logical unit, and is the LU you wish to select. The '*u*' is the user number. User numbers must be represented in hexadecimal format. That is, user #10 is '*a*', user #11 is '*b*', etc. The entire command must be exactly four characters long and consist of lowercase letters and numbers.

In Centipede, set the location with the *ldlu* command as the command, no secondary command, and "*x*:" as the prefix, where you replace '*x*' with the LU number. Within the BBS, you normally will not be prompted for the prefix, as Centipede can figure that out on its own.

The Lt. Kernal has an optional hardware component called a multiplexer, or mux, which allows for connecting up to four computers to a single hard drive. Multiple 'muxers' may be used to connect up to sixteen computers. Centipede is capable of running a dual-line BBS on two such multiplexed computers. See section 2.4 for details.

2.4 - Multiplexed System

Centipede was designed from the very beginning to be able to run *two* lines to your BBS. Because the C128 is not a multiplexing computer by nature, two C128's are required to achieve this effect - one for each line. It *is*, however, a single BBS system; since both computers share the same files.

There is only one system available for the C128 that can connect more than one computer to a single hard disk; that is the Lt. Kernal. Due to the limited number of Centipede SysOps who will use this function, I will not include full details on the Lt. Kernal dual-line add-on system in this manual. That information can be found within the "ltk-dualline.lbr" archive. This add-on may be installed at any time, and I suggest you wait until you have your BBS running on a single computer before adding it.

The only thing you must concern yourself at this planning stage is reserving the LU 9 for cross-port communications. The dual-line system requires exclusive use of LU 9, containing at least 64 Lt. Kernal blocks. A single cylinder should suffice on all hard disks. If you are not sure, give LU 9 two cylinders. That will certainly be enough. Activate the LU but do not put a DOS image on it, or access it in any other way.

2.5 - What Modem and Interface Are You Using?

Storage units are very important, but if you want your BBS to be available for people to call in, you will need a modem and possibly an interface to connect it with.

Centipede will work with any external modem that supports the Hayes *AT* command set. Nearly every modem of 1200 bps and above supports this. If you are not using a modem that plugs directly into the C128 user port (such as the Commodore 1670 and the Aprotek Minimodem-C series) you will also need a modem interface. A simple user port modem interface can drive a modem with Centipede at up to 4800 baud. The SwiftLink cartridge can handle up to 38400 baud and the Turbo232 cartridge, 203400 baud!

When using a SwiftLink or Turbo232, you may run into conflict with another cartridge you are using. If you are using a RAMLink, simply plug the SwiftLink or Turbo232 into the pass-through port on the RAMLink. In the case of an REU or Lt. Kernal host adaptor, you will need a cartridge port expander such as the EX2+1 from CMD. Use the EX2+1 rather than the EX3, so that the heavy REU or host adaptor can lay flat, rather than in the air.

For the Lt. Kernal, you will need an extra long ribbon cable to connect the C128 Adaptor Board to the host adaptor now on the cartridge port expander. Additionally, the jumper labeled P2 on the host adaptor circuit board must be connected to the right two pins (vertically) when looking at the cartridge as you would plug it into the C128. No other jumpers are to be attached to the host adaptor.

The following table offers information on how to use several types of modems with Centipede. If your modem is not listed, start with the generic version for your baud rate and play with the settings as necessary. Also, be sure to read the manual addendum on your Centipede disk for any updates or additions to this table.

MODEM NAME	CONFIGURATION	INITIALIZATION	FIXED BAUD	MAX BAUD
Commodore 1670[A]	none	AT X1 S0=0 F1 Q0 V1 M0 E0	NO	1200
Generic 1200 (including the MiniModem C[B])	none	AT X1 S0=0 S10=30 F1 Q0 V1 M0 E0	NO	1200
Generic 2400	AT &F &D0 &C1 &S0 X1 S0=0 M0 E0 &W0	AT S7=30 S10=30	NO	2400
MiniModem C24	none	AT S7=15 S10=30 X4 M0 E0	NO	2400
PPI 14400FXMT and FXSA	AT &F0 E0 M0 T &C1 S10=30 S36=5 S109=46 &W0	AT H0 Z0	NO	38400
Generic 14.4K	AT &F0 E0 M0 T &C1 S10=30	AT H0 Z0	YES	38400
Motorola Power Class 28.8	AT &F0 E0 D0 &C1 £J1 &W0	AT H0 Z0	NO	57600
Generic 28.8K	AT &F0 E0 M0 T &C1 S10=30	AT H0 Z0	YES	57600
Generic 33.6K	AT &F0 E0 M0 T &C1 S10=30	AT H0 Z0	YES	115200
Generic 56K[C]	AT &F0 E0 M0 T &C1 S10=30	AT H0 Z0	YES	115200

A. The 'newer' 4 dip-switch version is required; set switch 3 UP, and all others DOWN.
B. The MiniModem C dip switches should be set with switches 1, 4, 5, and 6 UP, and all others DOWN.
C. 56K modems can only call out at 56K. When receiving a call, such as on your BBS and any BBS you may network with, they act as a 33.6K modem

The following is an explanation of the fields:

Configuration: Run your favorite terminal program and send this command string to your modem. It stores your modem configuration in nonvolatile memory in the modem. The text should be entered in lowercase and spaces are optional. The '0's are zeros, not the letter oh. The '£' appears as a '\' in your modem manual.

Initialization: This is your modem initialization command that you will enter into the bbs-setup program for Centipede. (See subsection 4.1.3.) The text should be entered in lowercase and spaces are optional. The '0's are zeros, not the letter oh.

Fixed Baud: This field corresponds to a modem setup parameter in the bbs-setup program. High speed modems can communicate with the local computer at a higher speed than with the remote modem. This facilitates data compression. Some of these modems do not easily change the local baud rate according to the connection speed. These should then *fix* the local baud rate at the maximum speed. When using a fixed baud rate, the modem and computer will always communicate with each other at the maximum baud rate, no matter the connection speed. Using this will result in the draw back of the local and remote displays being slightly out of sync.

Max Baud: This is the maximum DTE (modem-to-computer) baud rate to use. This setting will also be entered into the bbs-setup program. On modems that support data compression, this should be set higher than the DCE (modem-to-modem carrier) rate. Please note that a user-port connection maxs at 4800 baud, and the SwiftLink at 38400 baud. Do not attempt to use a higher rate on a modem interface that does not support it.

2.6 - Planning File Structure

The Centipede files are split among several file locations (directories). Unless otherwise stated, any of these locations may overlap. For instance, you may choose to put your system files and support files in the same location. This is fine. If you can though, it makes things clearer not to place several types of files in the same location.

In each subsection, a recommended minimum storage space is given for the location. If you have the space to spare, it is recommended that you allocate two or three times this minimum. This will give you plenty of elbow room, thus preventing you from having to worry about running out of space any time soon.

2.6.1 - Program Files

Several types of files actually fall under this category. They are:

File Name	What They Are	Approximate Blocks Needed
ovl.*	BASIC program code *overlays*.	300
mdl.*	BASIC program code *modules*.	1200
mlc.*	*Machine Language* Code.	300
mnu.*	*Menu* structure definition.	70
scr.*	Executable *scripts*.	50

In addition, there are a few other miscellanies files that will be stored in this location. In total, you need around 2000 blocks (500 KB) of space in this location, and it needs to be capable of high speed loading. Refer to appendix A.1 for a listing of Centipede program files.

If you wish to run a large number of on-line games, or plan to install many add-ons to the standard Centipede system - you will need more space. Typical games require between 20 and 100 blocks of program space. In most of these cases however, it is possible to place the new code in a different location from the default location that the Centipede main system must use. This not only allows you to make better use of available disk space, but also creates a more organized file system.

This will also be your boot-up location. As such, many people like to make this some place that is easy to get to from BASIC. (I.e., Device 8, partition 1, root directory.)

2.6.2 - Support Files

There is a large collection of files, of the SEQ and REL types, that are needed by the program code. These files serve a supportive function toward the program code. They may serve to store variables ("var.*" files), title screens, player statistics in games, or just about anything else. You will need to reserve at around 300 blocks for the support files. Refer to appendix A.2 for a listing of Centipede support files.

If you wish to run a large number of on-line games, or plan to install many add-ons to the standard Centipede system - you will need more space. Support files from games may run from only a couple of blocks, to hundreds of blocks each. As with program files, you can usually place the new support files in a different location from the default location that the Centipede main system must use.

2.6.3 - Accounts Files

This location is the primary storage location for just two files: "accounts file" and "member memory". The member memory file stores the membership list in a special format that can be easily loaded into the computer's memory and used for fast member name look-ups. It is only a few blocks long. The accounts file, however, stores almost all information about each member. This REL file will be two blocks long for *each* member account you allow for. For example, if you set your system up to allow for up to 100 members, this file will be 200 blocks long.

The reason the accounts file (along with its companion, member memory) has its own location is because it is so important! If you lose this file, everyone will have to log in again as new members. People don't like that. This location should thus be on your most *reliable* device. This file is not accessed terribly often, so speed is not a deciding factor. *NEVER* place your Accounts Files location on a RAM disk!

2.6.4 - System Files

The system files are a special form of support files that are used at a lower level of the Centipede system than typical support files. There are only a few of these, and it would be perfectly logical to store them in the same location as your other support files. Allocating 30 blocks for this is typically fine - although if you are running a large collection of menu phrases (random phrases that you may set to pop up at menu selections), you may need some more space. Reserve 32 blocks for each 100 phrases. (If you are running a small enough system that you are worried about space like this, you are probably best off just not using menu phrases.)

2.6.5 - Help Files

Help files are simple menu command lists that pop up when a caller presses the question mark key in various places of the BBS. They are named with the "hlp.*" prefix. This is only for the help files for the main Centipede system, and perhaps some add-ons. Games will store their help files in the Support Files location. A mere fifty blocks will generally be enough space for this location.

2.6.6 - Caller Log File

The caller log file stores a detailed log of what each member did during his or her call. This location may require from ten to a hundred blocks of space, depending on how far back in time you want your log go.

2.6.7 - Network Files

Centipede can run up to ten networks. Support files for all of the networks are stored in this location. The amount of space needed depends on which networks you are running. Refer to the manuals for any networks you wish to run for information on file space requirements.

2.6.8 - E-Mail Files

E-mail is stored in one file per member who has e-mail waiting. Allocating three blocks per member should be sufficient, but more is safer. A fairly fast storage unit will make a notable difference when reading e-mail.

2.6.9 - Message Boards

Each message board requires its *own* disk location. Each posting on a board uses its own file and requires an average of 2½ blocks.

For message boards, any directory space limitation is the true deciding factor for the maximum number of postings on a message board. For instance, if you where to place a message board on a 1541 disk, the 644 blocks on a disk would appear to indicate for the ability to store 250 messages. Unfortunately, the 1541 can only store 144 files, no matter the size. In this case, you should store no more than 140 messages (leave a few spaces for the message board index file and any temporary files) in the message board. Refer back to the section 2.3 for information on any limitations on the number of files on your storage unit.

2.6.10 - U/D Directories

Each file transfer directory requires its *own* disk location. File space requirements depend entirely on what sort of files are being uploaded. You must use your own discretion here.

2.7 - System Structure

Centipede has a command structure that can be called polymorphic. This means that it can take many (unlimited, actually) forms. You could build your own system structure from scratch, but you will probably find it easier to start off with one of the included structures and then customize it. The structures included on your Centipede distribution disks are briefly described below. Other structures may also be made available for download in the future. In both cases, full description and documentation are included within the archives for each individual structure.

Centipede Default:

The Centipede default structure includes menus, help files, scripts, and hotkey definitions to give you a starting point for running your BBS, or creating your own structures. It is not the prettiest or the most efficient structure; far from it. What it does do is give you nearly every menu option available under a stock Centipede system - sometimes to the point of overkill. This is the system structure that is automatically on your BBS when you first install it. It can also be found in the "centi-struct.lbr" archive.

Color 64:

The Color 64 structure includes menus, help files, scripts, and hotkey definitions to give your BBS the look and feel of a Color 64 or V128 BBS. This is a good choice for maintaining some consistency if you are upgrading an existing Color 64 or V128 BBS. The archive for this structure is called "color64str.lbr".

Image:

The Image structure includes menus, help files, scripts and hotkey definitions to create a rough estimate of the look and feel of an Image BBS. Because Image uses a command prompt instead of hotkey menus, there is no way to make a true match. This system structure gets as close as possible though. The archive for this structure is called "image-struct.lbr".

3 - Installing The Centipede Files

Enough planning, let's get started on the computer! While reading through this chapter you have the simple task of preparing your storage units and copying and dissolving files. Nothing to it!

3.1 - Preparing Your Storage Units

In section 2.3 you learned how to use your storage units with Centipede, and in section 2.6, you planned out your file structure. It is now time to put all that knowledge and planning to work! Partition your drives, format your disks, and create your subdirectories. It may sound like a lot, but this is very easy. You've already planned it all out; this is the point where you simply do it. All the information you need is in the previous sections and your storage unit manuals.

