BINKLEYTERM

A Freely Available FidoNet Compatible
Electronic Mail Interface and Dumb Terminal Package

VERSION 2.60

User Guide

April 1996

Software Written by Vince Perriello and Bob Hartman,
with contributions from countless others.

Original Documentation written by Alan D. Bryant
Revised by Barrie Smith and W.R. Davis

BIT BUCKET SOFTWARE, CO.
P. O. Box 460398
Aurora, CO 80046

Table of Contents

GENERAL INFORMATION

HOW TO USE THIS MANUAL

The documentation for BinkleyTerm is supplied in two main parts.

This User's Manual (named BT_USER.DOC) explains how to install BinkleyTerm. It also describes basic operational procedures.
New users may find some concepts or terminology unfamiliar; a glossary is provided in the BinkleyTerm documentation (named Glossary.Doc)

The Reference Manual (named BT_REF.DOC) gives details of all the available configuration file statements, details of event file usage and much other information.

For inquiries, questions or comments regarding BinkleyTerm, please refer to the section in this User Guide titled "BinkleyTerm Support."

ACKNOWLEDGMENTS

The following names are either trademarks, registered trademarks, and/or the efforts of the person and/or company named:

PRODUCT	AUTHOR
AMAX	Alan Bryant
ARC, ARCmail, GroupMail, SEAdog, SEAmail, SEAlink, XlatList	Thom Henderson, System Enhancement Associates, Inc.
Atari, ST	Atari, Inc.
BGFax	B.J. Guillot
BNU	David Nugent
BONK, OOPS	Tom Kashuba
Commodore, Amiga	Commodore International
ConfMail, ParseLst, oMMM, Opus! Comm	Bob Hartman, Spark Software, Inc.
D'Bridge	Chris Irwin
DEC, Rainbow, VAX, VAX/VMS, VT-100	Digital Equipment Corporation
DECCOMM	Vince Perriello, VEP Software
DESQview	Quarterdeck Office Systems, Inc.
DoubleDOS	SoftLogic Solutions, Inc.
Dutchie	Henk Wevers
EchoMail	Jeff Rush
FANSI-CONSOLE	Hersey Micro Consulting
FastLst	Alberto Pasquale
Fido, FidoNet	Tom Jennings, Fido Software
FrontDoor	Joaquim Homrighausen
Hayes	Hayes Microcomputer Products Corporation
Heath/Zenith	Heath/Zenith Electronics, Inc.
Hydra	Arjen Lentz, Lentz Software Development, Joaquim Homrighausen
IBM, PC-DOS, OS/2	International Business Machines Corporation
InterMail	Peter Stewart

PRODUCT	AUTHOR
Janus	Rick Huebner
Maximus BBS, Squish	Scott Dudley, Lanius Corp.
MS-DOS, Microsoft Windows, Microsoft Windows NT, Microsoft C	Microsoft Corporation
msged	Jim Nutt
MsgedSq	John Dennis
oMMM	BS Software, Marshall Presnell, Jim Nutt
Opus-CBCS, ZedZap, YooHoo, WaZOO, OpusNode	Wynn Wagner III
PC Pursuit	GTE/Telenet
ProComm	Datastorm Technologies, Inc.
QM (echomail processor)	George Peace, LNH Software
Qmodem	The Forbin Project
QuickBBS	Adam Hudson
RBBS-PC	Capital PC Users Group, Thomas J. Mack
Sanyo	Sanyo Electric Co. (Japan) Ltd.
SIO, VSIO, VX00, X00	Ray Gwinn, The Software Division
Sirius	Bob Klahn, Micro Solutions
TBBS, TIMS	Phil Becker, eSoft, Inc.
Telix	Colin Sampaleanu, Exis Inc.
Timed	Gerard van Essen
Unix	Unix System Laboratories, Inc.
US Robotics, HST	US Robotics, Inc.
WinFOSSIL	Bryan Woodruff
XlaxNode	Scott Samet
Zmodem	Chuck Forsberg, Omen Technologies, Inc.

This manual is derived from the BinkleyTerm 2.30 documentation which was written by Alan D. Bryant. In the BinkleyTerm 2.30 documentation, some material in the section "How BinkleyTerm Handles Mail" was excerpted from MATRIX.DOC, Copyright (C) 1987 Wynn Wagner III, All Rights Reserved. Used by permission. Also in the BinkleyTerm 2.30 documentation, some material in the section "Control" was written by Ron McKenzie, and is Copyright (C) 1989 Ron McKenzie, All Rights Reserved. Used by permission. As it is unlikely that we have replaced all of this work, we again credit and thank the individual authors involved.

Every effort has been made to identify and give credit for trademarks mentioned in this documentation. Any failure to mention a particular trademark in the above list that may be found in the text, or failure to give proper credit for a particular trademark, constitutes merely an oversight and should not be construed as intentional, or in any way a claim of rights to the trademark.

PLEASE NOTE! Throughout this documentation, the mention of any particular software package or system should not be construed as an endorsement of any kind on the part of the authors.

Alan Bryant, who has in the past done such a splendid job of documenting previous versions of BinkleyTerm was unable to prepare the docs for this new version.

Barrie Smith and Bob Davis took on the task of collating the various sections of the docs , though Bob later had to drop out.

By all means send netmail to Barrie at 2:440/130, or use the Binkley echo, to point out errors or places where the wording is less than perfect, but keep it friendly.

KUDOS

A number of people have been involved in the creation, development and testing of this program, or have contributed to it in one way or another.

Tom Jennings, of course, gets credit for creating the beast that brought us all together and defining the basic method of mail transfer that we still use. Wynn Wagner gets our thanks for releasing source code for his file transfer routines and sending some of his WaZOO source code as well for inclusion.

Thom Henderson gets a humble tip of the hat for SEAdog, the prototypical mail-only front-end which was the basis for our front-end design, and for SEAlink, his extension of Xmodem/Telink which doubled the speed of network mail transfers overnight. Chuck Forsberg of Omen Technologies gets our thanks for developing and implementing the original Zmodem protocol. Rick Huebner is a virtual unknown in FidoNet, but his work was critical both for Opus-CBCS and for BinkleyTerm. Rick did the original adaptation of the Zmodem file transfer protocol for Opus, a derivative of which still lives in BinkleyTerm; he designed the WaZOO File Request mechanism using .REQ files; and finally, his design and implementation of Janus proved that full-duplex mail transfers and file requests were both viable and crucial for FidoNet.

There are many individual members of FidoNet who have responded to our challenge or to our licensing agreement and sent us code for worthwhile improvements and bug fixes to BinkleyTerm. For this release, the undisputed superstar of the group is Michael Buenter, who built a vastly extended version of BinkleyTerm 2.50 called BinkleyTerm Extended Edition, and then helped us incorporate most of his work into the common codebase. Michael has also helped debug some of the late-coming features of BinkleyTerm 2.60 and is continuing to work with us to help merge the Unix and DOS-variant codebases. Jim Dailey is another who has provided a number of new features and assistance in the OS/2 effort.

For the 2.60 effort, the one guy who really stuck it out and helped make it happen was Bob Juge. While the developers were essentially on sabbatical, he just stayed on and on and on as BinkleyTerm help, calling occasionally to ask the odd question. His help down the final home stretch was critical, including doing the actual builds of the OS/2 version and personally coordinating the OS/2 testing.

For this documentation, Barrie Smith and Bob Davis did a fine job of collecting all the various inputs and assembling something coherent.

Finally, there are the guys who have put to the test everything the authors thought would be worthwhile, the BinkleyTerm Beta Test Team, members both past and present. Without their efforts, who knows what we'd have wound up with. There are several others who made noteworthy contributions to the BinkleyTerm 2.60 effort. We call your attention to the file THANKS.TXT in the distribution archive.

FOREWORD

There is no question that BinkleyTerm is an extremely powerful communications tool. We also make no secret of the fact that BinkleyTerm is an extremely complex communications tool.

A set of documentation the size of an unabridged dictionary (about 150 mm thick or so) would still not address every possible use of BinkleyTerm in every possible environment.

BinkleyTerm can run on a number of personal computer platforms - a total of several thousand brands and models. Hundreds of different modems can be used. It works with several DOS operating environments, such as DESQview, DoubleDOS, Microsoft Windows and Windows for Workgroups.

Versions which operate natively on OS/2 and on Win32-based systems such as Windows 95 and Windows NT have also been released by Bit Bucket Software.

In addition, BinkleyTerm is designed with an open architecture, so it can be used with several BBS packages, nodelist processors, mail processors, editors, and so on. BinkleyTerm "ports" have been made for entirely different platforms such as the Atari ST, Commodore Amiga, and Unix and VAX/VMS based systems.

You begin to get the scope of what's involved.

This documentation attempts to cover fairly broad generalities in configuring and operating BinkleyTerm. It cannot and will not cover all possibilities or circumstances. Hopefully it will serve as a starting point: a ground level from which you may grow and expand. Further, this documentation describes only the versions of BinkleyTerm provided by the original authors: the versions which run in the PC-DOS/MS-DOS environment, the OS/2 environment, and the Win32 environment. Because the heritage of BinkleyTerm is rooted in the DOS environment, however, there will be more coverage of this area than the others.

Most of the enjoyment of the electronic mail hobby comes from trying new things - tweaking the system. Although BinkleyTerm is a dependable, powerful program that is not especially difficult to get going, it does provide ample opportunity to experiment and have fun. However, if you're looking for something that will meld itself into your computer in just a few minutes and work without modification forever more, BinkleyTerm should probably not be your first choice. In exchange for a little complexity, we give you power and an incredible amount of configurability and compatibility.

If you become frustrated in your effort to configure or operate BinkleyTerm, we suggest that you call on others in your area who have configured it for an environment similar to yours. We estimate that several thousand people around the world are BinkleyTerm users, and someone close to you in your network no doubt runs BinkleyTerm. With an architecture as open as that of BinkleyTerm, your best source of information is someone who has the benefit of time and experience configuring it. Of course, you will eventually become an expert yourself! Enjoy it, and delight in the wonder of dialup electronic mail technology.

Vince Perriello, Alan D. Bryant, Bob Hartman
Bit Bucket Software, Co.

WHAT IS BINKLEYTERM?

BinkleyTerm is an advanced, state-of-the-art telecommunications tool. It is primarily designed for the semi-automated sending and receiving of electronic mail and files within FidoNet- compatible electronic mail networks.

BinkleyTerm can be used as a dumb terminal program exclusively, as a mail interface for a Point system or FidoNet node, or as a front end mail interface for an electronic bulletin board system (BBS).

