

BINKLEYTERM

A FREELY AVAILABLE FIDONET COMPATIBLE
ELECTRONIC MAIL INTERFACE AND DUMB TERMINAL PACKAGE
VERSION 2.60

Technical Reference Manual

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

Copyright (C) 1987-1996 Bit Bucket Software, Co.

Terms and Conditions Contained Separately

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

"BinkleyTerm" and "Freely Available"
are trademarks of Bit Bucket Software, Co.

Table of Contents

WORKING WITH BINKLEYTERM

SYSTEM SECURITY

- Session Passwords
- Secured Inbound File Areas
- Protected Systems
- Known Systems
- NoSec Systems
- Implementing Incoming Mail Security

CONTROL OF FILE REQUESTS

- Outgoing file requests
- Incoming file requests
- Update Requests
- Request Response Files
- Response file Templates
- Template Statements:
- Function Requests

BBS INTERFACE

- BBS Exit
- BBS Spawn
- BBS Batch
- Passing control to the BBS

EXTERNAL MAIL PROGRAMS

MULTIPLE BBS SYSTEMS

POINTS

FAX

- Is your modem suitable?
- Adaptive Answer
- Baud Rate
- Fax Commands
- AT+FLI="xxx-xxx-xxxx"
- Modem Fax Response String
- Handling the incoming faxes internally with BinkleyTerm
- Handling Fax Reception with a Rear-end program
- Fax Transmission

DOMAIN ADDRESSING

DIAL TRANSLATION

FLAGS OR SEMAPHORE FILES

BI-DIRECTIONAL PROTOCOLS

THE JANUS MAIL PROTOCOL

- Configuring Janus
- Configuration File Statements for Bi-directional protocols

HYDRA MAIL PROTOCOL

CALL COSTING

The American Costing System

The Eurocost System

EMSI

THE CONFIGURATION FILE

LAYOUT

FULL ALPHABETICAL LIST OF CONFIGURATION STATEMENTS

EVENT FILES AND SCHEDULING EVENTS

EVENT FILE FORMAT

PARAMETERS AND FLAGS FOR EVENT FILES

SCRIPTS

SCRIPT FILE USAGE

PREPARING SCRIPTS

SCRIPT STATEMENTS

THE BINKLEYTERM WINDOWED INTERFACE

VIDEO FOSSIL

THE DISPLAY SCREEN

CURRENT SETTINGS WINDOW

TODAY AT A GLANCE WINDOW

PENDING OUTBOUND MAIL WINDOW

RECENT ACTIVITY WINDOW

TRANSFER STATUS WINDOW

BINKLEYTERM LOG FILE

COMMAND LINE PARAMETERS

TERMINAL MODE KEYSTROKES

UNATTENDED MODE KEYSTROKES

ZOOMED OUTBOUND WINDOW KEYSTROKES

MODIFICATION/TRANSLATION OF BINKLEYTERM INTERNAL TEXT

WORKING WITH BINKLEYTERM

SYSTEM SECURITY

Session Passwords

BinkleyTerm supports session-level password protection. Using such protection helps prevent situations where someone uses a FidoNet mailer to "impersonate" a legitimate FidoNet node, and pickup mail addressed to the node. When implemented, both the sending and receiving systems must have it in operation, with the same password on both ends (exceptions noted below).

With Point systems, password protection can be automatic. Simply use the 'BossPwd' configuration file option, and choose a password. Make sure your Boss also uses the same password.

For other types of systems, password information is kept in the nodelist data file. ParseLst (and some other nodelist processors) can easily add a password to a nodelist entry. Refer to the documentation for your nodelist processor.

When talking with other BinkleyTerm, Opus, FrontDoor or compatible systems that use the YooHoo (WaZOO) or EMSI session negotiation methods, full, two-way protection is available. The systems connect and exchange the password right at the beginning of the session. If the passwords do not match, the session is terminated, regardless of who initiated the call.

The only exception is when you have a password implemented, but the remote has NO password implemented. In this case, if YOU placed the call, then a session will still occur. If the REMOTE placed the call, then the session will be aborted.

When connected with systems that do not support WaZOO or EMSI session handshaking (such as older versions of SEAdog), password matching takes place only when you are picking up mail from the remote system. You may send to such a system without a password match taking place. (The assumption is that you always know who you're calling, but don't always know who's calling you.)

Note that session passwords must not exceed eight (8) characters, and are case insensitive.

When BinkleyTerm cannot find a password for a remote system when placing a call, it will not check for a password during the session. This responsibility is left to the remote system, if the remote chooses to perform checking.