3.2 - Dissolving The Archives

The Centipede files are stored in file archives in order to sort them into logical groups on disk. This is the same purpose that subdirectories serve in file systems. Since Commodore floppy disks do not support subdirectories however, we are stuck with archives. The *library* archival format is used on your Centipede distribution disks. They are named with the suffix ".lbr". The library format was chosen because it works in C128 mode with any storage hardware, is fast, and has a friendly interface that allows for individual file selection.

You'll find the "library v1.4" program on your first disk. Use this program to create and dissolve library archives. You'll probably want to copy this program to one of your BBS's disks for quick and easy access in the future. Load the library program now and then continue...

- Switch to your Default Program Files location, and dissolve to it the following archives: "modules.lbr", "overlays.lbr", "machinecode.lbr", "misc-prg.lbr", and "scripts.lbr".
- Switch to your Default Support Files location, and dissolve to it the "support.lbr" archive.
- Switch to your System Files location, and dissolve to it the "system.lbr" archive.
- Switch to your Help Files location, and dissolve to it the "help.lbr" archive.

If you received your Centipede files on 5.25" disks, you will need to flip and swap disks to find the right libraries. The BASIC command `directory` can come in useful for this. (Refer to your favorite BASIC 7.0 Encyclopedia for details.)

Besides dissolving entire archives, you are certain to find yourself in need of an ordinary file copier at some point. There are many copiers to choose from, and one is even built into File Maintenance within Centipede. You will also find a copy program on your Centipede disk(s) called "fcopyer". This C128 mode copy program, based on the "filecopy" program from Commodore, is fast and works with any storage hardware. It can even copy REL files.

3.3 - Installing a System Structure

After dissolving all of the archives, you will have a BBS using the Centipede Default system structure. (Refer back to section 2.7 for information about system structures.) You are encouraged to choose another structure that is more to your liking, but please wait until *after* you have Centipede up and running. Part of the installation process of a structure is done from within the Centipede software, so

you need to have it running first.

At this point, what you may wish to do is locate the archive for your chosen structure, and dissolve it onto a *blank* disk or directory. After you have worked your way through chapter 5, you will be ready to install the new structure. Be sure to read and follow the structure's documentation (included in the archive) completely in order to successfully install it.

4 - Configuring Your BBS

Now that you have planned out your BBS, and copied all of your files to where you want them, you now need let the software in on all of this information! This chapter explains the main setup program where main parts of the system are configured. Also covered is a utility for converting some Color 64/V128 system files over from a BBS that you may have run in the past.

Small, specialized areas of the BBS are configured from within Centipede. The configuration utilities available on all Centipede BBSs are covered in section 5.11. Many add-on systems (such as networks) will also have their own configuration utilities.

4.1 - BBS-Setup

The *bbs-setup* program is your centralized location for configuring all general BBS parameters. Many of these settings are used throughout the system. This is the place to go once you have planned for your BBS (chapter 2) and copied the files to your primary storage units (chapter 3). You will also return to the *bbs-setup* program on occasion to make changes to system parameters.

If you already have the BBS running, shut it down. Switch your active disk location to your Programs Files location and run the program "bbs-setup". On-screen instructions will guide you on how to navigate through the setup program. The following subsections will brief you on specific sections of the setup program and specific options.

4.1.1 - File Locations

In this section, you inform Centipede on how to find each of the main file locations that you choose in section 2.6. You will be prompted for the device number file prefix, and commands for each location you select to edit. If you do not have a second command for a location, simply leave that field blank. For a review on how to specify file locations on your particular storage units, please refer back to section 2.3.

Note: Caller Log #2 Files is only relevant in multiplexed systems.

4.1.2 - Programmable Key Definitions

The programmable `ALT` keys are analogous to the programmable F-keys (function keys) on the Commodore 128. From within Centipede, simply press and hold the `ALT` key and tap a number between `1` and `8`. The full text of the corresponding key definition that you define in this section will appear as if you had typed it.

4.1.3 - Modem Configuration

Refer to section 2.5 for information about configuring your particular modem, and then set the parameters:

`Use Modem DTR Signal?`

Setting this to `YES` will sometimes make for more reliable disconnections after a call. However, you cannot set this option and use the off-hook option. If you select `NO`, your modem configuration will need to include `AT&D0`. To set your modem to use DTR, you will need to use a `&D` setting of greater than zero, such as `AT&D3`.

`Use Off-Hook Command?`

Selecting yes here will take the BBS phone line off hook when it is busy and unable to accept a call. You must select `NO` to the previous setting for this to work correctly.

`Using a SwiftLink/Turbo232?`

Select `YES` if your modem is connected to a SwiftLink or Turbo232 cartridge. Select `NO` for modems (or interfaces) connected to the user port.

`Use Fixed Local Baud Rate?`

Select `YES` to force the modem and computer to always communicate at the highest baud rate.

`Maximum Baud Rate:`

Possible values for any modem interface are `300`, `1200`, `2400`, and `4800`. With a SwiftLink or Turbo232, you may also select `9600`, `14400`, `19200`, and `38400`. With the Turbo232 only, you may select `57600`, `115200`, and `230400`. These are the local (DTE) baud rates. Your actual modem connection rate (DCE) will generally be lower.

`Modem Initialization Command:`

Any modem settings that cannot be saved as a profile in the modem's memory is included here. If all settings are saved, then this *init string* will typically be `ath0z0`.

`Learn Carrier Type`

Select this option any time you change your modem or modem interface. The setup program will learn the type of carrier detection signal your system uses.

4.1.4 - Users' Time Limits

Callers are only allowed to remain on the system for a limited amount of time each day. This gives other people a chance to call in. Base daily limits are assigned by member access level. Individual adjustments may be made via Account Maintenance (see section 5.3).

In addition to daily limits, you may also set *per-call* time limits for both the AM and PM hours, as well as the maximum calls a user may make in a single day and the amount of time a user has to wait in between separate calls.

4.1.5 - Caller Purge

Accounts may be set to expire if a user does not call frequently. This is useful not only for preventing inactive members from taking up space, but also due to the fact that after a long period of absence, many people may simply forget their password!

You can set the number of days, per access level, that a member may wait between calls without their account being deleted. If you do not want to purge accounts for any particular access level, simply use a very large expiration period; 999 days will give a caller more than two and a half years to call back.

Note: All months are counted as 31 days. This makes a year 372 days.

4.1.6 - U/D File Transfer Areas

The U/D File Transfer Areas section of bbs-setup is split into three parts. Please refer back to your planning work in subsection 2.6.10 as you progress through this subsection.

4.1.6.1 - Categories

In this first part, you set the categories for your file directories. Select a category and press `RETURN` to edit its access level and name fields. To remove a category, simply clear its name field. That category and *all categories after it* will no longer be defined.

4.1.6.2 - Directories

The directory editor will present you with dual-pane view. The right side is a list of currently defined directories. Each line in the list consists of a directory number (for reference within bbs-setup only), the category number it belongs to, and the directory's name. Use the cursor up and down keys to scroll the list. On the left is a command menu of options: add directory to the end of the list, insert a directory anywhere, delete a directory, or edit an existing directory.

When adding or inserting directories, please keep in mind that the directories are categorized. As such, only the relative order of directories in the same category is important.

Choosing to add, insert, or edit will take you to the directory definition screen. Here, you may use the standard bbs-setup navigation commands (shown on screen) to edit the fields defining the directory's category, name, access level, location, and upload/download privileges. Please note that the access level for an individual directory is in addition to the category access level. The higher of the two access levels defines if a caller has access to the individual directory.

4.1.6.3 - Options

Besides category and directory definitions, there are several parameter settings related to the file transfer area. Following are descriptions of each:

`Minimum Number Of Blocks Available To Allow Uploads:`
: This value is used to reserve disk space for other files and to prevent an upload from overrunning the disk. This number is subtracted from the actual number of blocks available on disk. If the resulting number of blocks available is less than one, an upload will not be allowed.

`Use Extended File Descriptions?`
: Extended file descriptions are in addition to the thirty-eight character Short-View descriptions that are stored in the L/D directory. Each file in the file transfer area will have its own extended description file, containing the uploader's name, the number of downloads the file has received, and a member entered description that can be as long as a public message. Each description file is typically only a block or two long, but their use will double the number of files in your disk directories.

`Multiplication Factor Of Credits Awarded When File D/Led:`
: Each time a file is downloaded, the uploader may be awarded extra credits. Set this parameter to zero to disable the awarding of additional credits for each download. As an example: If this is set to 2 and a caller downloads a file of 100 blocks, the uploader will receive 200 credits. Credits will not be awarded if the downloader *is* the uploader. *Note:* This feature *requires* the use of extended file descriptions in order to function.

`Multiplication Factor Of Credits Awarded On File U/L:`
: When a file is uploaded to the public file transfer area, the uploader may be awarded credits based on this multiplication factor. For example, if a caller uploads a file of 100 blocks and this is set to 3, the caller will receive 300 credits. Set this parameter to zero to prevent any credits from being awarded when a file is uploaded.

`Maximum Number Of Downloads User May Make Per Call:`
: This is the number of files a caller may download in a single call. Keep in mind that a single double sided disk that has been zipped will account for eight downloads on its own.

`Access Level Exempt From Credits & DLs Per Call:`
: A member of this access level or higher will always be able to download, without regard for credit balance or number of files already downloaded. Set to 1 to effectively disable the entire credit system.

`Uploaded File Auto-Release Level:`
: Any file uploaded by a member with an access level less than this setting will not be available for

someone else to download until a SysOp or Co-SysOp *releases* the file. See section 5.15 for information on the file releasing utility.

`Access Level That Can Access Unreleased Files:`
Typically you will want to set this to the access level of the SysOp(s) or Co-SysOp(s) who can release these files.

`[Default UD Category Lock-Outs]`
Selecting this option will bring up a window where you can set the U/D category lock-outs assigned to new users. Each individual user may be locked out of certain categories, regardless of access level. By making a category's access level low and then locking all but a few members out (via Account Maintenance), you can create private file areas.

4.1.7 - Message Board Areas

Use this section of bbs-setup to define your message boards. The planning done in subsections 2.1.1 and 2.6.9 will guide you through most of this. We will only cover a few points here which were not covered in the planning stage.

When defining a message board's categories, you may edit each category's name, minimum access level, and associated network category. Category access levels below the message board's minimum access level will have no effect. You may leave this field set to zero for any category you wish to be accessible to anyone with access to the message board. The network category association is dependent on your network. Refer to the documentation for your network system(s) for instructions on what network category numbers to use. Set this field to zero in order to identify a category as local only. To remove a category from a message board, simply clear its name field. That category, *and all categories after it*, will no longer be a part of the message board.

The following parameters will appear on the final screen of options for the message boards:

`Minimum Number of Message Buffer Slots (0-25):`
Buffer slots are blank spots in the message board index that are available for new postings. When no blank spots are available, the BBS must remove an old posting from a full message board before it can complete posting a new one. This can cause a significant delay when the caller saves a message. Having buffer slots thus allows for quick posting of new messages. For optimum performance, the number of slots should be set to the number of messages you expect the average member to post in a single call. You will need additional directory space in each of your message board locations to store these extra messages.

`Maximum Number Of Files On Any Message Directory:`
By setting a number here, the system will insure that the BBS does not attempt to store more files on a disk than the disk is capable of holding. If there is no chance of a message board location running out of file space, set this option to 0 to disable this procedure and speed up the logoff sequence.

`Minimum # Of Blocks Until Cycle Off Messages:`
By setting a number here, the system will start cycling messages off a message board early if the number of blocks free in any message board location falls below this value. If there is no chance of a message board location running out of disk space, set this option to 0 to disable this procedure and speed up the logoff sequence.

`Access Level to Scratch & Edit Others' Postings (3-9):`
Members of this access level will be able to scratch (delete) and edit other members' public postings. You may wish to set this slightly lower than level nine if you have an area Co-SysOp who patrols the message boards for you.

`[Default Message Base Lock-Outs]`
Selecting this option will bring up a window where you can set the message board lock-outs assigned to new users. Each individual user may be locked out of certain boards, regardless of access level. By making a message board's access level low and then locking all but a few

members out (via Account Maintenance), you can create private message boards.

4.1.8 - System Colors

Centipede makes extensive use of color in order to present an appealing display. Rather than having predefined colors in the software, Centipede allows you to define eight *system colors*. When a new color is needed by the software, the system simply cycles to the next system color.

System colors are also used by *rainbowed* text, and for chat mode. In chat mode, the first system color is used for text received from the caller, and the second system color by the SysOp.

To change the colors, enter the color number you wish to change, then press the key combination for the desired color. Colors are defined on the face of the number keys on the main keypad. Hold the CONTROL key followed by the number to produce the top color, or hold the C= key and press a number to produce the bottom color.

4.1.9 - E-Mail & Messaging Options

This section allows you to define several parameters dealing with e-mail and the message editor.

`Number Of Days To Hold E-Mail & Credits:`
: Set this option to the number of days you wish for the system to hold onto old e-mail or pending credit awards (awarded when a file uploaded by the member is downloaded). Even if a piece of mail has been read and held, it will be purged according to the original date sent. All months are counted as 31 days.

`Minimum Access Level To Send Local E-Mail:`
: Access level a user must have in order to send e-mail to another local user.

`Minimum Access Level To Send Network E-Mail:`
: Access level a user must have in order to send e-mail to someone on another node of a network.