When used as a mailer, BinkleyTerm is designed to communicate with any FidoNet-compatible mail interface or BBS package. The program uses standard FidoNet protocol, as well as certain modified protocols supported by programs such as SEAdog and Opus-CBCS. It also offers easy-to-use event scheduling, single keystroke spawning to other programs (Command shell), excellent support for high speed modems, advanced session recovery, inbound call collision detection, and much more.

When used as a dumb terminal, BinkleyTerm offers a rich selection of file transfer protocols for exchanging files with a host system. The program also offers keyboard macros, optional VT-100 emulation, echoing of the on-line session to a flat text file or printer, support for baud rates of 300 to

115,200 and more. Mail interface and dumb terminal operations can be switched in and out with a minimum of effort, providing dual functionality.

In the DOS environment, BinkleyTerm is one of several software packages to utilize the FOSSIL standard communications driver. This standard allows for consistent console (keyboard and screen) and communications port I/O operations. By using a FOSSIL driver, BinkleyTerm is capable of running on practically any MS-DOS/PC-DOS capable machine, even those that are not 100% IBM hardware compatible.

BinkleyTerm endeavors to provide you with the widest possible variety of advanced features, combined with solid and efficient operation in a wide range of hardware environments.

BinkleyTerm features a windowed user interface. The windowed interface provides "at-a-glance" convenience for watching mail sessions in progress, as well as determining what activity has taken place with the system recently.

Details of the interface windows are given in the Reference Manual.

BINKLEYTERM REQUIREMENTS

GENERAL REQUIREMENTS

Minimum requirements for running BinkleyTerm are:

1. A personal computer running MS-DOS or compatible operating system, OS/2, Windows 95 or Windows NT.

2. If running MS-DOS:

   a) At least 320k of available RAM. See "Memory requirements under MS-DOS" for detail.
   b) MS-DOS Version 2.10 or greater, Version 3.20 or higher preferred.
   c) A FOSSIL driver designed for your particular hardware. (BNU and XOO are two in common use).
   d) A Video FOSSIL driver designed for your hardware. This is necessary for color displays and recommended in all cases as the windowed interface runs much faster. (VFOS_IBM is in common use).
   e) A hard disk is required to use BinkleyTerm as a front end mailer (the more space the better) but a single floppy disk drive of 1.2 Mb capacity or greater is enough to run in Terminal mode.

3. If running OS/2,
   a) Physical memory of at least 4 megabytes
   b) OS/2 Version 2.1 or Warp.
   c) Communications support driver must be loaded. COM.SYS is available in OS/2 but Ray Gwinn's SIO.SYS preferred.

4. If running Windows 95, Physical memory of at least 8 megabytes

5. If running Windows NT, Physical memory of at least 16 megabytes

6. An auto-dial, auto-answer modem; should be mostly "Hayes compatible." See below.

7. Basic knowledge of computer-based telecommunications, without which you would have no need for BinkleyTerm.

You will also need various utilities and adjunct programs, which vary with your application.

MEMORY REQUIREMENTS UNDER MS-DOS

BinkleyTerm requires approximately 320k of RAM in any operational mode. When used for Unattended Mode, additional memory sufficient to hold the nodelist index file (usually NODELIST.IDX) will also be required. Keep in mind that a small amount of overhead will also be required to accommodate a FOSSIL driver, as well as a Video FOSSIL if one is used. At least a 512k system is recommended when using Command Shells, or when BinkleyTerm is used in a multi-tasking environment. BinkleyTerm should, however, be completely functional even on a system with only 320k of RAM.

BinkleyTerm uses Thomas Wagner's excellent public domain swapper code. This gives BinkleyTerm XMS or EMS swapping by default, and swapping to a file if XMS or EMS is not available (or insufficient). It also uses the create-temp-file function in DOS if available.

Please note, however, that when using the 'SwapDir' configuration file statement, a RAM disk with at least 400k of available space is highly recommended for efficient operation.

MODEM REQUIREMENTS

A modem that is generally "Hayes compatible" is required for BinkleyTerm operation. Such modems are typically referred to as "smart" modems. Most popular, late model modems you're apt to own meet this requirement. Since you configure the various modem command strings, the modem does not need to be 100% Hayes "AT" command set compatible, but it does need to use the "AT" style of issuing modem commands.

Most smart modems are capable of returning VERBAL (full words) or NUMERIC (numbers only) response codes; the modem must be configured in such a way as to use the VERBAL codes ONLY. BinkleyTerm can be operative if the modem supports just one modem response string - CONNECT.

Generally, CONNECT is a default, and indicates a 300 baud connection. The modem normally returns a connect string of CONNECT followed by the baud rate of the connection e.g. CONNECT 2400.

The connect rate is recognized by BinkleyTerm up to the maximum supported speed of 115200 (including 7200, 12000, 14400 and other rates which may be reported by some newer modems). BinkleyTerm also recognizes some extra messages issued by certain old and slow modems, i.e., "CONNECT" followed by 103 (giving 300bps), 75 or 1275 (giving 1200/75bps), and 12 or 212 (which give 1200bps).

By default, BinkleyTerm has been programmed to handle and respond appropriately to the following modem response strings: NO ANSWER, BUSY, RING, VOICE, NO DIAL TONE, NO DIALTONE, ERROR, NO CARRIER, DIAL TONE and DIALING. The following response strings are recognized, but are ignored: RRING, RINGING and RING RESPONSE.

When using its default settings, BinkleyTerm will ignore any other data provided by the modem after a given response string, with the exception of the CONNECT response. For example, "NO CARRIER DETECTED" would be handled in the same manner as a "NO CARRIER" response.

BinkleyTerm may be reconfigured to replace the default response strings mentioned above, for special modem requirements.

BinkleyTerm needs to be told details of your modem in the Configuration File. One essential item is the "Baud" statement which tells BinkleyTerm the maximum rate at which the computer should talk to the modem (the DTE rate, see below).

Note that BinkleyTerm does not recognize 14400, 12000 or 7200, the so called "three-quarter rates" for purposes of the "Baud" statement. If you have a modem capable of these speeds set "Baud" to a higher full rate such as 19200 or 38400.

BinkleyTerm does *understand* the result codes CONNECT 14400, CONNECT 12000, and CONNECT 7200 and will pass these as the connect rate, but will pass the link rate as the next highest "legal" link rate.

As mentioned under "General Requirements", BinkleyTerm running under DOS uses a Fossil driver. The Fossil standard provided for baud rates up to 38400 but improvements in Modems etc., mean that higher baud rates may be needed for optimum operation.Version 1.53a of the X00 FOSSIL driver has an extension which supports additional baud rates.

This is not a standard so, if you want BinkleyTerm to lock the baud rate the program needs to be told to use this extension by means of an "ExtBaudRates" statement (which must appear in the Configuration file *before* any other statements about baud rates). BinkleyTerm can then support up to 115200 baud. Note: it is possible to lock the baud rate directly by a Fossil command (this can be done with either BNU or X00) in which case there is no need, and no point, in telling the program to lock the baud rate as BinkleyTerm is then unable to change the baud rate.

Win32 and OS/2 are not subject to the same limitations and definitions for 57600 baud and 115200 baud are included in BinkleyTerm for these versions. Note that for the OS/2 version, you must have a version of MAXCOMM.DLL which supports rates of 57600 and 115200. You need version 2.5 or above to support 115200 baud. Alternatively, you could use Gwinn's very popular SIOCOMM.DLL after using its HACKIT.EXE utility to modify BT32.EXE

Modem technology continues to improve and error correction and data compression are now both commonly employed to increase communication speeds. Because the modem offers data compression it is necessary to send data from the computer to the modem at a rate much higher than the rate at which the modem sends data "down the line". There seems to be some ambiguity in terminology, but the rate that data is sent from the computer to the modem is often called the link rate or the DTE rate, while the modem to modem or line rate is referred to as the DCE rate.

It is now quite common to "lock" the DTE rate so the computer always sends at a set speed and if the modem cannot accept characters at the rate they arrive then a hardware method of flow control is employed to tell the computer when to stop and start sending.

Locking can be accomplished most easily by locking the Fossil (by a simple command in the fossil configuration file). For many modems this is the preferred method. Both X00 and BNU can be locked in this way. (Hint: if using BNU stick to version 1.70, version 1.88 is a hack)

A "Baud" statement is necessary in the BinkleyTerm configuration file, but in fact, if the Fossil is locked BinkleyTerm cannot then alter the locked rate.

Fax reception adds a further complication. In most cases where fax reception is required the fossil is NOT locked and control is achieved by use of "FaxBaud" and "Lockbaud" statements (See the Fax section in the Reference Manual).

The "Autobaud" statement confuses some users. Its only purpose is to tell BinkleyTerm to take no notice of the rate given for a node in the nodelist but to *call out* at the rate given in the baud statement. This may help to achieve a connect at the highest possible rate if, for example, the node in question has not yet updated his nodelist entry after acquiring a faster modem.

Other Statements concerned with baud rates are:

BiDiBaud	which controls the bi-directional baud rate.

FaxBaud	which sets a user defined rate for Fax.
JanusBaud	Synonymous with BiDiBaud q.v.

If you are experiencing modem difficulties, try one or all of the following configuration file statements: 'SlowModem', 'SameRing' and/or 'NoCollide'. Additionally, not all modems are capable of using the 'Answer' statement and its related modem configuration. Explanations of all these statements can be found in the Reference Manual section, "Configuration File."

Users with modern High speed error correcting modems may wish to refer to the section on that subject later in this manual at page 28

Tip: Get a simple setup working before you try the more complex stuff.

INSTALLATION OF BINKLEYTERM

Due to the complexity and widely varying combinations of hardware and software, only a generalized, broad explanation of the installation procedure can be given.

General notes, applicable to all installations are given first and are followed by notes which apply to DOS, Windows for Workgroups, Win32 and OS/2, respectively.

INSTALLATION NOTES, GENERAL

1) Make a subdirectory on your hard disk and place the BinkleyTerm files from the archive in that directory.
If you only intend to run BinkleyTerm in Terminal mode then it is possible to put the files on a High Density floppy and run from there.

2) Most BinkleyTerm parameters are set through a configuration file. By default, the configuration file is named BINKLEY.CFG, and is expected to be available unless a different configuration file is specified on the command line.
If you are new to BinkleyTerm use any standard text editor to look at and make the necessary changes to the NEWUSER.CFG configuration file included with the distribution archive. The name of the file must be changed to BINKLEY.CFG before use.