`Absolute Max Lines Per Message:`
: Each message has a maximum length, defined by the amount of memory (RAM) that is available for storage of the message. Centipede automatically adjusts itself to changing memory availability. This setting lets you set an absolute maximum size that cannot be exceeded, even if more memory is available. Having this set higher than is necessary is wasteful (due to overhead), so once you see how many lines the BBS is generally allowing for messages according to memory needs, it is best to lower this limit to match. Please note that a 40 column caller will be allowed about twice as many lines as an 80 column caller, since their lines are smaller.

4.1.10 - Other Parameters

These options did not fit in with any of the other sections, nor are there enough related options to give them each their own individual sections. They are the miscellaneous options.

`Maximum Number Of Members On System:`
: This limit determines the how much space for membership accounts is created in the *accounts file*. This can be expanded in the future, but not reduced. You should make sure that there is enough space for all membership accounts, without making the file too large. A larger file can make some operations in the system take longer. Each membership account requires two blocks of disk space.

`New Member's Access Level:`
: This is the access level assigned to a new, unvalidated, caller. Most SysOps would set this to `1`.

`Membership Expired Access Level:`

Each user can be assigned a date on which their membership expires. On this date, their access will be dropped to this level. This is most useful for paid membership systems. If you are not using the membership expiration dates, this setting will have no effect. *Note:* An access level of `0` will remove the account entirely. Also, a member will still be subject to the inactivity purging limits of their old and new access levels, as defined in subsection 4.1.5.

`Save All Members' Applications?`

Set to `YES` to cause a new member's application to be stored in a permanent file for future reference. You may view and edit this file from Account Maintenance. No matter how you set this option, a new application will always be e-mailed to member #1 (you).

`Local Mode Usage Password:`

If your BBS computer is not located in a secure location, you may define a local usage password. If a password is defined, it will be required to initiate any local functions other than logging in. Leave this field blank if you do not wish to use a password. Please note that this system is far from foolproof. There are several ways around it, including the reset button!

`Minutes Idle At Wait4Call Screen Before Run Idle Script:`

The idle script includes system activities that you want to run on a regular basis during system inactivity. The idle script is generally used for making automatic network calls. Other BBS add-ons may find other uses. By waiting several minutes after a caller has logged off, the likelihood of this system activity preventing another member from calling is minimized. Set the number of minutes you want to allow the system to sit idly before it runs this script. The default is 10 minutes.

`Minutes Idle At Wait4Call Screen Before Run Screen Saver:`

Screen savers may be used with Centipede. This setting defines how long the system should wait at the *Wait4Call* screen before running it. Set this option to `0` to disable the screen saver. Do not enable the screen saver until you have actually installed one.

`Use Sim40c When 40 Column User Is Online?`

Sim40c will stretch your display so that the characters are double wide, thus simulating a 40 column display. Your C128 will still be using the 80c (VDC chip) display and running at full speed.

`Trap BASIC Errors And Keep BBS Running?`

When enabled, the system will attempt to recover from any errors encountered in the BASIC coding. Also, the `RUN/STOP` key will be disabled, preventing you from accidentally stopping the system. The only time you would want to disable this is to debug a piece of software. The system will then crash to BASIC and allow you to examine the state of the software at the crash. *Note:* Even when enabled, you can still stop the system and return to BASIC be pressing the `RUN/STOP`, `C=`, `SHIFT`, and `CONTROL` keys all at the same time.

`Size Of Caller Log:`

Enter the size, in disk blocks, that you would like the BBS caller log to be. This log keeps track of everything a caller does while online. A larger log can keep track of more calls, but uses more disk space. Changing the size of the log will destroy its old contents.

`Your BBS's Name:`

Simply enter the name of your BBS here. It is used in a few spots throughout your system, including being displayed at the Wait4Call screen, and in the 'From' line of an e-mail receipt. Some add-ons may also make use of it.

4.1.11 - Save

Selecting will make any necessary changes to the Centipede system files to reflect your new choices.

4.1.12 - Abort

Aborting will disregard all changes you have made.

4.1.13 - Starting Anew

If you wish to change all of your configuration settings defined by bbs-setup, or if you lose the "var.main" file which stores this information (where is your backup?), you will need to dissolve the starter "var.main" file back from your Centipede software disk. You'll find this file in the "misc-prg.lbr" archive. The bbs-setup program cannot run without a "var.main" file to read.

4.2 - Converting Color 64 or V128 System Files To Centipede

If you previously ran a Color 64 v7.3x or Version 128 BBS system, you can convert many of your system and support files from that software to be used by Centipede by using the "v128-to-centi" conversion program. You may convert membership accounts, message areas, e-mail, U/D directory listings, credits to award, and message macros.

Run the bbs-setup program first. For any area you wish to convert from Color 64/V128 to Centipede, the disk locations must match between your old and new systems. If you wish to place your files in different locations for Centipede, then move your old files to the new locations before running the v128-to-centi program.

Once the Centipede configuration is made, and the files are in the correct locations, run "v128-to-centi" and select any or all of the following conversion options...

4.2.1 - User Accounts

This option will read your old "√password file" and convert its contents into your new Centipede "accounts file". Both files must be located in your new Centipede Accounts File location. Once the conversion has been made, you may delete your old "√password file".

4.2.2 - Last Read Messages

If you were using the Multi-Message Base mod for V128, the last message each user read is stored in a separate file called "√last read file". This information will be transferred to the Centipede accounts file with this option. Note that the results of this operation only work if the order of your message boards remains the same from V128 to Centipede. Both files must be located in your new Centipede accounts file location. Once the conversion has been made, you may delete your old "√last read file".

4.2.3 - E-mail

This option will convert your old e-mail message files to the Centipede format. The old e-mail files will be automatically deleted during the conversion.

4.2.4 - U/D Directory Listings

This option will go through every U/D directory location you have defined for Centipede and convert any Color 64/V128 style directory it finds into the Centipede format. The advantages of doing this over simply regenerating the directories is that the uploaded date will remain unaltered, and Short-View file descriptions will be created. The old "√directory" files will be overwritten.

4.2.5 - Messages

Select this option and the computer will go through all of the message board locations you have defined for Centipede and convert any Color 64/V128 style messages found into the Centipede format. The old messages will be deleted as they are converted.

4.2.6 - Credits to Award

To convert any credit pending files created by the Earn v3 credit system mod, select this option. This conversion will not work if you were using an older version of the Earn system. The old "√credit *" files must be located in the new Centipede default support files location, and will be deleted after conversion.

4.2.7 - Macros

Centipede allows each member to define up to nine message macros. This option will convert member macros from most Color 64/V128 macro systems to the Centipede format. For old macro mods that only allowed for one macro per member, the converted macro will become Centipede macro number one. A macro mod that allows for more than nine macros should convert. However, the member will not be able to use the higher numbered macros.

5 - Maintaining and Modifying Your BBS

Now that you have planned out your BBS and set the main parameters, you are ready to put your system online! There is more to running a BBS than simply putting it up though, it must be maintained. In addition, there are more options than just the ones in bbs-setup. Separate configuration utilities are used for specialized areas of the BBS. The different sections of this chapter will cover these built in maintenance and configuration utilities. However, first thing's first...

5.1 - Booting Up

Turn on your computer, monitor, modem, and any storage units used by the BBS. If your computer is already turned on, it is generally a good idea to reset it before booting the BBS.

Set the currently active disk location to that of your program files, and run the program "bbs". *Note*: If you run the "reboot" program instead, no further input will be required from you during the boot-up process. All boot-up prompts will accept their defaults.

As the system boots, you will see the Centipede copyright screen. This is followed by the loading of your main parameters file, activation of the Centipede machine language, and a quick identifying scan of all of your storage units.

At this point, you will be prompted for the current date and time. Enter the numeric values for the month, day, and year, using any key press as a separator. If the date is already correct, just press `RETURN`. If the date entered is different from the last known date for the BBS, you will be asked if you want to perform a midnight reset. Answer `YES` if you are truly entering a new day. Next, you need to enter the current time. This may be done with military time, or by using the letters '`a`' or '`p`' anywhere in your input. You may also use a plus sign as the first character to signal the computer to add twelve to the hour you entered. There are a few other formats also recognized. Just use whatever you find to be easiest.

With the input complete, the computer will create any needed message indexes, draw the view panel, initialize any networks you may have installed, and load the membership list into memory. If needed, a new membership list will be created automatically. Some add-ons may add routines of their own into the boot-up process too.

At this point, you will arrive at the wait for caller (Wait4Call) screen...

5.2 - The Wait For Caller Screen

The wait for caller (Wait4Call) screen is where your computer will spend most of its time when nobody is connected to the BBS. A wealth of information and statistics is displayed on this screen. The Wait4Call screen is also your command center for SysOp activity. Pressing the '`?`' or `HELP` key will bring up a list of the built-in commands you may use. Additional hotkey commands may be created. See section 5.8 for details on adding your own Wait4Call hotkey commands. The following subsections describe each of the built-in commands.

5.2.1 - Login (F1/L)

Pressing the `F1` or '`L`' keys will let you log into the BBS, as if you had called over the phone line. Note that if you have defined a local access password in bbs-setup, this is the only Wait4Call command that can be used without that password.

5.2.2 - Instant Login (F2/I)

Press F2 or 'I' to perform an instant login. This will take you to the BBS main menu, as any member, without going through the login process. The only information that will be requested is a membership number.

5.2.3 - View Caller Log (F3/C)

The caller log details every major move made by a caller while logged into the BBS. Press F3 or the 'C' key to view the log. You have the option of viewing this log in the order of oldest entry first, most recent entry first, or to search for calls that contain certain information. Select the letter of your choice or press RETURN to repeat the selection of the last time you viewed the log.

If you have a printer attached, you will also have the option of sending the output to the printer.

5.2.4 - Terminal Mode (F4/T)

Centipede includes a very simple terminal program. It has a minimal feature set and is not meant for extensive use. On screen prompting will guide you through its usage. Press F4 or 'T' to activate.

5.2.5 - Read/Reply to E-mail (F5/R)

Pressing F5 or 'R' will let you quickly read any e-mail in a raw data format and to send e-mail (reply) without logging in. You also have the option of clearing the contents of the mailbox when you are done.

If you have a printer attached, you will have the option of sending the output to the printer.

5.2.6 - SysOp Maintenance Menu (F6/S)

The SysOp Maintenance Menu is the main command center for maintaining your BBS by editing menus, scripts, settings, and running of a large variety of utilities. When you press F6 or 'S' from the Wait4Call screen, you will be instantly logged into the BBS as member number zero, with the name SysOp, and placed into the SysOp Maintenance menu. This same menu can be accessed from online. In the default Centipede menu system, it can be accessed by a SysOp by pressing the exclamation key from the main menu. The later sections in this chapter detail the operation of some of the commands available under SysOp Maintenance. Logoff, hangup, or whatever the command is on the system structure you are using, to be returned to the Wait4Call screen. (The command is 'h'angup in the Centipede default structure.)

Note: Although you can go into any area of the BBS from the SysOp Maintenance Menu, it is recommended that you restrict yourself to maintenance functions when logged in this fashion. Some areas of the BBS will not function correctly for member number zero. For instance, you can send e-mail and post messages, but you cannot receive e-mail, and some networks will identify such messages as corrupted. Also note that membership account zero is not configurable. No record is kept of its activities or options.

5.2.7 - File Maintenance (F7/F)

Pressing F7 or 'F' takes you to file maintenance (often referred to as DOS mode) where you are

presented with a line input that accepts disk operations. Any disk command channel command may be used. Refer to your storage unit manual(s) for details on all available commands. The file maintenance utility also accepts numerous commands of its own. Press RETURN on a blank line to exit. The following table lists, and briefly describes, the most command commands available:

Command	Description
@	Read current disk status
#	Change current active device (#9 for device 9)
!	Run the file editor
*	Run the multiple file copier utility (cross device)
$	Display the disk directory (standard pattern matching accepted)
%	Update a file transfer directory
buff	Run the local buffer utility
c	Copy a file on the same device (c:copyname=originalname)
cd	Change directories on some storage units (cd:dirname)
cp	Change partitions on some storage units (cp2)
f	Display the file contents to the screen (f:filename) By appending the file name with ,p for PRG or ,r for REL, you may view BASIC and REL files.
lxxx	Change Lt. Kernal location (1812 for device 8, LU 1, user 2)
menu	Run the menu editor utility
n	Format a disk or directory (n0:diskname,id)
r	Rename a file (r:newname=oldname)
s	Scratch a file (s:filename)
uj	Reset the device
v	Validate the drive or partition
x	Run the single file copy utility (cross device)

5.2.8 - System Shut Down (F8)

Press F8 to shut down Centipede and return to BASIC. It is best to do a proper shut down via this command rather than just resetting or turning the computer off.

5.2.9 - Answer/Disconnect Call (C=)

Pressing the Commodore key from the Wait4Call screen with place the modem in answer mode. If the modem is already answering a call, pressing the Commodore key will abort the connection process.

5.3 - Account Maintenance

Account Maintenance is used to edit individual information for each member's account. Upon entering, enter the handle or number of the account that you wish to edit. The first one you should edit is your own, user #1. BBS-setup creates this account when it creates the accounts file. You will need to edit your account to fit yourself. To edit a field, press the field letter and then enter the new data. Following is a brief description of each field:

A) Handle:	The member's online name. Limited to 20 characters.
B) Real Name:	The member's real name. Centipede only uses this to display the information on the view panel. Add-ons may make further use of it.
C) Access Level:	The member's general access level from 1 to 9. Set to 0 to delete the member.
D) Password:	The member's password. Limited to 9 characters.
E) Expiration:	Day that the member's membership expires. Use standard MM/DD/YY format, or anything else to disable expiration.
F) Upload Blks:	Disk blocks member has uploaded. Used for calculating credits.
G) Download Blks:	Disk blocks member has downloaded. Used for calculating credits.
H) Xfer Protocol #:	Member's last selected file transfer protocol number, between 1 and 9.
I) ShortView Flag:	Member's choice as to whether or not to be shown the Short View descriptions in U/D directories.
J) Experience Lvl:	Member's self-designated experience level with the BBS.
K) Word Wrap:	Last state of the member's message editor word wrap selection.
L) Review Posts:	Member's choice as to whether to see their own last postings made when reading new messages.
M) Default Editor:	Member's selected editor type choice: Line or Full screen.
N) Autopause line:	Number of lines on member's screen - thus where to auto-pause. Set to 0 to disable.
O) Last Msg Read:	Message number member last read on each message board. Set value to a negative to lock the member out of a message board.
P) UD Cats Locked:	File transfer area categories you want to lock member out of, regardless of access level. A number shown indicates that the member is locked from that category.
Q) Last Called:	Day, time, baud rate, and emulation mode of member's last call. Only the date can be edited.
R) # Calls Today:	Number of times member called on last called date.
S) Time Left:	Minutes member has left for the day.
T) Daily Time Adj:	Number of minutes by which you want to adjust this member's daily time limit in relation to the normal limit for their access level. Enter a negative value to subtract time.
U) Misc Comments:	Any comments you want to make about the member. It is shown to you in Account Maintenance and on the view panel. You may also select to show it to the caller using MCI commands, and a skilled member may be able to view their own comments using those same MCI commands.
V) Edit Extra Info Fields:	Their are nine Extra Info Fields that can hold any data. See the explanation for the "application" file in appendix A.2.
Z) Edit 'user # info':	View and edit the information file about this member.

To delete an account, set the access level to 0.

On exiting Account Maintenance, you will be asked if you wish to create a new member list. Do so if you have added or removed a member, or changed a handle.

5.4 - Accounts Adjustments

This utility is used for making mass adjustments to all membership accounts. This is sometimes necessary after changing a system configuration. The following are the changes you may make, and the probable reason why:

Change general access levels, if you changed the access level system.
Change selected protocol number, if you reordered the protocols or inserted one.
Change U/D category locks, if you added or removed a category.
Rearrange message board last message read numbers, if you added or removed a board.

5.5 - Editor

A single standard text message editor is used throughout Centipede. From SysOp Maintenance, you may use it to edit any text based file on any of your storage units. Help on using the editor is available online.

5.6 - Selectable Picture Format (SPF) Files

The Selectable Picture Format (SPF) defines a hybrid text file format that combines various versions of displayable text (such as a picture) into a single file. Files of this format may be used throughout the system to show different variations of a menu or picture to callers in different emulation modes. The different modes are: 40c ASCII, 80c ASCII, 40c Commodore, 80c Commodore, ANSI, RIPScript, SupeRes, and a few undefined modes that may be used in future versions of Centipede. You may create separate pictures for each emulation mode or, more likely, use a common picture for several modes. For instance, many SPF files contain only two pictures, one for all 40 column modes and another for all 80 column modes. Unless you have some special pictures, you will want to show the 40c Commodore picture to SupeRes callers, and the ANSI picture to RIPScript callers.

When you enter the SPF editor utility, you will be prompted for a file to edit. Enter the file location information and name. If the file does not exist, you will have the option of creating it.

The main SPF editor screen displays an enumerated list of emulation modes and which picture is used for each mode. There is also a list of picture letters that have been defined. Each picture has a letter of the alphabet assigned to it as a label. Generally, the label you choose for each picture is not significant. We'll cover an area where the letter is significant at the end of this section.

Pressing a picture label letter will present you with the options to view, edit, or delete the picture. To add a new picture, select a letter that is not yet defined and edit (create) it. Use the delete option to remove pictures that are no longer used.

Pressing an emulation mode number will allow you to choose or change the picture (by label) that is shown to a caller using that mode.

Note on MCI And SPF:

The SPF uses advanced MCI features (see appendix B) to perform its magic. You may use MCI in your pictures also, but avoid the use of markers (£M command) with uppercase markers. These could conflict with the picture labels used in SPF.

Optimization:

There are three significant facts about the implementation of SPF that are useful to optimize your SPF files for speed:

- Pictures are stored in the file alphabetically by label.

- The system must skip through earlier pictures in the file to display later ones.
- Larger pictures take more time to skip.

Given these facts, there are two opposing techniques for making SPF files work faster:

- Use earlier letters in the alphabet for small pictures, and later letters for large pictures.
- Use earlier letters for the most frequently accessed pictures.

Typically, a mix of these two techniques may be used.

5.7 - Menu Editor

The Menu Editor is where you can truly redefine your BBS. By changing the menu structure, you can make your BBS look like an entirely different system. Of course, the Menu Editor utility can also be used for simple things like adding new games or add-ons to your system.

To edit or create a new menu, select Menu Editor from SysOp Maintenance. If you have managed to lose your SysOp Maintenance menu, you may also get to the Menu Editor by typing `menu` at the File Maintenance prompt.

The first thing you will need to do is to select either to create a new menu, or to load an existing one. In either case, you will be prompted for a name of the menu. Menu names may be up to twelve characters. All menus will automatically be prefixed with "mnu.". The name is for reference and file naming only and will not be displayed to callers.

There are only two required, predefine menus. The main menu, "mnu.main", is the first menu a caller will be sent to after logging in. The SysOp maintenance menu, "mnu.sysop", is accessed when you select SysOp Maintenance from the Wait4Call screen. You should maintain a hotkey command to switch between these two menus, so that you can access SysOp functions from online.

The following subsections detail each of the areas in Menu Editor.

5.7.1 - Edit Menu Prompt

The menu prompt is displayed to the caller when the system is waiting for a menu command. The prompt displayed will echo exactly what you type in the editor, including cursor movements. Your prompt should tell the caller to press the '?' key for a menu listing or for help. Other possible things to include are where pressing `RETURN` will take the caller (see subsection 5.7.6) and the amount of time the caller has remaining for the call. MCI commands can be very useful in this area. Refer to appendix B for a listing. Two very useful commands are £vd to display either `RETURN` or `ENTER`, depending on the emulation mode, and £v1 to display the number of minutes remaining for the call.

Pay attention to where you leave the cursor after drawing the prompt, since this is where the caller's cursor will be left waiting for input. Press `CONTROL-P` when done.

5.7.2 - Edit Box Prompt

Frequently when a caller is asked a question such as "`Are you sure?`", a special character input prompt is displayed after the question. A simple box is common, but you can be more elaborate. This editor works the same as the menu prompt editor. The box you define will be used by any system area or command that is access off of the menu being defined.

5.7.3 - Set Phrases Flag

The menu phrases are one of those useless yet fun additions that have evolved into play on BBSs. By setting this flag to ON, a single line (or possibly two lines for 40c callers) of text will be chosen at random from your "phrases" file and displayed before the menu prompt.

To create your phrases file, place your menu phrases in a simple text file with exactly one phrase per line. Lines may be up to 78 characters in length. Then run the Menu Phrases Copy utility from SysOp Maintenance to copy your phrases into the "phrases" REL file used for quick random access by Centipede.

5.7.4 - Set Display Command Name Flag

Every menu command has a command name assigned to it. This name is entered into your caller log. If you set this flag to ON, the name will also be displayed on the screen when the caller selects the command. If the flag is OFF, then the hotkey for the command is displayed instead.

5.7.5 - Edit Normal Menu Commands

This is where you assign the menu hotkeys to run programs, scripts, and other menus. Press a hotkey command letter, number, or symbol to edit or create the command associated with it. A listing of command fields will appear with the current settings. Choose a field number to edit it. You may also select to remove a command from the menu, or cancel any changes that have been made. The hotkey command fields are:

1) Hotkey:	Select to change the hotkey that this command is associated with.
2) Access Level:	Minimum access level required to use this command. Set to 10 to block all access.
3) Command Name:	This is the name put in your caller log, and displayed to the caller if you have the *Display Command Name* flag set to ON. It is a good idea to keep this short.
4) Program Filename:	The name of the file to load for this command. Allowed files are overlays ("ovl.*"), modules ("mdl.*"), menus ("mnu.*"), scripts ("scr.*"), and variable files ("var.*").
5) Subroutine Number:	Enter the subroutine number to be run in an overlay or module. The value to use should be clear from the module description or add-on documentation.
6) Killer:	The killer will remove any new program variables created in the usage of the command. Normally you would want to remove all such variables, which is the default. You may also choose to remove only arrays or not to use any killer. You should be told what to use for this rather technical setting for any add-on you are given. If you are not sure, accept the default of ALL.
7) Graphics Requirements:	Assigns the minimum graphics (emulation mode) required to use the command. This may be set to Commodore Graphics (C/G) only, ANSI or C/G, or Any, which would include ASCII.
8) Prg Files Location:	Disk location where the program file for this command is located. All standard system modules should be left in the default location defined in bbs-setup. Many games and add-ons may be stored in a different location.
9) Support Files Location:	Disk location where the command's support files are located. Once again, standard system commands should have their support files left in

the default location.

`L) Multiplex Port Lock:` If you are running a dual-line BBS on a multiplexed HD system (such as a dual port Lt. Kernal), turning this ON will prevent users from entering this same command area on both ports.

See appendix A.1 and A.4 for a listing and description of programs and scripts that are included with Centipede.

5.7.6 - Edit RETURN Command

Use this section to define the behavior of the RETURN key at this menu. You may define it to do nothing (just redisplay the prompt), or to act as a hotkey for a command. The command definition works just as in the previous subsection.

A typical definition for the RETURN key is to have it return to the previous menu. For example, pressing RETURN from a games menu may return the caller to the main menu.

5.7.7 - File Name for Menu Help

Each menu has a help file associated with it. This help file is displayed to the caller when the question mark key is pressed from the menu, and is stored in your help files location. The default name is the same as that of the menu, with the "mnu." prefix replaced with "hlp.". Using this command, you may change the name of the help file to anything you wish, including file names with imbedded MCI commands to substitute a variable.

5.7.8 - Edit Help File

This option will place you into the standard Centipede message editor with the help file for the current menu. This can only be used for a simple, single file and single picture menu. If you are using MCI for a variable in the file name, or a SPF file, this editor will not work for your needs.

5.7.9 - Auto-Generate New Help File

It is recommended that you create your own attractive looking help files. In a pinch though, the system can generate a very simple menu listing for you using the command names.

5.8 - Script Editor

One of the abilities that makes Centipede so flexible is its scripting language. Scripting allows for easy writing of simple functions, and for quickly adding pre-written add-ons to the system.

Upon entering the Script Editor utility, you are presented with a short menu of options:

`Create New Script:` Use this to create a new script. You will be asked for a disk location and file name where the script is to be stored. Please note that by default, scripts are stored in your Default Programs file location.

`Load a Script:` Use this to load an old script from disk.

`Save the Script:` This saves the current script in memory back to its original location.

`Edit Script:` Selecting this option will uncompile the script in memory and place you into the Centipede text editor to edit the script. Saving the message will recompile the script. Any syntactical errors will be reported to you.

Please refer to appendix C for an explanation of the scripting language.

Tutorial:

Should your BBS crash due to errant BASIC coding, Centipede will trap the error, report it to the screen and your caller log, and then recover from the error. Before recovering, a *snapshot* of the screen is taken and stored in a file named "error snapshot" in your system files location. This snapshot can be of great help in tracking down the cause of a problem by showing the programmer exactly what was on the screen at the time of the crash.

We will build a script that will display the "error snapshot" file when you press the 'E' key from the Wait4Call screen. In doing so, this tutorial will demonstrate usage of the Script Editor, the scripting language, adding hotkey commands to the Wait4Call screen, and is actually a useful addition to your system! (What a bargain!)

Run the Script Editor utility and select `Create New Script`. Press `RETURN` over the location prompts to store the file in the Default Program Files location. For the file name, enter "scr.wait4call-e". All scripts must start with the prefix "scr.". Whenever a key is pressed at the Wait4Call screen, the system first checks to see if it was a built-in hotkey. If not, it then tries to run the script named "scr.wait4call-?", where '?' is the key pressed. Allowed key presses include lowercase letters, digits, and the following symbols: + - . / @ [ ] ^ ←

By naming our script "scr.wait4call-e" and storing it in the Default Programs Files location, it will be run anytime you press the letter 'e' from the Wait4Call screen.