A configuration guide list, showing which statements are essential etc., is also in the package under the title CFGGUIDE.TXT.

The Reference Manual contains a complete listing of all BinkleyTerm configuration statements and their proper use.

Each time the program is invoked, BinkleyTerm will scan and process the raw configuration file, setting internal values to those that you have selected.

3) If you are an "old hand" look in the reference Manual for possible new configuration statements which may give you improved functionality. Check your configuration file carefully as configuration files from older versions may need amendment (check particularly on the current meaning of the arguments used with the BBS interface statements and the ModemTrans statement as these have changed since early versions of BinkleyTerm).

4) As an "old hand" you might think of using your old language file (BINKLEY.LNG). Do NOT do this as it will probably lead to an error message along the lines of "Count of 368 from file does not match 420 required".
Either use the new BINKLEY.LNG included in the 2.60 archive or update and recompile your old text language file. Details of how to do this are in Reference Manual under "modification of internal text"

At this point, if you only want to run BinkleyTerm in Terminal Mode then your installation is complete and you can invoke BinkleyTerm (enter 'BT' or 'BT32', as appropriate, on the command line). Once running, Press Alt-F10 for a brief help screen.

The various keystrokes available in Terminal Mode are covered in detail in the Reference Manual.

If you intend to run BinkleyTerm as a mailer (Unattended Mode) then proceed with the installation as follows:

5) You NEED an Event file. Assuming you are a new user, change the name of the NEWUSER.EVT file included in the archive to BINKLEY.EVT and edit it as required. You will only need to use a very simple one line file if you are a point. If you are a node (see glossary if that does not mean anything) then change the event file to reflect the correct Zone Mail Hour for your zone.

When BinkleyTerm starts to run it will normally compile two more files it needs, BINKLEY.SCD and BINKLEY.DAY, using the information you have provided in BINKLEY.EVT.

Note: Whenever you change your event file you will need to delete these two files while BinkleyTerm is not running, and let BinkleyTerm re-create them when it restarts.

There is further information on event files in the reference Manual but you should not need that until you get a basic system running.

6) Obtain the necessary utilities and programs for packing and unpacking mail. Install them in accordance with the instructions included with each package.

BinkleyTerm is usually run from a batch file which has provision to exit and run these utilities for purposes such as maintenance, handling mail and so on.

7) If you are a new user take a look at the included very simple sample batch file called NEWUSER.BAT. It is written to use SQUISH as a mail processor and TIMED as the mail reader (you might want to alter this if you prefer other programs). Refer to the section "Control of BinkleyTerm Operation" for more information.

8) If you're a fully qualified FidoNet node obtain a current nodelist. Process it into usable form. using a nodelist compiler such as Xlaxnode, Qnode or Fastlst. Points may use a Nodelist but it is not essential.

10) If BinkleyTerm is to act as the "front end" for an existing Bulletin Board system (BBS) then make the appropriate changes to the batch file used to start your BBS. If it’s a new BBS then the software will give instructions for building a suitable batch file.

If the installation is completely new try the very limited batch file included in the archive as NUSpawn.BAT, editing this as necessary. (It is at present written to invoke the MAXIMUS BBS program). The filename "NUSpawn.bat" must be changed to SPAWNBBS.BAT before use.

The section on "Control of BinkleyTerm operation" later in this Manual is a useful guide to batch file usage and control by errorlevels.

That completes the installation. How you start your system will depend on how your batch files are written. If you have used NEWUSER.BAT then simply enter NEWUSER. (You can rename this file to BINK.BAT or whatever you fancy).

Alt-F10 displays a brief help file of commands available in Unattended Mode. Alt-Z zooms the Outbound display window to full screen and offers the opportunity to amend various aspects of any impending mail transfers.

The Reference Manual contains a full listing of the commands available from the Unattended Mode screen and the Zoomed Outbound window.

Please be clear that the "New User" sample files are intended as only a very basic starter for those who are new to BinkleyTerm.

It is advisable, with a program such as BinkleyTerm which has many options and optional features, to start with a very basic setup and, once that is working satisfactorily, add other features one by one.

In particular do not try to set up for Fax until the basic mailer is working well (Fax init strings are complex and varied and do not always show clear error messages when you get them wrong).

NOTES FOR DOS-BASED APPLICATIONS:

A FOSSIL driver is necessary to run BinkleyTerm.
A VIDEO FOSSIL (VFOSSIL) is needed with a color screen but optional with mono, though some features of the mono display are clearer and display more quickly with a VFOSSIL.

Drivers in common use at the time these notes were written are the X00 or BNU fossils and the VFOS_IBM video fossil. A newer Vfossil is VFOS_50 which runs a 50 line screen mode under VGA (VFOS_IBM can also do 50 lines if you set the screen mode before calling the Vfossil). Install the drivers you choose in accordance with the directions included with the driver.

NOTES FOR WINDOWS FOR WORKGROUPS

This was submitted by a user in the Binkley Echo and is offered "as is":

The same batch file can be used to run under WFWG as for DOS.
The settings in WFWG are:

Standard:
    Video Memory: Text
    Memory Requirements: KB Required: 128 KB Desired: 640
    EMS/XMS Memory: Set all fields to zero
    Display Usage: Full Screen
    Execution: Background
Advanced:
    Foreground/Background priority: 3000/8500
    (worked best for the users system, your mileage may vary)
Memory Options:
    Uses High Memory Area, Lock Application Memory
Display Options:
    Emulate Text Mode

NOTES FOR AN OS/2 INSTALLATION

Many users are now running the OS/2 version of BinkleyTerm because OS/2 offers superior and stable multi-tasking, thus enabling BinkleyTerm to be kept on-line whilst other tasks can operate at the same time to read mail etc., etc..

The installation of an OS/2 version of BinkleyTerm is straightforward.

It is best to start by obtaining SIO, Ray Gwinn's much acclaimed Shareware replacement for OS/2's own COMM drivers. This program has its own simple install program which replaces the internal comm drivers with SIO.SYS VSIO.SYS and VXOO.SYS.

Next obtain OS/2 versions of BinkleyTerm and the various ancillary programs you wish to use. Without wishing to decry any other programs, Squish, with Fastlst or Qnode to compile the nodelist, and Timed or GoldED to read and answer the mail, are known to work together whilst Maximus dovetails neatly as the BBS of choice with these programs. Don't forget that you will also need the

OS/2 versions of ZIP, UNZIP, ARC and LHA. (an aside here, if you get the InfoZip versions of ZIP and UnZIP be aware that they use pathnames as a default and you will probably need to make the unzip parameters in compress cfg read UNZIP -jo %a %f).

If you have previously used a DOS version of BinkleyTerm then it is likely that all you will need to do to run in basic form is to amend your old configuration to use the BBS SPAWN method of calling the BBS (if you run one), because it is necessary to use that method to avoid OS/2 dropping DTR when the BBS is called.

The configuration will not be optimum, but should work. Look up the AfterMail and ErrLevelShell statements as these can be used to good effect.

You will need to rename the BINK.BAT file used with DOS to BINK.CMD, and delete all references to fossils and vfossils. Rename all other batch files to .CMD files. Once it is all running you can delve into the mysteries of REXX which is a great improvement on batch files, making far more options available.

There are many ways to implement multi-tasking such as using ErrLevelShell to start a separate process as detailed in the description of that statement. Special programs, such as Jim Dailey's IMM (Inbound Mail Monitor) have been written to take advantage of the features of OS/2.In the next version of these docs we hope to cover OS/2 more fully.

NOTES FOR A WIN32 INSTALLATION

Some users have installed the Win32 version of BinkleyTerm on their Windows 95 and Windows NT systems for essentially the same benefits as the OS/2 version of BinkleyTerm: superior and stable multi-tasking, thus enabling BinkleyTerm to be kept on-line whilst other tasks can operate at the same time to read mail etc., etc..

The installation of a Win32 version of BinkleyTerm is straightforward.

Obtain Win32 versions of BinkleyTerm and either DOS or Win32 versions of the various ancillary programs you wish to use. Without wishing to decry any other programs, Squish, with Fastlst or Qnode to compile the nodelist, and Timed or GoldED to read and answer the mail, are known to work together whilst Maximus dovetails neatly as the BBS of choice with these programs. Don't forget that you will also need the Win32 or DOS versions of ZIP, UNZIP, ARC and LHA.
If you have previously used a DOS version of BinkleyTerm then it is likely that all you will need to do to run in basic form is to amend your old configuration to use the BBS SPAWN method of calling the BBS (if you run one), because it is necessary to use that method to avoid having the OS drop DTR when the BBS is called. You should also delete all references to fossils and vfossils.

The configuration will not be optimum, but should work. Look up the AfterMail and ErrLevelShell statements as these can be used to good effect.

Note that on Windows 95, you can’t get a Win32 BBS to run under BinkleyTerm with the standard command processor because comm port inheritance (the ability of one Win32 application to pass a “hot” port to another Win32 application) doesn’t work if there’s a DOS app (like the command processor) in the way. One way to use the Win32 BinkleyTerm to work on Windows 95 with a BBS is to install WinFOSSIL (a shareware driverwhich provides improved comm support), and use a DOS BBS.

There are many ways to implement multi-tasking such as using ErrLevelShell to start a separate process as detailed in the description of that statement.

NO REGISTRATION REQUIRED

Although BinkleyTerm is Copyrighted, it is "Freely Available Software" and as a non-commercial user you are not expected or required to register it, nor to pay for the privilege of using it.

While a registration is only required for extreme commercial use (see LICENSE.260 in the distribution archive), the program declares to other systems that it is "UNREGISTERED", which is quite true if you think about it, but if you prefer this message not to be displayed you can do one of the following:

1. Use the "Serial" configuration statement and pick a serial number of your own choice. Note: Do not include anything except numerals in the number.

or,

2. Edit the file ENGLISH.TXT that is in the 2.60 archive to display any alternative wording that appeals to you. After you do this you need to recompile ENGLISH.TXT to BINKLEY.LNG as described under "Modification..." in the Reference Manual.

OPERATION AS A TERMINAL COMMUNICATIONS PROGRAM

TERMINAL MODE OVERVIEW

BinkleyTerm's Terminal Mode offers functionality similar to that provided by programs such as Telix, ProComm or Qmodem. You can use your computer and modem to call out to on-line services and electronic bulletin board systems (BBS).

BinkleyTerm offers a full selection of file transfer protocols for use in exchanging files with remote systems. Also offered is optional VT-100 terminal emulation, an open architecture for adding additional file transfer protocols, support for high speed modems up to 115,200 baud, session logging to a flat file or printer, and more.

BinkleyTerm also features Zmodem Auto-Downloads. Once the transfer begins at the remote host, BinkleyTerm will detect the unique Zmodem start-of-transfer signal, and will immediately go into Zmodem download mode. If for some reason the Auto-Download does not occur, a Zmodem download can always be initiated manually.

VT-100 EMULATION

VT-100 terminal emulation is provided optionally by BinkleyTerm. Outward keystrokes are always active with the VT-100 keypad mapping. However, incoming screen control codes require ANSI X3.64 support, such as that provided in firmware on a DEC Rainbow computer.

On DOS only, for users of IBM PC computers and compatibles, external ANSI support is required, as provided for DOS users by FANSI-CONSOLE, a separately available utility. Load FANSI-CONSOLE in accordance with its directions prior to running BinkleyTerm for full VT-100 terminal emulation. OS/2 users enjoy support for the IBM variant of ANSI built into the operating system; while this is not a proper subset of ANSI X3.64, this level of support is exactly what most BBS systems expect when you identify your terminal emulation as ANSI.

On Win32 and OS/2 systems, you need to load an appropriate ANSI driver. ANSI.SYS for OS/2 is barely sufficient. We’ve found no appropriate driver on Win32 systems thus far.

Shown overleaf is an illustration of the VT-100 keypad, and the keystrokes that are required in order to emulate the respective key. The keypad layout and functionality should be familiar to VT-100 users.

```
+--------+--------+--------+--------+
|        |        | Shift- | Shift- |
|   F1   |   F2   |   F1   |   F2   |
|        |        |        |        |
+--------+--------+--------+--------+
|        |        | Shift- | Shift- |
|   F3   |   F4   |   F3   |   F4   |
|        |        |        |        |
+--------+--------+--------+--------+
|        |        | Shift- | Shift- |
|   F5   |   F6   |   F5   |   F6   |
|        |        |        |        |
+--------+--------+--------+--------+
|        |        | Shift- |        |
|   F7   |   F8   |   F7   |        |
|        |        |        | Shift- |
+--------+--------+--------+   F10  |
|                 | Shift- |        |
|      F10        |   F9   |        |
|                 |        |        |
+-----------------+--------+--------+
```

Note that BinkleyTerm also allows the use of the arrow cursor control keys on the keyboard. When used with a host that supports VT-100 or ANSI, the arrow keys are functional for cursor positioning. One of the more common non-VT-100 applications for these keys is with Opus-CBCS' "OPed" full-screen on-line message editor, and many of the full-screen external on-line editors used with QuickBBS.

OPERATION AS AN AUTOMATED ELECTRONIC MAILER

UNATTENDED MODE OVERVIEW

This mode is used when BinkleyTerm is to function as a front-end mail interface, whether in a BBS or "Point" environment.

The Glossary provides information on BBS and Point systems.
BinkleyTerm, in this operational mode, provides functionality similar to that provided by programs such as SEAmail, Dutchie, InterMail, FrontDoor, D'Bridge and other FidoNet compatible mailers.

BinkleyTerm answers the telephone, and exchanges mail with other compatible mail interface packages, or in the case of a human caller, passes control of the connection to a BBS system.

THE BINKLEYTERM CONCEPT

It is important to remember that BinkleyTerm is a mailer program with a completely open architecture. As such, BinkleyTerm can be operated with a tremendous variety of BBS systems, mail packing and unpacking programs, and accessories.

BinkleyTerm will answer the telephone and respond to another mail program. Mail will be placed in an incoming files area.

Incoming faxes will be stored into a separate files area set aside for fax reception. If the caller is human, control is passed entirely to the BBS program, if one is installed.

On the outward side, BinkleyTerm uses an outbound holding area to hold outward mail. It is the responsibility of third-party packing software to pack new messages into the required format, and place them in the outbound area.

BinkleyTerm neither packs nor unpacks mail. This responsibility is reserved for third-party software. Allowing this functionality to be provided by other software allows BinkleyTerm to be compatible with practically any BBS software, regardless of message base structure.

For mail handling, you might use programs such as QM, Squish, or ConfMail and oMMM to process mail, along with programs such as Sirius (nostalgia for you! but no longer being developed), Timed, Msged or Msgedsq to read and enter messages, or allow your users to do so in a BBS installation such as Fido, Maximus, or Opus-CBCS. Some BBS packages such as TBBS and QuickBBS come complete with their own proprietary packing and unpacking software. Finding the utilities and programs needed to work with your system is your responsibility.

HOW BINKLEYTERM HANDLES MAIL

Sending outbound mail with BinkleyTerm is fairly simple. The concept of mail handling with BinkleyTerm is the same concept that Opus-CBCS uses. If you're already familiar with Opus-CBCS, then BinkleyTerm will fall into place easily. If you are a complete neophyte, this section is intended to give you an understanding of the concept.

Idea #1: Cost is a prime consideration

Mail events are less important than they are with other mailing methods and systems. With BinkleyTerm's events, you paint with a wide brush telling the system what to do with 'classes' of remote systems.

When systems handled mail only at specific times, routing and times were of great importance. Because nearly all FidoNet technology systems can now process mail at any time, the idea of routing mail to particular systems on a scheduled basis becomes less important. The item of prime importance with BinkleyTerm is COST. We are going to try and relieve you of the tedious details of scheduling, and concentrate on doing things for the least cost.

Cost is, of course, determined by the nodelist entry. With a properly compiled nodelist, the entries have cost fields that realistically reflect the actual cost of sending mail to a particular node.

The 'L' flag is used when scheduling events to call out when costs are lowest. In most areas, it is cheapest to send toll calls during night-time hours. Therefore, BinkleyTerm is normally set-up to send such mail only during nighttime hours.

More about this is in the Reference Manual section, "Scheduling Events."

Idea #2: Create the packets once

Another new idea deals with the way that packets are created.
If you have used other mailer systems, you're probably used to seeing packets generated several times. With some programs, packets are built every time a mail schedule starts. As a result, one message may be put into a packet several times.

With BinkleyTerm, packets are built once by an external mail packing program. They may be remarked and rerouted once they're built, but they are physically built only once, and placed in a special sub- directory called the outbound holding area.
The original external packing program was called oMMM. oMMM stands for Opus Matrix Message Masher, and was originally designed for use with the Opus-CBCS system as its packer. BinkleyTerm is able to use oMMM because it chose to implement the holding area in a compatible (although not identical) way.

Most systems do not use oMMM any more; many mail packing programs for your particular environment can handle oMMM-like functions internally. These programs will often refer to this functionality as "oMMM-", "Opus-" or "BinkleyTerm-Compatible".

Idea #3: Continuous Mail

If you are already using a program that supports 'Crash Mail' then you understand a little of what Continuous Mail does.

Continuous Mail differs from Crash Mail in a couple of areas.
In other mailer systems, you would mark a message as 'Crash' meaning that you wanted this message to go out NOW. These mailer programs would shut down human caller access to the BBS and try like heck to get the message through.

BinkleyTerm uses Continuous Mail, meaning that this is a message going to a system that can accept mail at any time of day. BinkleyTerm makes no frantic attempts to dial out, rather it will try and deliver the message between callers.

Idea #4: Use File names to control traffic

The driving forces of outbound traffic are file names!
You'll have a special sub-directory set aside just for packets, compressed mail packets and other network files. This sub-directory belongs to BinkleyTerm, which will maintain the directory for you. It's a good idea not to play with this area unless you know exactly what you're doing.

Note also that when zoned operation is active (BinkleyTerm default) there are separate outbound areas for each zone. The default outbound area (for your zone) and one additional area for each other zone you deal with. The names of these additional areas are simply the outbound area name, with a three-digit extension that is the zone number in hexadecimal with leading zeroes. See "ZONE SUPPORT" on page 26.

The file names of the packets tell BinkleyTerm how to treat the different packets. Here's a typical packet name:
00680024.OUT

That says that the packet is for 0068/0024 (in hexadecimal) or 104/36 in more familiar terms. The ".OUT" means it is a Normal packet.

Other packet extensions include:

.HUT	Hold this packet for pickup by the remote system.
.CUT	The other system can receive Continuous Mail.

.DUT	Direct, meaning the other system can NOT receive Continuous Mail.

One nice thing is that you can manually change the file extension if you need to, or you can use fancy utilities such as AMAX or BONK to do this sort of thing for you on your command.

For the remainder of this section, we'll assume that you'll be using oMMM as your mail packer. As mentioned previously, you probably will be using another program that has oMMM-like functionality; it depends on your environment.

The oMMM program knows about these extensions and creates them based on information you put into the oMMM control file. You'll have statements like this:

```
NormHold 124/102
```

Any messages you enter to 124/102 would be turned into a .HUT packet file, placed into the outbound area, and BinkleyTerm would hold that packet for 124/102 to call and pick it up.
Files are also sent through FidoNet compatible networks. oMMM builds and maintains a file that tells BinkleyTerm what files to send (or hold) for whom. A typical 'file attach' file might be named:

```
00680024.FLO
```

This would designate a that there is a file waiting to be sent to 0068/0024 (in hexadecimal) or 104/36 in more familiar terms. The ".FLO" says it is a Normal file attach. File attach files are also called 'flow files' - named after the .FLO file extension.

Other flow file extensions are:

.HLO	Hold these files for pickup by the remote system.
.CLO	The other system can receive Continuous Mail.
.DLO	Direct, meaning the other system can NOT receive Continuous Mail.

A flow file is just a text file. It contains a list of files that are to be sent to another system:

```
#c:\binkley\outbound\0000fc9c.mo1
^c:\myfiles\wizzle.doc
c:\pascal\notes.doc
```

The '#' prior to a flow file entry says to truncate the file to zero-length after successfully sending the file to the remote system. This is normally only employed when sending compressed mail (archived mail) to the remote. The '^' prior to a flow file entry says to delete the file after sending.

The oMMM program (or the packer of your choice) will put messages into archives for you. Details on how this is done can be found in the EchoMail processor/message packer documentation. The point is that these packers combine the functionality of "generating packets" with that of traditional standalone programs like ARCmail.

A Sample Message, Start to Finish

So here's a practical example. Say I enter a message to Rod Lamping at 104/610. I mark the message as KILL/SENT when I enter it. I also enter the message designating a file to attach to Rod, named C:\FILE\REQ\FOOBAR.ARC.