Now we must define the commands of the script. Select `Edit Script` and enter the following commands:

```
support=8,"0:","i0"
```

This command sets the current support location. *Substitute the location above* with the location of your system files. The first number is the device number, followed by the prefix, first command, and optionally the second command. The prefix and command(s) must be enclosed in quotes. If your System Files location is the same as your Default Support Files location, then this command is not needed.

```
read "error snapshot"
```

The `read` command displays the contents of the named file, located in the current support location. MCI commands may be used as part of the file name.

```
display "Press Any Key£g1£h9£h4Wait"
```

The `display` command displays the given text to the screen. Here, MCI commands are used to wait for a key press (`£g1`), and then to backup 13 spaces over the text (`£h9£h4`).

```
run #2 in "mdl.wait4call" as "Wait4Call" with kill=none, gfx=any
```

The `run` command executes a program module or overlay. In this example, routine #2 in "mdl.wait4call" is run in order to return to the Wait4Call screen. Refer to appendix A.1 for program files and their routines.

Save the message. If you made any typing errors, the system will notify you of the problem and give you the chance to correct them. At the Script Editor menu, select `Save the Script`, and then press `RETURN` to exit the utility.

5.9 - SysOp News Editor

As a part of the login script, the system checks for any new SysOp News since the last time the member has called. Any new news is displayed.

The SysOp News Editor allows you to add a news item, delete an old news item, or view all of the current news items. Old pieces of news should be deleted when not needed in order to spare new

members from reading them, and to speed up the process of checking for new news.

5.10 - Voting Booth Editor

The voting booth may contain a single topic that your callers can vote on. Each member has one vote, which they may change at any time. A voting tally is computed after each vote is cast.

To change the voting topic, use the Voting Booth Editor. The Voting Booth Editor uses the standard Centipede message editor to allow you to enter a voting question, and up to twenty-six choices. That is all there is to it.

5.11 - Configure Area Commands

The main menus are not the only place in Centipede with hotkeys. Many of these other hotkeys may be configured from a collection of utilities available from the Config Commands menu, available off of the SysOp Maintenance menu. The following subsections detail each of these configuration utilities.

5.11.1 - Configure the Message Editor

The commands from the *edit* menu within the standard Centipede message editor can be configured with this utility. Most of these commands are straight forward, so we need not go into lengthy detail here. There are just a couple of notes about the *edit* menu hotkey commands:

- If you are logged in locally, attaching a file to e-mail and uploading a message will prompt you for a file to load from your local storage units, rather than accepting a file upload.
- The *go to line* command is only significant within full-screen edit mode.

When you change any commands here, be sure to update the "hlp.editor" file accordingly.

5.11.2 - Configure the E-Mail Reader

At the end of reading each item of e-mail, the caller will be prompted for a command. That prompt and those commands are configurable with this utility.

The prompt editor works the same as the prompt editor in Menu Editor (See section 5.7.) The command names make their functions obvious, so we need not go into lengthy detail here. After making any hotkey changes, be sure to update the "hlp.email-read" help file accordingly.

Note: Only level nine members will have access to the *Edit Sender's Account* command.

5.11.3 - Configure the Message Board Reader

At the end of reading each public message posting, the caller will be prompted for a command. That prompt and those commands are configurable with this utility.

The prompt editor works the same as the prompt editor in Menu Editor (See section 5.7.) The command names make their functions obvious, so we need not go into lengthy detail here, just a few comments.

Each hotkey has a minimum access level assigned to it. Members below level nine can only scratch or edit their own messages. Moving a message, scratching an entire thread, and changing a thread's subject should only be assigned a level so that only SysOps and Co-SysOps may access it.

When creating your prompt, please note that `REPLIES: #` will automatically be displayed prior to your prompt if the message has threaded replies following it, where # is the number of replies.

After changing any hotkeys, be sure to update the "hlp.msgs-read" file accordingly.

5.11.4 - Configure the File Transfer Commands

When flagging files and during some file transfer options, the user may choose to view a directory, select a new directory, or select a new transfer protocol. The hotkeys for these options are defined with this utility.

5.12 - Regenerate Message Board

Each message board has an index file that is used to track message numbers, categories, and threading. The file index is stored in a file named "msg index", in each Message Board Files location. Should one of these index files become lost or corrupted, use this utility to rebuild it. The regeneration process can take several minutes, so be patient.

5.13 - Regenerate/Update File Directory

This utility is used to recreate or update a file transfer directory file. Each directory location has a file named "√directory" which contains the file transfer directory information. If the file is lost, use this utility to create a new one. If you have added a file by some means other than by uploading to the BBS, or removed a file by means other than the Delete File utility (see next section), then use this utility to update the directory. The updating process may take a minute or two.

5.14 - Delete File from Transfer Area

Run this utility to remove files from your file transfers area. Up to fifty files may be selected at once from any combination of directories. Any selected files will be scratched and their directories updated upon exiting the utility.

5.15 - Releasing Files in the Transfer Area

In the bbs-setup program, you set an access level at which a member's upload is automatically released. You also set an access level at which a member can have access to unreleased files. Any file uploaded by a member below the auto-release level will appear in the directory listing to those with access in reversed type. These files must be released in order for other members to be able to download them. Using this utility, select the files to be released. The directories will be updated upon exiting the utility.

5.16 - Transfer Protocols

Centipede can accept up to ten file transfer protocols. From the transfer protocol selection module, a level nine member may also choose to press the '!' key to edit the selection of protocols available on the system.

In the protocol selection editor, a new listing will show each protocol's properties:

- Hotkey: Each protocol must have a unique hotkey, which the caller presses in order to select the protocol.
- Name: Each protocol has a name. This name is shown to the caller (along with the hotkey) when a protocol is to be chosen.
- File name: The actual protocol coding is stored in two files in your Default Program Files location. The files are both named with the "mlc." prefix, and one with the "-s" and the other with the "-r" suffix. These prefixes and suffixes are used automatically, and are not shown in the editor.

To add a new protocol, first copy the two protocol files to your Default Program Files location. Then enter the transfer protocol selection editor, select an empty slot in the list, and assign it a meaningful hotkey, name, and the name of the protocol's files.

To remove a protocol, enter the transfer protocol selection editor, select the slot with the protocol to remove, and answer YES to the confirmation prompt. You may then delete the protocol's files from your Default Program Files location.

6 - Using Add-Ons

Centipede was designed to support easily installed add-on systems. Two general types of add-ons are games and networks. Other add-ons may add any sort of added functionality you can think of.

Every add-on should contain documentation giving explicit instructions on how to add it to your system. This usually entails copying some files to your storage units, adding commands to menus, and/or modifying scripts.

It is a very good idea to keep a list of add-ons that you have on your BBS. It is also important to keep the installation archives of any add-on you are using. Keeping them handy, particulary the documentation, will help you in the event that you must rebuild a menu or script. An example of when this would be necessary is if you change your system structure. (See section 2.7.) System structures completely replace many of your menus and scripts. As such, it will be necessary to redo any modifications to these menus and scripts that you have performed in order to install an add-on.

6.1 - About Network Add-Ons

Centipede supports the use of up to ten message network systems. When installed, these networks are seamlessly inserted into the e-mail and message board systems[2]. Replies to both public and private messages are automatically routed to the correct network. A list of nodes, on all networks, is available to send new e-mail.

At the time of this writing, there are two networks available for Centipede:

- ComLink - a centralized network consisting of shared message categories and e-mail.
- Net64 - the Color 64 Network - a system of independently connected BBSs with e-mail and limited public messaging abilities.

The add-on systems for both networks are included on your Centipede disks. The following subsections describe each network.

6.1.1 - What is ComLink?

ComLink is a centralized, message-oriented network for Commodore BBSs. Entire message categories are sent to a central hub, run by Bugsoft, which distributes the messages to all nodes in the Link. E-Mail can also be sent to any user on any node in the 'Link. The entire transfer is done in only a few minutes, with a single call from your BBS to the Hub system.

In order to use the ComLink system, you must subscribe to the Hub service. The price is kept low, running about the same price as a magazine subscription. With a magazine, you receive information once a month. With ComLink, you receive and *send* information every day! It's quite a deal!

Once a night, your BBS will automatically call the ComLink Hub (located in California) at a time of your choosing. New messages are exchanged and then you are quickly disconnected. Depending on your baud rate and current activity, a call to the ComLink hub generally lasts only one to three minutes.

For information on how to subscribe to, install, or remove ComLink, please refer to the ComLink manual included on your Centipede disks within the ComLink archive.

---

[2] Net64 is not a fully integrated Centipede network. It is included largely in its original form and only for the sake of communicating with older BBSs.

6.1.2 - What is Net64

Net64 is the name I have bestowed upon the network designed for Color 64 systems. Saying "the network for Color 64" is a bit long winded, so throughout this manual and Centipede BBS, it is called Net64.

Net64 was created in 1987 by Sam Lewit as a way of exchanging messages (public and private) among the hugely popular Color 64 BBSs. Net64's interfacing into the rest of the BBS, in both the original Color 64 version and now in Centipede, is limited. Creating new messages (pubic or private) requires going into a new area specifically for posting Net64 messages. Replies to e-mail return to the sender at the proper node - but replies to public messages do not. With the newest version of Net64, supported by Centipede, SysOps may also send files to other BBSs.

There is no centralized control over Net64. BBS SysOps must apply for network connections with every other BBS they wish to communicate with. The receiving SysOp must then give the requester access and send a return application - to make the connection two way. While this 'free for all' network where each SysOp has control over whom they are connected to has its advantages, there is one very clear disadvantage; to send messages to multiple BBSs, multiple long distance phone calls must be made. Sending a message to all fifty nodes you may be connected to is not only a chore to set up, but also requires the system making fifty phone calls.

Finally, Net64 contains an optional charging system. Under this system, members must pay in advance to be able to send Net64 messages. The sending charge of a message is based upon its length.

For information on how to install or remove Net64, please refer to the Net64 manual included on your Centipede disks within the Net64 archive.

7 - Converting Color 64 and V128 Games and Add-Ons

Centipede includes a program called Cocoon[3]. This program preforms much of the labor of converting an online game from Color 64 or Version 128, into a Centipede game. It may also be used to help convert some Color 64/V128 *mods* to Centipede add-ons. Finally, Cocoon may also be used to *dot* a new program module written for Centipede. If you are a software developer, please refer to your Centipede Programmer's Manual for information about *dotting*.

7.1 - What can I convert?

Cocoon does not magically turn any program into something that can run on Centipede. It is little more than an over glorified Find and Replace routine; finding known bits of Color 64 or V128 coding, and replacing them with Centipede coding. Quite often, a small amount of programming work will be necessary to get the program running efficiently under Centipede. Cocoon is best at converting games which are in the form of a *merge*. These games would require being merged into a *skeleton* overlay on Color 64, but make for easy converting into Centipede modules. Games that contain code prior to line 9000 will need to be renumbered, and multi-overlaid games present their own unique conversion challenges.

Color 64/V128 systems also have hundreds of *mods* available. These *mods* were usually merged into on one or more of the main system overlays. Such *modding* is discouraged in Centipede, with modules and scripting being the preferred alternative. Converting a Color 64/V128 mod into a Centipede module can be a considerable effort, but Cocoon can be used to make the straightforward conversion needed as a first step.

7.2 - The Cocoon Conversion Program

Simply run the Cocoon program like any BASIC program for the C128. You will first be prompted for a disk location of the file to convert, followed by the file name to convert. Multiple files may be selected for conversion at the same time by entering a directory search string as the file name. (E.g., "$:√bbs*" to select from all files starting with "√bbs.") If you selected a single file to convert, then you will be asked for a file name of the converted file. Otherwise, the original file will be replaced. Finally, you must enter the source type. Cocoon can convert programs from Color 64, Version 128, or UnDotted Centipede.

During the conversion process, the file name and line number currently being worked on will be displayed on the screen. If the source file is a Color 64 program, there is a small chance that a warning message may pop up identifying that a variable named DS$ has been found on a certain line. This variable is *reserved* by the Commodore 128. You will need to change the variable to a new name. Pressing the ESC key during the conversion process will abort.

---

[3]The "Cocoon" name is derived from the idea that a file goes in, metamorphs, and out comes a new file. In this case, rather than converting a worm into a butterfly, a Color64/V128 game is converted into a Centipede game.

8 - Wrapping Up

We shall wrap up this manual with update policy, contact information, and special thanks to many of the fabulous people who have made Centipede possible.

8.1 - Updates, Games, and Add-Ons

Centipede is sure to have more updates in the future, and far more games and add-ons are available beyond the selection included with this package. The largest collection of all of these can be found on the author's BBS. When logging onto the Nature Reserve, at 714-828-7296, be sure to leave feedback to the SysOp reminding him that you are a Centipede owner. By doing so, you will receive access to the Centipede owners' message board and file category.

Bugsoft does not believe in charging its customers for the elimination of rogue employees (err, bugs) that may turn up in the software. Any such updates are available free of charge on the Nature Reserve BBS and several other Centipede BBSs. Other small updates will also be distributed in the same way.

Bugsoft does reserve the right to charge for large enhancements to Centipede. Availability and pricing will be determined at the time of the upgrade's release. The copy of Centipede included in this package is guaranteed to be the most recent version at the time of the original sale.