I then enter a message in an EchoMail conference. My conference host is Phil Kaiser at 104/904, for whom I hold my mail for pickup. Among other things, I have two lines in my oMMM control file:

```
NormCM 104/610
OneHold 104/904
```

'NormCM' tells oMMM to mark the message as Continuous Mail (since Rod runs a mailer 24 hours a day). 'OneHold' tells oMMM to archive the mail to 104/904, and mark it Hold-for-Pickup.

oMMM users should refer to the oMMM documentation for the full set of available oMMM control file statements.

First, my EchoMail utilities are run to turn EchoMail messages into Normal packets, and place them in the outbound area for processing by oMMM. Next, I execute oMMM. It first scans the NetMail message area (where I entered my message to Rod) and turns new messages there into Normal packets, and if there are files attached, it creates Normal flow files. oMMM's second step is to use its control file, and apply the statements in the file against the mail in the outbound area that is marked as Normal.

Since I have Rod's board listed as NormCM, oMMM renames the file extension of the Normal packet and flow file for Rod to .CUT and .CLO respectively, for Continuous Mail.

Since I have Phil's board listed as OneHold, first oMMM archives the packets to Phil, then creates a flow file with a file extension of .HLO for Hold- for-Pickup.

I would then have the following in my outbound area:

00680262.CUT	Message to Rod
00680262.CLO	Flow File to Rod
00680388.HLO	Flow File to Phil
0000FC9C.MO1	Archived Message to Phil

For more information on how oMMM or your processor/packer works, refer to its specific documentation.

CONTROL OF BINKLEYTERM OPERATION

In Unattended Mode, BinkleyTerm is normally used with batch files that control system operation. You'll have a batch file used to invoke BinkleyTerm and invoke system utilities such as mail processors.

OS/2 users can either use batch files (known for some reason as command (.CMD) files) or the much more powerful and complex REXX program language included with OS/2... or even a combination of both.

Since BinkleyTerm can be used in such a wide variety of situations, there is not a particular "correct" or "incorrect" way to configure BinkleyTerm. The remainder of this section explains some of the operational theory and methods behind batch processing as it applies to BinkleyTerm.

If you are adding BinkleyTerm to an existing FidoNet compatible BBS system, chances are excellent that you are already familiar with batch files and the various tricks that they must typically perform in a FidoNet environment.

If advanced batch processing is beyond your scope, it is highly recommended that you review the manuals you received with your operating system, or the numerous third-party system usage guides for information to get you started.

One of the fundamentals of using batch files with BinkleyTerm is the concept of the 'errorlevel.' This system environment variable is set to a given value when a program completes execution. A batch file can detect and act upon the value, branching to various parts of the batch file. This section provides many good tips and ideas for errorlevel and batch file processing. Also consult your operating system manual for additional information.

A practical application might be after BinkleyTerm receives mail. If BinkleyTerm is configured to do so, it exits to the batch file, setting the errorlevel to a predetermined value. The batch file detects the value, and branches to a section in the batch file where mail unpacking programs are invoked. The batch file would then be restarted, invoking BinkleyTerm to again wait for or send mail.

ERRORLEVELS AND BATCH FILES

Errorlevels present a difficult hurdle to the new user of BinkleyTerm, especially those with no experience operating a BBS or a Point. In addition, BinkleyTerm's great flexibility confuses some Sysops. Few things are "fixed" or "cast in concrete." This section attempts to draw together pertinent information from many sources relating to errorlevels and their use in crafting a batch file to operate a BBS or a Point.

This section looks at what errorlevels are, which errorlevels are returned by BinkleyTerm, distinguishes between those errorlevel values set by BinkleyTerm and by the user, shows how you use errorlevels in your batch file and, finally, offers some hints and shortcuts.

What is an "Errorlevel"?

An errorlevel is a numeric value which a program may "return" when it terminates. The name is misleading because the value returned is really an exit code, rather than an indication of an error per se. It distinguishes different types of normal exits, in addition to denoting an abnormal condition on exit.

The software's creator (or, in some instances, the program's user) sets the value(s).

Errorlevels permit the user to craft a batch (.BAT or .CMD) file, using system batch commands such as IF and GOTO, to logically organize the BBS or Point operating process. Automagically, this batch file determines which tasks need be performed and in what order. It branches between tasks based on the exit code generated by each program and on the program logic created by the Sysop.

For example, an editor might return the following:

Exit Code	Condition

---------	--------------
3	EchoMail and NetMail Created
2	EchoMail Only Created
1	NetMail Only Created
0	No Mail Created

Branching or jumping within the batch files is facilitated with the use of labels. A label is used to identify a particular block within the batch file. For example, you might choose to label a portion of your batch file "nodelist" because that particular section of the file handles nodelist processing. Labels are designated with a colon, followed by the label name.

You may wish to identify the top of the file with a label to make re-starting the batch file as easy as using a GOTO statement.
Here is a short example of the use of labels:

```
:start
MYPROG -p1
if errorlevel 2 goto process
if errorlevel 1 goto start
if errorlevel 0 goto end
:process
PROC -c -d1
if errorlevel 1 goto start
if errorlevel 0 goto end
:end
```

Note that the top of the file is identified by a label named "start" and that other sections are labeled as well. The batch file starts out by invoking a program called MYPROG. Once MYPROG exits, the errorlevel values determine where in the file that control will be passed. One errorlevel references another label that in turn will cause a program named PROC to be run. Another branch goes back to the top of the file, and yet another goes to the end of the file. The PROC program also gives errorlevels that are processed by the batch file.

Errorlevels and associated batch processing commands permit tailoring the batch file and bypassing of unnecessary processing steps, saving substantial amounts of time, which, in turn, translates into more time for user access and other processes. (There is NEVER too much time.)

When does BinkleyTerm Produce Errorlevels?

BinkleyTerm returns errorlevels in four general cases: Sysop-Initiated, Sysop-Defined, Caller-Initiated, System-Generated.

Sysop-Initiated errorlevels are generated by pressing a function key or Alt-X.

Sysop-Defined errorlevels are generated using the E1, E2 and E3 event scheduling exits, and exits associated with external-mail programs.

Caller-Initiated errorlevels are generated for calls that are not mail-related, such as external mail programs or BBS users (if BBS batch or BBS exit is chosen).

System-Generated errorlevels are caused by fatal compiler runtime errors, or fatal BinkleyTerm configuration errors, like not being able to find its address in the specified nodelist.

What errorlevels does BinkleyTerm return?

Many values are "hard coded" in BinkleyTerm or are provided in the sample files included in the distribution package. However, only some of the values have "fixed" definitions.

Exit	Meaning	Caused By
Code	----------------	--------------------
1	Exit	Alt-X Keypress
3	300 bps Call	Non-Mail Call at 300 Baud
10	E1= Exit {note 2}	F1 Keypress {note 1}
12	1200 bps Call	Non-Mail Call at 1200 Baud

Exit	Meaning	Caused By
Code	----------------	--------------------
20	E2= Exit {note 2}	F2 Keypress {note 1}
24	2400 bps Call	Non-Mail Call at 2400 Baud
30	E3= Exit {note 2}	F3 Keypress {note 1}
40	{note 2}	F4 Keypress
48	4800 bps Call	Non-Mail Call at 4800 Baud
50	{note 2}	F5 Keypress
60	{note 2}	F6 Keypress
64	57600-bps Call	Non-Mail call at 57600 Baud
70	{note 2}	F7 Keypress
80	{note 2}	F8 Keypress
90	{note 2}	F9 Keypress
96	9600 bps Call	Non-Mail Call at 9600 Baud
99	ExtrnMail	External Mail String Recvd {note 6}
100	{note 2}	F10 Keypress
128	38400 bps or 115,200 bps Call	Non-Mail Call at 38,400 Baud or 115,200 Baud {note 4}
192	19200 bps Call	Non-Mail Call at 19,200 Baud
254	Error	Address Not Found in Nodelist {note 5}
255	Error	Compiler runtime Error

Notes:

1. In the event file it is usual, but not essential, to set errorlevels 10, 20 and 30 for E1=, E2= and E3= exits respectively. When this is done, pressing F1, F2 or F3 has the same effect as BinkleyTerm making an E1=, E2= or E3= exit.

Exits are defined as:

E1= Beginning of Event
E2= Mail Received
E3= Compressed Mail Received
E4-E9= Exit after event when preset file type received
EF= FAX received

2. All function-key exits have user-defined functionality.

3. On receiving a non-mail call (i.e., a BBS call) BinkleyTerm exits with an errorlevel equal to the modem speed in bps divided by 100.

DOS limitations are such that DOS cannot handle errorlevels greater than 255 (they start from 0 so there are 256 errorlevels). Values returned by BinkleyTerm over 255 (such as from a 28.800 bps call) "wrap" round to a lower value.
Basically, to find the errorlevel which DOS will see, divide the modem speed in bps by 100 and if the answer is greater than 255 keep subtracting 256 until the number is less than 256.

i.e.	Errorlevels
----------	----------------
0-255	no change
256-511	subtract 256
512-767	subtract 512
115200 bps	see note 4.

Therefore, the value of a 38,400 baud exit "wraps" around to equal 128, instead of 384. Unfortunately, the value of a 115,200 baud exit also wraps to the same value. It is recommended that users in this scenario use the "BBS spawn", "BBS batch" or "Extern Spawn" facilities and not use errorlevels to communicate baud rates.

5. BinkleyTerm will exit with errorlevel 254 when it cannot find its own address in the nodelist. BinkleyTerm also checks certain other parameters designated in the configuration file viz.,

. Non-Zero Zone Number Designated
. Non-Zero Net Number Designated
. System Name Designated
. Sysop Name Designated

. Outbound (Hold) Area Designated
. Inbound Area Designated

If any of these parameters are not designated or are improperly designated , then an errorlevel 254 exit is performed.

6. The external mail program functionality features configurable exit values, but the default value is 99.

Making Errorlevels Work For You

You, the user, define ALL errorlevels associated with the E1=, E2= , E3= and EF= exits, setting them in BINKLEY.EVT. You 'program' a specific task to occur at the beginning of a selected BinkleyTerm "event" (time interval) by selecting a unique "E1=" code for that task.

The task can be set to occur periodically throughout each day, once-per-day, once-a-week or on selected days within the week; further, you may execute a given task at different times on different days, if so desired, depending on the way a particular event is configured.

Similarly, by having a variety of E2= , E3= and EF= codes, you can use different mail unpacking routines and FAX postprocessing agents throughout the day or week. Refer to the section "Scheduling Events" in the Reference Manual for details.

You customize the operation of your BBS or Point to meet your specific needs and those of your "customers".

Errorlevels, Batch Files and ExtrnMail

This BinkleyTerm option is intended for invocation of an external mail handling program, possibly an alternate mailer, upon reception of a user-defined string of characters. It can also be used to set-up multiple BBS installations, allowing a user to specify which BBS he or she wants. Also, an external FAX reception program can be invoked using this option in combination with definition of FAX result codes. This is covered in detail in the sections "External Mail Programs" and "Fax" in the Reference Manual.

Errorlevels, Batch Files and Housekeeping

Another common application of exits and exit values is for housekeeping. For example, a Point system could be set up:

1) To do daily housekeeping (deleting and renumbering of the message base) daily at 8:00 AM.

2) To Poll his bossnode twice a day, at 4:00 AM and 11:00 PM,.

3) To request the current NODEDIFF from the bossnode each Saturday and then, on Sunday, process the weekly NODELIST.
A distinct E1= errorlevel handles each task.

It is up to you to prevent chaos and create an orderly functioning batch file for your BBS or Point. Be vigilant, and avoid re-using the same errorlevel for several different tasks.

Using Errorlevels

How do you use errorlevels? By testing for their existence with the batch file statement:

IF ERRORLEVEL n GOTO xxx
Where:
n is a numeric value
xxx is a label within the batch file

Bear in mind the following Rules:

The "IF ERRORLEVEL ... " statements must follow immediately after the program they test.

For most command shells, "IF ERRORLEVEL n" returns true for any value GREATER THAN or EQUAL TO "n", therefore, Test errorlevels in descending numerical sequence, and,

Trap unwanted exit codes so that a step will be executed if, and only if, the desired errorlevel is encountered.

In this excerpt from a batch file, the "Start_BT" label refers to a section that loads BinkleyTerm:

```
IF ERRORLEVEL 21 GOTO Start_BT        <- Traps any errorlevel greater than 20 {note 1}
IF ERRORLEVEL 20 GOTO OpenMail        <- Branch & Open Mail
IF ERRORLEVEL 11 GOTO Start_BT        <- Traps 11 to 19 {note 1}
IF ERRORLEVEL 10 GOTO Start_BT        <- "Non-Event"
IF ERRORLEVEL 3 GOTO Start_BT         <- Traps 3 to 9 {note 1}
IF ERRORLEVEL 2 GOTO Bad_Exit         <- Nodelist Entry Error
IF ERRORLEVEL 1 GOTO Exit_BT          <- Branch to 'Exit'
```

Note 1. Invalid responses are 'trapped', BinkleyTerm restarts.

Errorlevel and Batch File Hints and Kinks

When looking at sample batch files, note carefully how each author organized the program logic in his batch file. Here are some reminders.

Under DOS and derivative command processors, a batch file loses 'control' if it merely invokes another batch file (e.g. you execute a second batch file within the first); instead, "shell out" to a second batch file from the first batch file by invoking another copy of the command processor. Using the /C switch closes that second command processor when the second batch file finishes. Consult your operating system reference materials for more information on this point.
Examples:
(for DOS) COMMAND.COM /C Pak_Mail.BAT
or:
(for any system) %comspec% /C Pak_Mail.BAT

This is true for all MS-DOS versions up to and including 3.2, and all versions of OS/2 and Win32 (although in the case of OS/2 it is better to give batch files the extension .CMD).

MS-DOS/PC-DOS 3.3 introduced the batch command CALL which overcomes this limitation, when it is used. Refer to your DOS manual. All versions of OS/2 and Win32 platforms support the CALL command.

Sometimes, your batch file must "remember" what it is doing. You can accomplish this by "setting" an environment variable, for example:

SET PROC=UNATTENDED
Your batch file can then determine its "logic path" or invoke other programs using statements such as:

IF %PROC% == UNATTENDED GOTO xxx
or:
IF %PROC% == UNATTENDED BT UNATTENDED
or:
IF %PROC% == UNATTENDED %comspec% /C Pak_Mail.BAT

In the above examples, the program follows a certain logic path and performs specific tasks when it's in the unattended mode.
Verify that you have sufficient environment space for any variables which you want to "store" in it. Review the section concerning the SHELL entry for CONFIG.SYS in your DOS or OS/2 reference material. This will explain how to enlarge the environment space if that is needed. The exact adjustments which you may need to make are dependent upon your version of MS-DOS/PC-DOS, Win32 system or OS/2, and several other factors such as memory use, size of paths and so on.

You can also pass a variable into a batch file when you invoke a program. For example, the author's POINT.BAT batch file immediately comes up in the Unattended Mode if you invoke it with:

```
POINT -u (or, POINT -U)
```

POINT.BAT then leapfrogs directly into unattended mode using the following statements:

```
IF %1 == -u SET PROC=UNATTENDED
IF %1 == -u GOTO Start_BT
:Start_BT
IF %PROC% == TERM BT      <-Terminal Mode
IF NOT %PROC% == TERM BT %PROC% <-Other Modes
```

Finally, use of errorlevels is not always necessary. Some programs provide an alternative which may meet your needs.

For example, ConfMail and some FidoNet message editors produce a ".OUT" file containing the name of each EchoMail area for which mail has been received or created. If this is true of your processor, you can test for the existence of the .OUT file rather than testing the errorlevels and, thereby, simplify the logical branching in your batch (.BAT) file. For example:

```
if exist AREAS.OUT ConfMail Export .-F AREAS.OUT
if exist AREAS.OUT del AREAS.OUT
```

or:

```
if exist ECHO.OUT ConfMail Maint .-F ECHO.OUT
if exist ECHO.OUT ReplyLnk -F ECHO.OUT
if exist ECHO.OUT del ECHO.OUT
```

The point here is that such an .OUT file can be used by several programs serially, as in the last example, and thus, avoid using errorlevel driven logic. Further, it can continue working for you after you have "lost" your errorlevel by running another program.

THE BINKLEY ENVIRONMENT VARIABLE

When running under DOS in environments where multiple drives and paths are used, it is generally a good idea to set an environment variable named BINKLEY. BinkleyTerm will then use this variable to determine where its configuration file is located, should it not be located in the current directory. Place in your AUTOEXEC.BAT file (or in the batch file that starts BinkleyTerm) the line:

```
SET BINKLEY=C:\PATH
```

Where C:\PATH is the drive designator and complete path name where the configuration file is located.

COMMAND SHELL KEYS

There are also 9 programmable Command shell exits which are invoked by using Alt/Function key combinations. You enable and set these up in the configuration file. Refer to the "Shell" statement in the Configuration File section of the Reference Manual.

These shells produce NO errorlevels but are mentioned because they are an alternate and/or additional solution to creating Sysop controllable exit/task options. When configured, these shell keys invoke a new copy of the system command processor. You may not want to use these as they consume extra memory for the additional copy of the command processor.

NODELIST

The listing of FidoNet compatible systems in your network is called a 'nodelist.' Once a current nodelist is obtained, it can be kept up-to-date by using NODEDIFF files that are distributed weekly within the network.

The nodelist and update files are distributed in 'raw' form. Adjunct software must be used to process and compile the raw nodelist and NODEDIFF files into a form usable by BinkleyTerm.

From now on, when we refer to "nodelist" we're referring to the compiled, ready-to-use nodelist data files, not the raw nodelist file as distributed within the network.

Point installations do not need a nodelist at all if they don't care to have one, simply by using all of the following configuration file statements:

. BossPhone
. BossPwd
. Address

Of course, this requires establishing a session password with your boss (as designated by the 'BossPwd' statement). With this method, Terminal Mode must be used to poll the boss using the Alt-Y keystroke, as Unattended Mode will not operate without a nodelist available.

Nodelist Formats

BinkleyTerm is capable of using a variety of compiled nodelist formats but due to the size of the current nodelists some earlier formats cannot compile a full nodelist. Use of the Version 7 nodelist format is recommended.

a) Version 5 Nodelist and QuickBBS Nodelists
Version 5 and QuickBBS Nodelists are now obsolete and are not supported in the .EXE files. They are available by an option in the source.

b) TBBS Nodelist
The TBBS nodelist format can be used but does not provide Zone support (Refer to the BinkleyTerm Version 2.30 documentation for usage details)

c) Version 6 Nodelist
This type of Nodelist can be used (refer to the BinkleyTerm Version 2.30 documentation for usage details) though the size of the current compiled nodelist may cause difficulties. The later Version 7 offers a large saving in memory usage.

d) Version 7 Nodelist
This version offers a 40% saving in file size over Version 6. Be sure to use a Nodelist compiler which can handle Version 7 nodelists.

Notes:

1. Check on the names of the files produced by your compiler and give BinkleyTerm the correct names.
   Version7 names differ from earlier versions
2. The size of the current nodelist causes older versions of some nodelist processors to fail.

ZONE SUPPORT

Zones are a high-level addressing scheme devised for use within FidoNet. In a full FidoNet address, such a 1:104/36.0, '1' would be the zone, '104' the net, '36' the node number, and '0' the point number. Currently, zone addressing is not supported within the FidoNet message or packet structure, allowing software such as BinkleyTerm to provide only "kludged" support of zones. BinkleyTerm offers such support, and endeavors to make it as seamless as possible.

BinkleyTerm will assume that a full, zone-based nodelist is available for its use.

If for some reason you do not wish zone support to be active, place the statement 'NoZones' in your configuration file.
For information on how to properly compile a nodelist for BinkleyTerm, refer to the section "Nodelist".

When attempting to send mail to nodes in other zones, BinkleyTerm will assume that mail for other zones will be held in separate outbound areas by zone number. For example, if you are in Zone 1, and your outbound mail directory is C:\BT\OUTBOUND, mail for your zone will be held there. However, mail for nodes in zone 2 would be expected in C:\BT\OUTBOUND.002, mail for zone 3 would be expected in C:\BT\OUTBOUND.003, and so on.

The zone number for which your default outbound directory applies is determined by the FIRST appearance of the 'Address' statement in your configuration file. Subsequent 'Address' statements identify your alternate identities within other zones (and/or networks). For example, if the first 'Address' statement designates an address in zone 2, then the outbound area designated by the 'Hold' statement in your configuration file is the default, and mail to other zones would require their own distinct outbound areas with extensions that match the zone number.

The multi-zone outbound areas are in hexadecimal. For example, the outbound area for zone 10 would be C:\BT\OUTBOUND\.00A.

Using this method, it is possible to support up to 4,095 zones.