8.2 - Contacting the Author

Centipede is the result of years of BBS programming experience by Adam *'Ant'* Fanello. He can be contacted for comments, technical support, updates, suggestions, or million dollar job offers at any of the following:

Postal Mail:	Adam Fanello 4822 Larwin Ave Cypress, CA 90630-3515 USA
BBS:	Nature Reserve BBS (Net64 capable) 714-828-7296 User #1: 'Ant' User #2: 'Centipede'
ComLink:	Send e-mail to: 1-Ant -or- 1-Centipede
Internet:	AdamF@acm.org -or- ant@bugsoftware.com http://www.bugsoftware.com

8.3 - Getting Help From Other SysOps

Adam can get real bogged down with mobs of e-mail. Quite often, the best place to get support is from other Centipede SysOps. Most SysOps are very happy to help out a fellow Centipede BBS owner. You can contact these other SysOps through e-mail, but the best place to get help is in the Centipede SysOp message category of ComLink or some other network that hold a large assembly of Centipede owners.

8.4 - How to Report Problems

There is no such thing as an error-free program, with the possible exception of the infamous one line program, "Hello World." This is particularly true of huge systems like operating systems, productivity suites, and BBS programs. Every effort is made at Bugsoft to eliminate these problems, but it is a continuing battle of human vs. machine.

When you wish to report a problem or ask for technical help, there is a right way and a wrong way to state the problem. A statement such as "It crashes when reading messages!" tells anyone trying to help you almost *nothing* about the cause of your problem. If you state "I get an illegal value in line 9300 of mdl.mb-main when trying to read a message," you will get a quick response informing you that you have a corrupted posting that you need to delete.

The best strategy for reporting an error that causes a system crash is to include your "error snapshot" file. The error snapshot is a picture of what was on the screen when the crash occurred, including the error message. Other insightful information includes a description of what the system or the caller was doing leading up to a problem, a list of what add-ons you are using, and a description of your hardware and configuration. There is no such thing as too much information, only too little.

8.5 - Thanks from Adam Fanello

A project of this size could not have been accomplished by one person alone. We all owe Centipede's existence to a gaggle of people. I cannot possibly name everybody who has an effect on Centipede here, but the following people, presented in semi-random order, have stood out and have earned their thanks here.

- Thanks to all the people who purchased my previous BBS program, V128. Without the enormous success of V128, I doubt I would have had the courage, or prodding, to have created Centipede.
- Mounds of thanks to the miracle worker, John *'Iceman'* Pinson of the Inner Circle BBS. Not only was he the first to declare that I would one day be a SysOp, but he has continued to be one of my biggest supporters. John's natural skills of generating publicity and making connections could very well be how you came to learn about Centipede. His ability to destroy what little modesty I had lead me to reach out to possibilities I had hardly even imagined.
- Lots of love to my mother, who not only put up with all my computers and dreaming, but supported me every step of the way.
- Thanks to Roger *'Frog'* Hyatt, who was the first brave SysOp to allow me to tack a crack at programming on his Color 64 BBS - the forerunner of V128 and later, Centipede!
- Thanks to Richard *'Wanderer'* Cunningham (not the TV character) of the Desert Oasis BBS for his lasting support, and one really incredible code optimization!
- Thanks to Ken *'Megga Force'* Nealey of the Ultimate Force BBS, for making sure there was a selection of games already converted to Centipede in time for its release.
- Thanks to Michael *'Byte1'* Crisp of Thunder Publishing, for designing the Bugsoft and Centipede logos, as well as hosting the Bugsoft web site.
- Thanks to *'Grampa'* Charles Nichols for drawing the Commodore text/graphics version of the Bugsoft and Centipede logos.
- And of course, a big thanks to *you*, for purchasing Centipede!

A - File Listings and Descriptions

This appendix contains a listing of all of the standard files found on a stock Centipede system - excluding any add-ons.

A.1 - Program Files

A.1.1 - Modules

Program modules make up the majority of the programming code in Centipede. Multiple modules may reside in memory at once, providing that they are at different *levels*. The BASIC line number ranges reserved for each level are as follows:

Level	Line Range
1	9000-49999
2	50000-59999
3	60000-62998

These ranges are a guide-line, not a set rule. Each module's level is listed with its description. Typically, a module is level 2 or 3 because it is called by a lower level. This does not prevent a higher level module from being loaded directly (such as from a menu or script) without a lower level module being present.

All modules must be named with the "mdl." prefix.

mdl.acnt-maint	
Account Maintenance utility. (Level 1)	
1	Prompt for member account to edit
2	Account to edit is in the variable 'fr'. Used to validate from e-mail.
3	Create a new membership list and member memory files.
4	Load member memory file into memory. Calls routine #5 below first.
5	If membership list or member memory file is missing, they are created.

mdl.acnts-adjust	
Make global adjustments to all accounts' access levels, selected protocol, file area category access, and message board ordering. (Level 1)	
1	Enter the utility.

mdl.apply

New member login. (Level 1)

1	Configure new account and enter application date. Called from scr.new-login. Aborts rest of script upon an error (such as carrier loss).

mdl.bootup

Boot the BBS. (Level 1)

1	Identify devices, load var.main, var.bootup, days stats, and call scr.bootup

mdl.caller-log

Update and display the caller log. The data is stored in the support file "caller log". (Level 1)

1	View the caller log
2	Update the caller log with current call - data in the ig$() array. Called in logoff scripts
3	Copy past day's worth of log into daily backup. *(Unimplemented)*

mdl.chat-req

Request chat with the SysOp. Will play digitations or a telephone sound effect, depending on support files available. (Level 1)

1	Entry point.

mdl.copy

File copying routines. (Level 2)

1	Single file copier - with prompting.
2	Multiple file copier- with prompting.
3	Single file copier - via programming: Source device 'b', prefix 'b$', name f$ Destination device 'dv', prefix 'dr$', name fi$ File type included in file name. I.e. ",s" or ",p". REL not supported.

mdl.copy-phrases

Create and update the menu "phrases" file. (Level 1)

1	Enter the utility.

mdl.credits	
The credit system notification and credit awarding system. (Level 1)	
1	Notify member of any downloads of files that s/he has uploaded. Credits are awarded if system is configured to do so. Run as a part of the login scripts.
2	Purge old credits files that have been sitting around for too long. Called as part of the midnight reset script.

mdl.date&time	
Read, compute current date and time. (Level 1)	
1	Prompt SysOp for current date and time. Called during bootup script or from SysOp Maintenance.
2	Compute new long format date variable.
3	Read current date and time from a CMD device with a RTC. May be called during the bootup script.

mdl.deletefile	
Select files a delete them from the file transfer directories. (Level 1)	
1	Enter the utility.

mdl.dos	
File Maintenance command prompt and utilities. (Level 1)	
1	Enter the utility.

mdl.download	
Download (send) files from the BBS using a file transfer protocol. (Level 1)	
1	Select files and download them.
2	Download flagged files.
3	Download file at h=10 with local name in 'fi$', send name as 'b$'.
4	*Unused.*
5	Term program file send.

mdl.edit-caller	
Edit information on caller currently online. (Level 1)	
1	Enter the utility. Called by pressing the F5 key from a menu.

mdl.editdesc

Edit file transfer area extended and short-view file descriptions. (Level 1)

1	Select files and edit descriptions.
2	Add to (tag) an existing extended file description.
3	Prompt caller as to whether or edit the existing description or add to it. Only SysOp or uploader may edit existing.

mdl.edit-menu

Menu editor. (Level 1)

1	Edit/Create a menu. Called via SysOp Maintenance.

mdl.editor

System text editor - both line based and full screen. (Level 2)

1	No read or save routines. Text already in a$(0 - (a-1))
2	No read, do save. File name in fi$ at location h=10.
3	Read old, save new. Prompt for file name & location. Used from File Maintenance and SysOp Maintenance.
4	Read old, save new. File name in fi$ at location h=10.
5	Merge file into message in memory. 'a'=starting index, file #8 already open.
6	Read old, no save. File name in fi$ at location h=10.

mdl.editr-config

Configure message editor command hotkeys. (Level 2)

1	Enter the utility.

mdl.edit-script

Script editor. (Level 1)

1	Edit/Create a script file. Called from SysOp Maintenance.

mdl.email

Read or send e-mail. (Level 1)

1	Prompt caller: Open mailbox or send new e-mail?
2	Open mailbox.
3	Send e-mail. If 'fr', 'fr$', and 'nf$' are set, will automatically send reply e-mail to that user. Otherwise, caller will be prompted for person to e-mail.
4	Send feedback. Reads "feedback.menu". If not present, will e-mail member #1.
5	Show/Search membership list.

mdl.email-config

Configure e-mail reading command hotkeys. (Level 2)

1	Enter utility.

mdl.filescan

Scan file transfer area for a file by name, short-view description, or date uploaded. (Level 1)

1	Enter the utility.

mdl.gfxtype

Select emulation mode: ASCII, ANSI, RIP, Commodore, or SupeRes. Also select screen width of 40 or 80 columns in some modes. (Level 2)

1	Try auto-detecting ANSI and SupeRes first. Call routine #2 below if not detected.
2	Prompt caller for emulation mode and screen width.
3	Set the emulation mode to type in 'd%(20)' and screen width in 'c'.

mdl.history-log

Upload/View caller history log for current and previous days. (Level 1)

1	Add last call to the log with baud rate.
2	Add last call to the log with line number (for dual-line BBSs).
3	Midnight reset (for single-line BBSs).
4	Midnight reset (for dual-line BBSs).
5	Display logs.

mdl.login

BBS login process. (Level 2)
Not a conventional level 2 module. Uses lines 20000-49999. Can load normal level 2 modules as level 3.

1	Normal modem login. Waits fro RETURN to begin. Checks for Net64 activity and then falls through to routine #2.
2	Select emulation mode, display login msg, and fall through to routine #3. Entry point for local login.
3	Prompt for member name or number, password, or NEW. Calls new-login script for new members. Completes by calling the login or quick-login script.

mdl.loginkey

Module for handling network call logins and possibly other specialized logins. The file name ends with the PET-ASCII value of the character press (at the "PRESS RETURN" prompt) that activates the module. (Level 3)

mdl.logoff

Log a caller off the BBS. If in SysOp Maintenance mode, either entry point will simply return to the Wait4Call screen. (Level 1)

1	Play raw.logoff digitation and update member account file. Returns to Wait4Call screen.
2	Play raw.logoff digitation and update member account file. Returns to calling routine. Use this entry point if being called from a script and is not the last part of the script.

mdl.macro

Message macro editor/creator. (Level 1)

1	Enter utility.

mdl.mb-config

Configure the message read command hotkeys. (Level 2)

1	Enter the utility.

mdl.mb-extra

Miscellaneous routines used by Message Board system. (Level 2)

1	Scratch current thread. Called from mdl.mb-main.
2	Change current thread subject. Called from mdl.mb-main.
3	Move current message to another category. Called from mdl.mb-main.
4	Let member define message board auto-scan.

mdl.mb-main

Main message board system module. (Level 1)

1	Read messages menu. (One board, auto-scan boards, all new in auto-scan boards, define auto-scan.)
2	Read in current message board.
3	Read in selected auto-scan message boards.
4	Read all new in selected auto-scan message boards.
5	*Undefined.*
6	Post in current message board.
7	Post - prompt for a message board.
8	Scratch a message - prompt for message board. Non-SysOp can only scratch their own.
9	Edit a message - prompt for message board. Non-SysOps can only edit their own.

mdl.mb-purge

Purge extra messages from message boards. Usually included in the logoff scripts. (Level 1)

1	Scan all message boards and cycle off old messages if exceed maximum message per message board, or maximum files per disk directory.
2	Same as #1, but only look at message boards accessed by the last caller.

mdl.mb-select

Select current message board, access message indexes. (Level 1)

1	Prompt for a message board. (Line 60100)
2	Load the message index for the current message board. (Line 61000)
3	Save the message index for the current message board if it has changed. (Line 62000)
4	DIM message index variables in preparation for use. (Line 63000)

mdl.midnight	
Perform several midnight reset routines. (Level 1)	
1	Put daily stats in caller log, reset time limits, check for membership expiration, purge old e-mail, validate directories, and update file transfer files number of downloads stats.
2	Put daily stats in caller log. None of the others.
3	Call routine #1 above if running on line 1 of a dual-line BBS, or routine #2 above if line 2.

mdl.modem	
Specialized modem communications routines. (Level 2)	
1	Answer an incoming call. (Line 60100)
2	Disconnect/Hangup modem. (Line 61000)
3	Take modem off-hook, if configured to do so. (Line 61500)
4	Place modem on-hook. (Line 61800)
5	Send modem initialization command. (Line 62000)
6	Send modem AT command in 'b$'. (Line 63000)
7	Set baud rate to 'br'. (Line 62500)
8	Call phone number in 'f$' at baud rate 'br'. (Line 63500)

mdl.options	
User configurable account options. (Level 2)	
1	Change password.
2	Set auto-pause screen length or disable.
3	Set experience level.
4	Set Short-View display choice.
5	Set "Review Own Posts" choice.
6	Set default to Full Screen editor choice.

mdl.panel-draw	
Draw the View Panel. (Level 1)	
1	Draw the panel. Called by mdl.view-panel.

mdl.protocol

Select, configure, and load file transfer protocols. (Level 2)