Your mail packer also needs to support the discrete outbound areas for multiple zones. oMMM versions greater than 1.30 support them, as well as the MOOO package. oMMM is commonly available wherever BinkleyTerm or Opus-CBCS files are available. MOOO is also available from many such locations.

More advanced software such as QM and Squish can also handle these areas.

MULTIPLE NETWORK OPERATION

This can best be achieved by using "domain" addressing (see "DOMAIN SUPPORT" in following section).

An older method, still used, assumes that networks such as AlterNet, EggNet, RBBS-Net and so on are implemented as discrete, separate zones. To facilitate operation of a BinkleyTerm system within multiple networks, you may specify a separate system address, each with a different zone.

For example, if you wish to use a different address for FidoNet (currently zones 1, 2, 3 and 4) and for two alternative networks, you might have the following in your configuration file:

```
Address 1:1010/89.0
Address 9:569/999.0
Address 11:334/1.0
```

If your system connects with a system in zone 9, your system will identify itself as 9:569/999. If connected with a zone 11 system, it will identify itself as 11:334/1. The first 'Address' statement is the default, and callers in zone 1 (and any zones other than 1, 9 and 11 - those specified with 'Address' statements) would find your system identified as 1:1010/89.

DOMAIN SUPPORT

BinkleyTerm now features support for "domain" addressing. Due to the depth of the subject, domain addressing is covered completely in its own section in the Reference Manual.

DOMAIN, ZONE, NET ADDRESS ASSUMPTION

BinkleyTerm supports zone, net and domain assumption. If you have more than one address and different zones, nets, or domains are designated in the configuration file, then BinkleyTerm will use whichever address is appropriate when transacting mail.
For example, net assumption might work as follows. If I have two addresses - 104/36 (primary address) and 1052/1 (secondary address) - and someone calls from net 1052, then my system will identify itself at 1052/1. In all other cases, my system will identify itself as 104/36 (since it's my primary address). The purpose is that your system will identify itself as the most appropriate address depending on the address of the calling system.

SECURITY

In the ideal world, we would not need locks, police, or jails; there wouldn't be crime. But we don't live in an ideal world, and for this reason, BinkleyTerm offers a selection of features that are intended to offer your system a certain level of security against "electronic mail crime."

The existence of security features is not intended to evoke fear. Chances are excellent that you will have no need for security features in most cases. But just as high crime areas see more locks and iron gates than low-crime areas, the choice of how much security to put in place is up to you, and is based on your needs and experience.

Refer to the section on System Security in the Reference Manual for full details.

BBS INTERFACE

One of the most common uses of BinkleyTerm is as a mail front-end for a bulletin board system, or BBS.

For DOS users, BinkleyTerm offers three different methods for passing control to a BBS. The method used is determined by a configuration file statement.

For OS/2 and Win32 users, the BBS Spawn method should be used as the connection will be lost if the other two methods are used.
Refer to the BBS Interface section in the Reference Manual for full details.

EXTERNAL PROTOCOLS

BinkleyTerm provides a rich selection of file transfer protocols, integrated into the package for ease of use and efficient operation. On occasion however, other protocols may also be desired.

BinkleyTerm supports special external file transfer programs that conform to the standard used by Opus-CBCS. External file transfer protocols MUST identify themselves as being compatible with Opus-CBCS in order to work with BinkleyTerm. Modules implementing such protocols are often available from Opus-CBCS systems.

Protocols currently available include WXmodem, Kermit and others. To use such an external protocol, add a 'Protocol' statement to your configuration file along with the required path information.

Some of the Opus-CBCS compatible protocols are specially designed and optimized for BBS use, and may not be operable in BinkleyTerm's Terminal Mode. Check the individual package for information, or test the performance of the package on your own.

"True" external protocols, those that are designed to be accessed by any communications program via a Command Shell, can also be used with BinkleyTerm. BinkleyTerm provides a Command Shell command that can be invoked during a communications session. An external protocol of this type can then be executed from the DOS command line, or from a batch file, depending on your situation. External protocols such as Jmodem and BiModem can be accessed in this manner.

HIGH SPEED ERROR CORRECTING MODEMS

Many of the newer modems available on the market today offer an advanced feature that, used in combination with a special setting of BinkleyTerm's "LockBaud" configuration verb, will give your callers with high-speed error-correcting modems the speed benefits of a locked port and others will have the responsiveness of a normal connection.

The feature in question relates to the modem's ability to maintain a high-speed computer-to-modem communications rate on any error-correcting connection, and to float the communications rate for any non-error-correcting connection.

Here are commands on certain modems to enable this feature:

U.S. Robotics	AT&B2
Newer Hayes or Practical Peripherals	AT&Q6 and ATS36=5
Telebit modems such as the T2500	ATS66=5

NOTE: You can continue to use your current floating or locked port setup by leaving the "LockBaud" configuration verb commented out and ignoring the following.

Courier HST and HST Dual Standards manufactured prior to February 1989 (those not supporting the S-register 27 lock options) will not be able to utilize the new "LockBaud" options.

Here's how to install this new floating/locked setup:

1. DON'T lock your FOSSIL or other communications driver.

2. In BINKLEY.CFG, uncomment or enter "LockBaud /ARQ" for U.S. Robotics or Hayes modems, or "LockBaud /REL" for Telebit T2500. For other modems, consult your manual to determine the connect string suffix which denotes an error-free connection, and how to configure the modem to ensure that this string appears like "Connect dce-rate/errorfree-string. On Hayes, this is accomplished using ATS95=3.

3. If your BBS software allows you to pass the port rate separately (as with Maximus), call up the BBS as follows:

```
Max -b%2 -p%3 -t%4 -s%1
(where %1 is the port rate and %2 is the connect rate)
```

4. If your BBS doesn't allow passing the link (port) speed separately from the connect speed (as with Opus), you can use the following kludge in your SPAWNBBS or EXTMAIL batch file (using X00's XU.EXE or the similar utility included with your FOSSIL driver):

```
REM convert 1-based port from BinkleyTerm to 0-based for XU
If "%3" == "1" SET PORT=0
If "%3" == "2" SET PORT=1
REM it's always OK to lock with XU since unlock follows
XU LOCK:%PORT%:%1
Opus Bbs -b%2 -p%3 -t%4
if ERRORLEVEL . .
REM unlock the port
XU LOCK:%PORT%:OFF
```

Note that Opus uses a 1-based communications port number, but XU & X00 use a 0-based communications port number.

With versions of Opus subsequent to 1.72a, you can also pass the lock information to Opus, making it unnecessary to play FOSSIL locking games:

From a batch file you would call Opus as:

```
Opus BBS -b%1 -p%2 -t%3 -a%5
```

With this feature enabled, when the modem establishes a connection with another error-correcting modem (an /ARQ or /REL connect), it will shift its DTE rate (the speed it uses to talk to your computer) UP to the rate you stored in its non-volatile ram (NVRAM) when you initially set it up. To adjust this stored rate, set your favorite communications program (or BinkleyTerm's terminal mode) to the desired rate and send the modem an ATZ<enter> AT&W<enter>. The modem stores the bps rate of the command in its NVRAM. Each time it makes an /ARQ or /REL connection, it checks NVRAM for the specified DTE rate, and sets it accordingly. For non-/ARQ or non-/REL callers, it sets the DTE rate to the connect rate.

For example, here's a table showing how BinkleyTerm reacts to connects with and without use of the "LockBaud" verb:

No "LockBaud /ARQ" & "Baud 38400" (or 19200)

Modem Connect String	%1 (Link Rate)	%2 (Connect Rate)
CONNECT 14400/ARQ	19200	14400
CONNECT 12000/ARQ	19200	12000
CONNECT 9600	9600	9600
CONNECT 9600/ARQ	9600	9600
CONNECT 7200	9600	7200
CONNECT 7200/ARQ	9600	7200
CONNECT 4800	4800	4800
CONNECT 4800/ARQ	4800	4800
CONNECT 2400	2400	2400
CONNECT 2400/ARQ	2400	2400

Using "LockBaud /ARQ" & "Baud 38400"

Modem Connect String	%1 (Link Rate)	%2 (Connect Rate)
CONNECT 14400/ARQ	38400	14400
CONNECT 12000/ARQ	38400	12000
CONNECT 9600	9600	9600
CONNECT 9600/ARQ	38400	9600
CONNECT 7200	9600	7200
CONNECT 7200/ARQ	38400	7200
CONNECT 4800	4800	4800
CONNECT 4800/ARQ	38400	4800
CONNECT 2400	2400	2400
CONNECT 2400/ARQ	38400	2400

Using "LockBaud /ARQ" & "Baud 19200"

Modem Connect String	%1 (Link Rate)	%2 (Connect Rate)
CONNECT 14400/ARQ	19200	14400
CONNECT 12000/ARQ	19200	12000
CONNECT 9600	9600	9600
CONNECT 9600/ARQ	19200	9600
CONNECT 7200	9600	7200
CONNECT 7200/ARQ	19200	7200
CONNECT 4800	4800	4800
CONNECT 4800/ARQ	19200	4800
CONNECT 2400	2400	2400
CONNECT 2400/ARQ	19200	2400

PROBLEM SOLVING

BINKLEYTERM SUPPORT

Since BinkleyTerm is a product which is "free for the asking" you cannot expect a toll-free software support line (as much as we may want to provide you with one).

The primary means of support is the BINKLEY EchoMail Conference. This conference is carried worldwide via FidoNet. Contact your EchoMail Coordinator or Hub for information on hooking into the conference, or finding a system that you can use to participate in the conference.

The BINKLEY conference is monitored and read by the BinkleyTerm authors and beta testers.

A secondary source of support is provided by BinkleyTerm HELP, Bob Juge, at FidoNet Address 1:1/102.0. If you have an urgent question, or are unable to hook into the BINKLEY EchoMail Conference, send a NetMail message to BinkleyTerm HELP.

Replies are issued as time and resources allow, so please be patient.

TROUBLESHOOTING

3,000 pages of documentation would not entirely eliminate the potential for problems with the installation and operation of BinkleyTerm. Due to the wide variety of hardware and software configurations that BinkleyTerm may be used with, as well as the varying levels of experience of the BinkleyTerm user, problems will sometimes occur. This section attempts to present common problems and possible solutions.

If there is not a solution to your problem presented here, please read over the appropriate sections of the manual again. If you still are having difficulty, place a message in the BINKLEY EchoMail conference, or contact BinkleyTerm HELP at 1:1/102.