1	Select a file transfer protocol and load 'receive' version of protocol into memory. A level 9 member may also edit the protocol selection.
2	Load caller's choice protocol into memory for file receiving.
3	Select a file transfer protocol and load 'send' version of protocol into memory.
4	Load caller's choice protocol into memory for file sending.

mdl.regen-dir

Regenerate, or add to a file transfer directory. (Level 2)

1	Regenerate current directory.
2	Add file 'i$' with description 'b$' to the current directory.
3	Update the current directory with file names 'f$(1-mt)' and descriptions 'b$(1-mt)'.
4	Select a directory and regenerate it. Called from SysOp Maintenance.

mdl.regen-mb

Regenerate message board indexes.(Level 1)

1	Automatically scan all message boards and regenerate any missing indexes.
2	Prompt for a message board to regenerate, and regenerate it.

mdl.releasefile

Select unreleased file transfer area files and release them. (Level 1)

1	Enter the utility.

mdl.screensaver

The screen saver from the Wait4Call screen. There are many screen savers to choose from. (Level 2)

1	Activate the screen saver. Called from mdl.wait4call.
2	Configure the screen saver.

mdl.spf-edit

Edit/Create a Selectable Picture Format file. (Level 1)

1	Enter the utility.

mdl.sys-buffer

System buffer access. (Level 2)

1	Prompt with menu to load or save the contents of the system buffer.

mdl.sysop-news

Edit or display the SysOp News. (Level 1)

1	Display any new SysOp News since the member's last call. Called in the login scripts.
2	Edit/Delete/Add new SysOp News.
3	Display all SysOp news.

mdl.term

Terminal mode program. (Level 1)

1	Enter the program. Called by pressing 'F4' or 'T' from the Wait4Call screen.

mdl.ud-config

Configure the file transfer area command hotkeys. (Level 2)

1	Enter the utility.

mdl.ud-dir

File transfer directory selection and viewing. (Level 2)

1	View current directory. (Line 54000)
2	Full screen interactive directory viewer. (Line 52000)
3	Choose download directory. (Line 56000)
4	Choose download directory and display it. (Line 50400)
5	Choose download directory and full-screen interactive view. (Line 50500)
6	Choose upload directory. (Line 56000)
7	Choose upload directory and display it. (Line 50400)
8	Choose category only. (Line 56600)
9	Select (flag) files. (Line 57000)
10	List flagged files. (Line 58000)

mdl.upload

Upload (receive) files to the BBS using a file transfer protocol. (Level 1)

1	Upload file(s) - prompt for directory and file names on single file protocols.
2	*Undefined.*
3	Upload single file to location h=10 with the file name in 'fi$'. Returns 'fi$=""' if transfer fails.
4	Term program file receive.

mdl.user-acnt

Read from or write to the accounts file. (Level 2)

1	Read all normal info for current member. (Line 62000)
2	Read all normal info, last read message numbers, and extra info for current member. (Line 62500)
3	Write all normal info for current member. (Line 61000)
4	Write all normal info, last read message number, and extra info for current member. (Line 61500)
5	Delete all support files related to current member. (Line 63000)

mdl.view-panel

Draw and update the View Panel. (Level 1)

1	Update the view panel. Called by login scripts.
2	Draw the view panel. Called by bootup script. Calls mdl.panel-draw to perform operation.

mdl.vote

Voting Booth. (Level 1)

1	Vote
2	Configure voting question. Called from SysOp Maintenance.

mdl.wait4call

Wait for a caller. (Level 1)

1	Save caller log and daily stats, clear current caller variables, and fall through to routine #2.
2	Wait for a call. Handles function key and hotkey commands, calls modem and login modules to answer calls, and runs idle and midnight scripts.
3	Save daily stats and caller log.

A.1.2 - Machine Language Code Files

Machine language file are only run from within another program module. Their names and descriptions are included here for completion of this listing only.

mlc.*-r

Files matching this pattern are the file receive coding of a file transfer protocol.

mlc.*-s

Files matching this pattern are the file send coding of a file transfer protocol.

mlc.ansi

ANSI emulation mode machine language code. Also used for RIP emulation.

mlc.ascii

ASCII emulation mode machine language code.

mlc.chattext

Chat mode entry and exit messages. Use the "chattext editor" program to edit this file.

mlc.commodore

Commodore color/graphics emulation mode machine language code.

mlc.main0 v4.0

Main BBS machine language coding. Version number may vary.

mlc.main1 v4.0

Main BBS machine language coding. Version number may vary.

mlc.screensaver

Some screen savers have this supplemental machine language code file.

mlc.swiftlink

Machine language device driver for a SwiftLink or Turbo232 cartridge.

A.1.3 - Overlays

Overlays form the lowest building block for the BASIC coding. An overlay must always be present in memory. Only *one* overlay may be in memory at once, and loading an overlay will remove any modules from memory. Overlays may be thought of as level 0 modules.

All overlays must start with the prefix "ovl.". There are only two overlays included with the base Centipede system. Other add-ons, such as networks, may add their own overlays.

ovl.main

Main overlay. This is the one that is in memory most of the time.

0	Boot the BBS.
1	Return to the current menu.
2	Continue with a script. Return to the current menu when complete.
3	Log the current caller off by running the auto-logoff script.
4	Call routine #1 of mdl.wait4call.

ovl.trap

Automatically run when a BASIC error has occurred and the system has been configured to intercept them. Displays error to screen and caller log, takes a picture of the screen, and takes steps to prevent further crashes. Returns to the current menu or logs the caller off if it is a repeat error.

A.1.4 - Miscellaneous

Following are any other program coding files that do not fit in any of the previous subsectins.

bbs

Special program run in order to boot the BBS.

chr.ansi

Special character set for use with ANSI and RIP emulation modes. Character sets for other emulation mods may also be created.

reboot

Special program run in order to reboot the BBS - accepting the default at any prompts during the bootup sequence.

A.2 - Support, System, and other Specialized Files

These files serve a supportive function to the BBS program code. The fields in the table are:

File Name: The file name may contain a wildcard character (* or ?). As in file matching in Commodore DOS, these wildcards can represent different characters. Such properties are explained in the description.

Location: This letter code defines the location where the file is stored. The letter codes are defined as:

A	Account Files Location	M	A Message Board Location
E	E-Mail Files Location	P	Default Program Files Location
F	A File Area Directory Location	S	Default Support Files Location
L	Caller Log Files Location	Y	System Files Location

Description: A description of the file.

accounts file	A	Membership account information. Stored in Account Files location.
application	Y	New user application questions.[4]
caller log	L	The Caller Log is stored in this REL file.
credits *	S	Download notification and credits awaiting awarding for member #'*'.
days stats	L	The day's statistics - number of calls, posts, new members, downloads, uploads, network posts, calls, total calls etc...
default msg ?	E	The default message created by the current caller on port '?'. '?' will always be '0' on a single-line BBS.
device id	Y	Information used to identify storage units.
√directory	F	File transfer area file directory. Stored in "var." format.
√dir.bak	F	Temporary √directory file during regeneration and a backup.

---

[4] The application file uses a special format. All text is simply displayed unless it is an input field. Input fields are indicated by placing a '#' character in the first column of a new line. After the '#', a format code defines the type of input allowed:

'f' → Free form. This is the default if no letter is defined.
'm' → Force Mixed upper and lower casing.
'n' → Number only.
'u' → Force Upper case only.

The format code may be optionally followed by an Extra Info Field number (1 through 10). EIFs can be viewed in Account Maintenance and used for certain purposes by add-ons.
The format code and/or EIF number may be optionally followed by the lower-case letter 'r'. This will store the input as the member's real name in his or her account.
The same EIF number or 'r' code may be present in multiple input fields. The answers will be concatenated together.
All codes are followed by a second '#' character, and then a short field name.
An example application file is included with Centipede.

dnlds today	S	Files (and their directories) that have been downloaded on this day. This data is used during the midnight reset to updated the "# DOWNLOADS" field in the file's extended description, and to award credits to the uploader.
email *	E	Member #'*'s E-mail.
email-dl *	E	Temporary file for e-mail message downloading.
feedback menu	S	Define this file to add a menu of SysOps for sending Feedback. It is a text file. Start with the member numbers of the people to be in the feedback menu. Starting with the first line that does not start with a number, the remainder of the file is shown to the caller as the menu. A sample menu is included. Deleting this file will result in feedback automatically going to member #1.
hlp.chat-req	S	List of chat request digitation descriptions and the hotkey to activate each. If you are not using multiple digitized chat request files, do not include this file in your system.
login msg	S	Picture or text displayed upon connecting to the BBS before the login prompt.
logoff msg	S	Picture or text displayed upon a normal logoff from the BBS under most system structures.
macro * ?	S	Member defined message macro - which may be added to the end of any public message or e-mail. The first number '*' is the caller's membership number. The second number '?' is the macro number. Macros are added to the end of a message by saving the message with the /s? command, where '?' is the macro number.
mailbox empty	S	Picture displayed if the caller does not have any e-mail waiting.
mailbox filled	S	Picture displayed if the caller has e-mail in his or her mailbox.
member memory	A	The membership list in a format that can be quickly loaded into memory at boot up.
membership full	S	Message displayed if a new user tries to join and there is no space for new members in your accounts file.
msg index	M	The message board's index file - containing message number, category, and threading information.
new user msg1	S	Displayed to a new user in between entering a name and password, and the new user application.
new user msg2	S	Displayed to a new user after filling out the application.
nothing flagged	S	Displayed if a caller attempts to perform an operation on all flagged files, and nothing has been flagged.
password msg	S	Displayed before a member enters a new password.
phrases	Y	Menu phrases. Menu phrases are displayed at random before a menu prompt. This file is created by the Copy Phrases utility.
raw.chat-?	S	Chat request digitation file, where '?' is the hotkey to select that digitation.

raw.chat-req	S	Default chat request digitation - or the only digitation played if the hlp.chat-req file does not exist. If this file does not exist, a telephone ring sound effect is used instead.
raw.logoff	S	Digitation to play when someone logs off the BBS.
sysop news	S	Storage file for the SysOp News. Use the SysOp News Editor under SysOp Maintenance to edit and view.
sysop not here	S	Displayed after a chat request has been made and the SysOp has not responded.
upload msg	S	Displayed before a caller uploads files.
user * info	S	Member #'*'s original application, if you have chosen in bbs-setup to save them.
var.bootup	Y	Extra parameters and variables to be loaded at bootup, that are not defined by bbs-setup.
var.main	P	Main BBS parameters and variables, as defined in bbs-setup.
var.protocols	S	File protocols defined for use on the BBS.
var.screensaver	S	Parameters for the current screen saver. Not all screen savers may use this file.

A.3 - Help Files

All of the following files are stored in the Help Files location.

hlp.*	Each menu that you define (mnu.*) has an associated help file that is displayed if the caller presses '?' from the menu prompt.
hlp.download	Displayed when the caller presses '?' from the download prompt: "[U]nflag, [F]lag, [P]rotocol, RETURN=Continue"
hlp.editor	Command list for text message editor's *edit* mode. Displayed if caller presses '?' at the "Edit:" prompt in the editor.
hlp.editor-full	Command list for the full-screen editor. Displayed if the caller presses 'HOME ?' while in the full-screen editor.
hlp.editor-line	Command list for the line-based text editor. Displayed if the caller enters /? on a line.
hlp.email-read	Command list for the E-mail reader. Displayed if the caller presses '?' at the prompt between e-mail messages.
hlp.fileselect	Command list for the full-screen interactive file selector.
hlp.login	Information on how to log into the BBS. Shown before the membership list at the login prompt.
hlp.protocol	Description of the different file transfer protocols. Displayed if the caller presses '?' from the protocol selection prompt.

A.4 - Script Files

Scripts are collections of commands that are run in sequence. See section 5.8 for details. The following scripts are used on all systems no matter your chosen structure. They have the file name prefix "scr." and are stored in the Program Files location.