Common Queries and Answers

Error message "Count of xxx does not match yyy required"

You have probably used the BINKLEY.LNG file from a different version. Either use the version provided with the 2.60 archive or correct and recompile your old text language file. See "modification of internal text" in the Reference Manual

New installation and BinkleyTerm "does nothing"

You have probably not created the main inbound directory which you specified in the "NetFile" configuration statement.

New installation and BinkleyTerm reports "No Boss in Nodelist"

Until you are listed in the Nodelist you will need to either include the "Boss" information in your Binkley.cfg or edit your nodelist by adding yourself to it.
Of course, this message will also appear if you have not told BinkleyTerm the correct information about your nodelist.

Baud Rate Locking Trouble

Do not use BinkleyTerm's 'LockBaud' configuration file statement. Lock the baud rate of your FOSSIL instead, and configure your modem to always work with a locked DTE rate.
Note that the 'Autobaud' statement simply tells BinkleyTerm to dial out at the rate set by your 'Baud' statement, instead of using the baud rate in the nodelist. It has nothing to do with negotiating a connect speed with another modem.

Garbage received after changing locked baud rate of Fossil

When you initially set things up, or when someone calls in after you have changed the fossil fixed rate you may receive pages of rubbish. This can also happen if you have been in terminal mode and have reset the modem speed.

With many modern modems a default speed is set in NVRAM when an AT&W command is issued and this speed will be used by the modem when it is next reset (ie., each time Bink issues an ATZ, for example).

At the next call, if the Fossil is using a different baud rate to that last set in NVRAM then garbage will result.

The fix is to go into BinkleyTerm's Terminal mode (after locking the Fossil at the desired speed), and issue the following three modem commands:

```
ATZ
AT
AT&W
```

This should leave all of your modem settings unaltered EXCEPT for the port speed which should now be the same as your fossil rate.

Outward Dial Aborting

Many people installing BinkleyTerm mistakenly use the 'Suffix' option in their configuration file. Unlike most communications programs, BinkleyTerm by default adds a carriage return to the end of the dial string. 'Suffix' is used to add instructions to the end of the phone number, but BEFORE the carriage return.

By adding a carriage return code (pipe symbol, |) with the 'Suffix' statement, you are essentially telling BinkleyTerm to send TWO carriage returns, yours, plus the default. With most modems, this will immediately abort the dialing process before it even gets started.

In nearly all installations, the 'Suffix' statement should NOT be used (it should be omitted or be COMMENTED OUT). If deleting the 'Suffix' statement does not fix the problem, you may also try adding the 'NoCollide' and/or 'SlowModem' statements to the configuration file. Refer to the Reference Manual section "Configuration File" for more details.

False Call Collision Reports

In some installations, BinkleyTerm may abort the dialing process due to an incoming call, even when there is no incoming call. This is probably due to the modem reporting "RING" on both incoming and outgoing calls. Use the 'SameRing' configuration file option to partially disable the call collision feature; use the 'NoCollide' option to totally disable the feature.

DOS FOSSIL Driver Compatibility Problems

The most popular FOSSIL drivers in use with BinkleyTerm in the DOS environment are Ray Gwinn's X00, David Nugent's BNU, and Bob Hartman's OpusComm, all three of which are for the IBM PC and close compatibles. If one of the drivers fails to work correctly in your installation, please try another.

OS/2 and Win32 versions of BinkleyTerm do not require the use of a Fossil driver.

BinkleyTerm Will Not Recognize Node Addresses

You probably have not compiled the nodelist correctly. Use a nodelist compiler (such as Qnode or ParseLst) to compile a fully-zoned Version 7 nodelist for your system. To do this, make sure the following statements exist within your ParseLst configuration file: UseZone, Complete (or Gated), and Version7.

Also, check that you have the correct filenames in your BINKLEY.CFG file, Version 7 uses NODEX.DAT and NODEX.NDX, not NODELIST.DAT and NODELIST.IDX as in earlier versions.

TBBS Difficulty - BinkleyTerm Runtime Errors

When used with TBBS, BinkleyTerm must be renamed MAILER.EXE. TBBS users should be certain to use BTBIG.EXE, the non-overlay version, on their systems, renaming it to MAILER.EXE before use.

The other version of BinkleyTerm in the archive, named BT.EXE, uses overlays to provide more efficient memory usage and can not be renamed.

Some TBBS Sysops have patched TBBS.EXE to invoke BT.EXE, which allows them to use the overlaid version. While if this is done correctly it will probably cause no problems, we cannot recommend this course of action.

Zone Support Does Not Operate Correctly

Chances are excellent that you have not compiled the nodelist correctly. Although the actual entries for nodes in other zones do not need to be included in the compiled nodelist files, what are called "zone identifiers" DO need to be included in order for zone support to work. See "NODELIST" on page 25 for more information on correct compilation of a nodelist.

Another item to check is that outbound areas are created for the other zones to which you want to send mail. See "ZONE SUPPORT" on page 26 for information on how outbound areas for other zones are constructed.

Domain Support Does Not Operate Correctly

There are several potential causes for this: the "Domain" statement may have the wrong nodelist filename on it (for Version 7, the FidoNet domain line should look like "domain fidonet fidonet nodex"); you don't have all the necessary "Domain" statements, or the system you are calling is misconfigured.

Another possibility is that you are using DomainKludge statements but have placed them before the Domain statements. .. They must be placed *after* the Domain statements.

Date Rollover Problem

BinkleyTerm keeps schedule information in binary form in a file named BINKLEY.SCD. If for some reason the file is newer than the current date and time, BinkleyTerm will report "Date Rollover Problem?" as this condition usually indicates that DOS is not rolling the date over at midnight as it should. If there appears to be no difficulty with date rollover, delete the BINKLEY.SCD file, and allow BinkleyTerm to construct a new one the next time it is invoked. A date rollover problem sometimes does exist with certain versions of DOS. If the problem persists, a DOS upgrade may be indicated.

BinkleyTerm Does not do what Event file says

Chances are that the cause is one or more of the following:

1. You have not written the event file correctly. Check for a full set of events, with no gaps, starting from 00.00 and ending 23.59. If an event stops at 08.00 then the next event should start at 08.00, NOT 08.01 (which would leave a one minute gap). Check that the flags you have used allow the action you want to perform at this time.

2. In order for BinkleyTerm to look at a revised Event file, you must close BinkleyTerm down and then delete BINKLEY.SCD and BINKLEY.DAY. When you restart BinkleyTerm will use the event file to create new .SCD and .DAY files. If you use multitasking and BinkleyTerm is still running anywhere on your system the new information will not be used.

3. If it's just that BinkleyTerm Poll does not do what you expect, the reason is that BinkleyTerm POLL causes a direct deamon poll attempt to the called system (repeated calls, one after the other, until successful). It IGNORES the EVENT file and therefore ignores any flags in that file intended to limit frequency of calls or number of attempts.

"Modem Protocol Negotiation Filtered". What does this mean?

In some situations, incoming MNP or V42 negotiation sequences would cause early versions of BinkleyTerm to improperly respond to incoming callers. In this release, the negotiation sequences are recognized (if your modem does not filter or use them internally) and will properly be discarded.

Hints from the Binkley Echo

1. Some modems report "OK" to the screen after the modem int. To suppress this add one or more ~ at the end of the init string. E.g., Init AT<whatever>|~~

2. Having difficulty getting EMSI connects? try adding "NoFilter /Arq" to your Binkley configuration file.

3. *** D.D.: Someone asked "..all my file areas are set to twit, so FileSec = 0 should work since it allows disgrace access for freq'ing right?"
It's backwards from what you might think. At 0 you only allow freqs for the file areas twits can access. Set it higher.

4. *** D.B.: Using USR modem and found the AfterCall statement didn't actually give any figures... What you need to avoid is having an ATZ in your init string as that will reset the I6 display to all 0's when Binkley uses it to hang-up the phone.
*** I.S.: added. "Aftercall is issued after your PreInit (if any) then Init strings. These are used by Bink as its hang-up string".

5. *** C.F.: answered the question " Every connection I get says So & So's BBS from Somewhere. What causes this?"
It is caused by the failure of the remote system to include a "MyLocation" entry in the EMSI section of their BinkleyTerm or OtherMailer(tm) configuration file. There is nothing amiss on your end when this occurs.

6. *** S.H.: Answered the reverse question.." Binkley keeps insisting I'm from "Somewhere". I would like it to say "Pittsburgh, PA" or whatever.
Comment out or omit the NoEMSI statement in your configuration file and add these statements (all are required):

```
MyLocation
MyPhone
MyListFlags
MyMaxBaud
```

7. *** M.W.: dealt with the query .."I have only 1 node running Binkley, yet I get a message saying: other node sending to 1:153/xxx "
Your system probably got interrupted while packing mail or making a call so you've got a .BSY flag (zero byte file) still left behind in your outbound directory (or possibly your 'Flags' directory, if you have one defined)

8. *** I.S.: Commas are not delays, except after ATD.

9. *** P.N.: In order to go "Off Hook" the usual command is ATH1 but some new modems do not stay off-hook after an ATH1. It's becoming fairly common, and a work-around for it is "@echo AT X1 D; > COM1:"
The semicolon following the D is not optional, it's needed for this command to work.

10. *** B.B.: I put a blank line in my EVT file (just an extra carriage return at the end of the file) and found this gives an error msg.

And last but not least from the "Help" man himself...

11. *** B.J. :The *ONLY* way the "FaxBaud xxxxx" verb will have the desired effect for modems that need the port shifted to a specific speed during fax operation, is to *NOT* lock the port using FOSSIL or SIO, and let Bink do the port locking by use of the "LockBaud" verb.

"LockBaud xxxxx" tells Bink to lock the port for xxxxx reported connect speed and higher. "LockBaud" alone is the same as "LockBaud 0" (lock at all speeds). The value of "Baud" will be the actual speed of the lock. In other words, "Baud 57600" and "LockBaud 14400" in BINKLEY.CFG would tell Bink to lock the port at 57600 bps for all connects reported by the modem as "CONNECT 14400" or higher.

"LockBaud /Arq" would tell Bink to lock the port at the value of "Baud" for all connects that have the string "/Arq" reported (as in CONNECT 2400/Arq)

Remember, for either of the above "LockBaud" variations to work, the FOSSIL must be installed "floating"; i.e. NOT locked. "LockBaud" and "Baud" do *NOTHING* if the FOSSIL's locked. Bink will *display* the value you put in for "Baud", but the port (if locked by the FOSSIL and not Bink), will stay at the speed originally specified in your X00 or BNU installation line.

Both of these scenarios require that your modem also be able to react appropriately and in concert with Bink locking and unlocking the port.