The tables below list describe each of these scripts and their default contents. Using different system structures and add-ons may modify these scripts. You may also choose to modify them yourself.

scr.auto-off
Automated logoff - usually used in the event of a carrier lost or of the caller chooses to auto-logoff after downloading files.
run #2 in "mdl.logoff" as "Logoff" with kill=none, gfx=any run #1 in "mdl.mb-purge" as "Purge Msgs" with kill=none, gfx=any run #2 in "mdl.history-log" as "Update" with kill=none, gfx=any run #1 in "mdl.wait4call" as "Wait4Call" with kill=none, gfx=any

scr.bootup
Completes the bootup sequence after mdl.bootup has finished.
run #1 in "mdl.date&time" as "Set" with kill=none, gfx=any run #1 in "mdl.regen-mb" as "Check Indexes" with kill=none, gfx=any run #2 in "mdl.view-panel" as "Draw Panel" with kill=none, gfx=any run #4 in "mdl.acnt-maint" as "Load Member List" with kill=none, gfx=any run #1 in "mdl.wait4call" as "Wait4Call" with kill=none, gfx=any

scr.inst-login
Performs an instant local login direct to the main menu. Activated by pressing 'F2' or 'I' from the Wait4Call screen.
run #2 in "mdl.view-panel" as "Update" with kill=none, gfx=any run #0 in "mnu.main" as "Main Menu" with kill=none, gfx=any

scr.login
Normal login sequence -run after the caller has successfully entered an account and password.
read "caller stats" run #1 in "mdl.view-panel" as "Update" with kill=none, gfx=any run #1 in "mdl.sysop-news" as "Display New" with kill=none, gfx=any run #2 in "mdl.email" as "Check" with kill=none, gfx=any run #1 in "mdl.credits" as "Check" with kill=none, gfx=any run #0 in "mnu.main" as "Main Menu" with kill=none, gfx=any

scr.logoff

Normal system logoff.

```
prompt "Log Off Now (Y/N)?",N
read "logoff msg"
run #2 in "mdl.logoff" as "Logoff" with kill=none, gfx=any
run #1 in "mdl.mb-purge" as "Purge Msgs" with kill=none, gfx=any
run #2 in "mdl.history-log" as "Update" with kill=none, gfx=any
run #1 in "mdl.wait4call" as "Wait4Call" with kill=none, gfx=any
```

scr.midnight

New day reset script.

```
run #2 in "mdl.date&time" as "Set New Date" with kill=none, gfx=any
run #1 in "mdl.midnight" as "Full reset" with kill=none, gfx=any
run #2 in "mdl.credits" as "Purge Old" with kill=none, gfx=any
run #3 in "mdl.history-log" as "New Day" with kill=none, gfx=any
run #1 in "mdl.wait4call" as "Wait4Call" with kill=none, gfx=any
```

scr.new-hour

Runs at the beginning of each new hour when at the Wait4Call screen. On a stock system, nothing productive actually happens.

```
display "Checking network windows"
run #1 in "mdl.wait4call" as "Wait4Call" with kill=none, gfx=any
```

scr.new-login

Application and login process for a new user.

```
run #1 in "mdl.apply" as "New user" with kill=none, gfx=any
read "caller stats"
run #1 in "mdl.view-panel" as "Update" with kill=none, gfx=any
run #3 in "mdl.sysop-news" as "Display all" with kill=none, gfx=any
run #0 in "mnu.main" as "Main Menu" with kill=none, gfx=any
```

scr.quick-login

Quick login sequence -run after the caller has successfully entered an account and password.

```
read "caller stats"
run #1 in "mdl.view-panel" as "Update" with kill=none, gfx=any
run #3 in "mdl.sysop-news" as "Display all" with kill=none, gfx=any
run #0 in "mnu.main" as "Main Menu" with kill=none, gfx=any
```

scr.sys-idle

Runs after the system has been idle at the Wait4Call screen for a period of time as defined in bbs-setup. On a stock system, nothing productive actually happens.

```
display "Checking network windows"
run #1 in "mdi.wait4call" as "Wait4Call" with kill=none, gfx=any
```

scr.wait4call-?

'?' is a single character key press. This script is run when that key press is made from the Wait4Call screen. The actual contents of the script varies from one to another - and none are defined on a stock system.

B - MCI Commands

MCI stands for Message Command Interpreter. It allows you to insert command codes into messages, which are interpreted by Centipede when the message is displayed.

All MCI commands start with the attention character '£'. On Commodore computers, this appears as an English Pound symbol. On most other systems, it is a backslash '\'. The £ is immediately followed by a command parameter, usually a digit.

B.1 - Simple MCI Commands

The following is a list of the simple MCI commands:

£c# Changes the color of the text. The number or letter you place for # will change the active text color to:

0 = Black	8 = Dark Purple
1 = White	9 = Brown
2 = Dark Red	A = Light Red
3 = Light Cyan	B = Dark Cyan
4 = Light Purple	C = Medium Grey
5 = Dark Green	D = Light Green
6 = Dark Blue	E = Light Blue
7 = Yellow	F = Light Grey

This color list is for display on a C128 in 80 columns. A C64 or C128 in 40 columns, and an ANSI terminal, will show some colors slightly differently.

£d# Where # is: 0 to home the cursor or 1 to clear the screen and home the cursor.

£g# Get key press. # is the key press buffer number of 0 through 3.

£h# Send # back-space/delete characters.

£k# Rainbow mode. 0=off, 1=letter, 2=word, 3=line.

£n# Send # carriage returns.

£p# Output content of key press buffer #.

£r# Reverse mode, where # is 0 for reverse off or 1 for reverse on.

£u# Case, where # is 0 for lower/upper case or 1 for upper case/graphics.

£v# Output information:

0 = Access level	8 = Morning/Afternoon/evening
1 = Time left	9 = Last caller
2 = handle	A = The BASIC variable a$
3 = Credits	B = Box prompt
4 = Baud rate	C = BBS name
5 = Date last call	D = RETURN or ENTER
6 = Current date	E = Emulation mode
7 = Current time	F = The BASIC variable f$

£w# Pause for 1 to 8 half seconds.

£x Skip rest of file.

£z Output rest of file without emulation interpretation. (i.e. raw data)

£@# Skip to tab stop. Stops every 8 characters. 0=column 1, 1=column 8, 2=column 16, etc.

£^# Cursor up # lines.

£!# Cursor down # lines.

£># Cursor right # columns.

£<# Cursor left # columns.

ŁŁ Display a single £.

B.2 - Advanced MCI Commands

These commands are for advanced use only. They are designed for SysOp use. Some of them only interpret properly when contained within a file that is being displayed from disk. These commands will thus not show when editing a message.

DISPLAY VARIABLE:

The £[ ] command displays any BASIC variable, or the result from any BASIC expression. It will not display past a null character in a string variable. This fact is used to prevent the viewing of passwords by starting all password variables with a null character.

Examples:

£[na$]	Displays the caller's name.
£[d$(6)]	Displays the caller's real name.
£[mn(m)]	Displays the number of the posting currently being displayed.
£[pd/d%(3)]	Divides the number of blocks the user has ever uploaded by the number of calls made. I.e., Average blocks downloaded per call.

Variables and expressions must be enclosed within brackets. The closing bracket may be replaced with a carriage return.

This MCI command, and *only* this command, may be used within file names throughout most of the BBS.

JUMPING:

The £j1 command will skip all text in a file until a £m1 is found, or the end of the file is reached. The '1' is a marker identifier and may be replaced by any character, providing you with the ability to have multiple markers in the same file. Markers are skipped without being displayed.

The £x command will immediately stop reading the file. All following text will be ignored. A £x that is jumped over will of course be ignored.

CONDITIONAL JUMPING:

Using syntax similar to the expression displayer and the jumper, we can make jumps based on the value of variables or user responses. The MCI command £?expression] preforms the test. The conditional jump £t1 will jump only if the last expression evaluated to True. £f1 will jump only if the last expression evaluated to False. As with the £j command the '1' marker identifier can be replaced with any character, and a corresponding £m marker is needed.

Example:

The BASIC variable `lv` represents the caller's access level. We can give names to access levels using conditional jumping:

```
£?lv=1] £f2 NEW USER £je £m2 £?lv>7] £ts VALIDATED £je £ms SYSOP £me
```

Spaces were added in order to make the example easier to read. You'll want to be careful with using space however, because some of them will be displayed to the reader.

You may also branch based on input from the reader. Refer to the £g command for this example:

```
Press Q to quit, or any other key to continue: £g1£?'1'="Q"]£f1£x0£m1
```

The user input is placed into MCI key press storage register #1. There are four registers numbered 0 through 3. By placing the register number in an expression surrounded by single quotes (apostrophes), you can test for the value of the key press. The stored value of a key press will always be in uppercase.

A restriction in the conditional testing coding requires that the expression does *not* end with a '$' character. So `£?"Sysop"=na$]` will give an unpredictable result, whereas `£?na$="Sysop"]` will test to see if the name of the caller is "Sysop". In the case of something like `£?a$=b$]` where reversal will not help, you can simply surround the expression with parenthesis: `£?(a$=b$)]`.

C - SCRIPT LANGUAGE

Centipede has a simple scripting language, which is compiled within the Script Editor utility. The following explains each of the scripting commands available.

```
support = device,"prefix:","command1","command2"
support = 0
support = default
```

Set the current support files location for use in all the remaining commands in the script, or until the next `support=` command. Command2 is optional. Setting to `0` or `default` sets the location of the support files to the default support files location. If the `support=` command is never used in a script, `default` is assumed.

```
program = device,"prefix:","command1","command2"
program = 0
program = default
```

Set the current program files location for use in all the remaining commands in the script, or until the next `program=` command. Command2 is optional. Setting to `0` or `default` sets the location of the program files to the default program files location. If the `program=` command is never used in a script, `default` is assumed.

```
run #routine in "filename" as "command name" with
      kill= no/none/dim/array/var/variable/all and
      gfx= any/ascii/ansi/ibm/commodore/c=
      at level lv
      prompt
      lock
```

Run the subroutine number *routine* in the module, overlay, or variable file *filename*. All other syntax is optional. The words *in*, *as*, *with*, *kill*, *and*, *gfx*, *at*, *level* , and the equal signs are all optional and ignored. Their parameters do have an effect if used though:

kill:	Enter the type of killer to be used. Default is all. The following combinations are synonymous: `no`/`none`, `dim`/`array`, `var`/`variable`/`all`.
gfx:	Required graphics emulation mode. Default is any. The following combinations are synonymous: `any`/`ascii`, `ansi`/`ibm`, `commodore`/`c=`.
level:	Required minimum access level. Default is 1.
prompt:	Give the caller a yes/no prompt asking whether or not to run this line of the script. The *command name* will be displayed as the prompt.
lock:	The lock flag will only allow one port to run this routine on a multiplexed system.

```
Prompt "text",c
```

Displays the prompt text to the caller and waits for a 'Y' or 'N' key press. The script will be aborted if the caller pressed the letter defined by `c` or `CONTROL-P`.

```
Display "text"
```

Display the *text* to the screen. Simple MCI commands may be imbedded into the text.

```
Read "filename" at level lv
```

Display the SEQ file *filename* from the support files location to anyone of level *lv* or higher. The words *at* and *level* are optional. `lv` defaults to `0`, thus eliminating the level restriction.

`ifport=nd, end`
`ifport=nd, skip n`

This command is used by mutliplexed dual-port systems only.

If the current port number (0 or 1) is *nd*, then either abort the rest of the script or skip the next *n* commands in the script.

Note: `support=` and `program=` lines do not count as commands when skipping!

D - KEYBOARD COMMANDS

D.1 - CONTROL Key Commands

The CONTROL key commands are produced by pressing and holding the CONTROL key, while pressing the listed letter key. Unless otherwise noted, these character commands can be entered from online (such as in messages), or included in program code.

CONTROL-A: Turn rainbow mode on.

CONTROL-B: Change the background color if followed by a color key, otherwise turn character underlining on.

CONTROL-C: Cycle to the previous system color.

CONTROL-D: Cycle to the next system color.

CONTROL-F: Turn character flashing and underlining off.

CONTROL-G: Produce a bell sound.

CONTROL-J: Produce a carriage return. Works from within program code only.

CONTROL-K: Copy a character entered at the current column position on the last line entered. (Certain restrictions apply for security reasons.)

CONTROL-O: Turn character flashing on.

CONTROL-P: Abort current operation when online, or produce a soft return in when in program code. (A soft return will output a space when still on the first half of an 80c display.)

CONTROL-S: Pause output. This is the same as the HOME key on a Commodore.

CONTROL-V: Verify current line being entered by redisplaying it on the next line. Useful for checking for line noise.

CONTROL-W: Word delete; delete back to last space.

CONTROL-X: Delete back to the beginning of the line.

CONTROL-Z: Turn rainbow off.

D.2 - ALT Key Commands

The ALT key commands are produced by pressing and holding the ALT key, while pressing the listed command key. These commands only work from the local computer that the BBS is running on, and thus cannot be activated by a caller.

ALT-1 ...
ALT-8: Display the numbered programmable ALT key definition, as defined in bbs-setup.

ALT-C: Toggle Auto-Color Changing in chat mode. With this feature active, anything you type will be shown in the first system color, and anything the caller types will appear in the second system color.

ALT-G: Produce a bell sound on the BBS computer only. This is useful for adjusting the audio

volume.

`ALT-S`: Toggle Split-Screen Chat Mode. When active, the chat screen will be split into two halves, one for the SysOp and one for the caller.

`ALT-W`: Toggle Word-Wrap in chat mode.

`ALT-O`: Clear the system buffer.

`ALT-+`: Start capturing output into the system buffer.

`ALT--`: Stop capturing output into the system buffer.

`ALT-@`: Dump the contents of the system buffer, as if it were typed into the keyboard.

In addition to these commands, simply holding the `ALT` key while a script is executing will prompt you for execution of each command in the script.

D.3 - Function Key Commands

When a member is online, there are a few commands that can be executed by pressing the F-keys on your keyboard:

`F1`: From nearly anywhere in the BBS, you may press `F1` to jump into chat with the caller.

`F3`: Pressing `F3` will exit chat mode, having frozen the caller's time for the day while in chat mode.

`F4`: Pressing `F4` will exit chat mode, while having allowed for the caller's time for the day to be used up during the time spent in chat.

`F5`: Edit the caller's access level, time for the day, or credits. You may also put the caller on hold, so that you may roam through the BBS with the caller online, but unable to see anything you are doing or interfere with you. This command *only* works from a menu prompt.

`F7`: Run a program module. You will be prompted for a location, file name, and routine number. This command *only* works from a menu prompt.