BinkleyTerm allows easy implementation of new passwords. Simply add the agreed upon password as outlined above, and send the remote system a message. BinkleyTerm will allow an outbound session to take place in cases where you have designated a password, but the remote has not yet implemented it. This is handy for letting a remote system know that a password was implemented. Note that in this case, BinkleyTerm will NOT allow the remote to have a successful session when the remote calls your system; only when you place the call (as stated above, the idea is that you know who you're calling, not who's calling you).

Secured Inbound File Areas

Another method of securing the flow of mail involves controlling the processing of incoming mail to your system. In most cases, you will want to protect common mail links with session passwords, as outlined in the previous section. Still, the potential exists for another abusive system to send you rogue mail, or mail "planted" onto your system in hopes that it will be routed to another system at your expense, or find its way into a national EchoMail conference.

BinkleyTerm makes it possible for mail from "Protected" and "Known" systems to be directed to separate inbound areas whilst mail from unknown systems (or all systems if security is not implemented) will be placed as designated by the 'NetFile' statement.

Protected Systems

A group of systems with which you have implemented session passwords. These links are "protected" by passwords. See page 1

"Protected" systems are the top-level, or "most favored" systems since you generally know the other person at the other end (a pre-requisite to establishing a password).

Configuration Statements concerned with these systems commence with the letters "Prot"

Known Systems

Systems that are listed in the nodelist ("known" to you) but with whom you have not established a session password.. This group represents the middle-level of security.

Configuration statements concerned with these statements commence with the letters "Known"

NoSec Systems

All mail not "filtered out" by a Prot or Known statement is placed in the area designated by the statement in this group.

Obviously, if no security arrangements are enabled then ALL incoming requests are dealt with by the NoSec group.

When Prot and/or Known statements have been enabled, all other systems, i.e. those without passwords or not in the Nodelist, for example the "Points" of other systems, are considered as "unknown" or no security (NoSec) systems.

This group of systems represents the lowest-level of security, since you really have no idea who owns such systems, or where they are located.

Implementing Incoming Mail Security

The configuration file statements used to implement redirection of the incoming mail are as follows:

SYSTEM GROUP	PROTECTED	KNOWN	NO SEC
-----	-----	-----	-----
STATEMENT	ProtInbound	KnownInbound	NetFile

It is essential to enable the 'NetFile' statement if you wish to receive mail or files as this is the "catchall" statement telling BinkleyTerm where to put all mail and files not directed elsewhere.

The statements in the "Prot" and "Known" groups are optional.

In a typical "secured" installation, you would arrange that mail is automatically processed only if received to the 'ProtInbound' area. Mail received to 'KnownInbound' or 'NetFile' must be examined and processed manually. Of course, you could choose to configure your system in a such a manner that mail received in any of the three areas is processed automatically to your liking, possibly to special message areas.

The only "hole" in this is with FTS-0001 sessions, such as those with SEAdog or Fido. Since BinkleyTerm has no way of knowing the address of the sending system until a packet is received, that first packet (.PKT file) will always be placed in the default area specified by the 'NetFile' statement. Once the packet has been received, BinkleyTerm will then move it to the appropriate directory if need be.

CONTROL OF FILE REQUESTS

BinkleyTerm supports the two popular file request methods currently in use in FidoNet: "Bark" requests as designed for and used by SEAdog, and "WaZOO" as designed for and used by Opus-BCBS. Either style can be used on an incoming or outgoing basis.

Outgoing file requests

These can be generated by BinkleyTerm or by using one of the many utilities designed for the purpose, such as Amax, Bonk, Please, etc. Any Opus-compatible file request builder can be used with BinkleyTerm. You may also manually build file requests. They are a single-line flat ASCII text file, and are named in the same manner as packets (refer to the User Manual section "How BinkleyTerm Handles Mail" for information on the naming convention) and have a file extension of .REQ. Outgoing requests should NEVER have a drive and path designation; only the file name. The remote system handles the drives and paths.

The .REQ file is placed in your outbound holding area.

BinkleyTerm will not place calls for a stand-alone .REQ file. In order for a call to be placed, a packet or file attach of the proper flavor must also exist (this will be prepared, along with the .REQ file, by utilities such as Bonk and Amax). You can also manually poll to send the request immediately.

Incoming file requests

Where the facility is required, BinkleyTerm offers a method of establishing limitations on file requests received from systems that are not session password protected, or are not found in the nodelist.

This support is optional; if you do not wish to limit file requests in this manner, use only the files and configuration file statements mentioned in the NoSec Group.

The list of statements concerned with File Requests may be tabulated as follows:

Protected Group -----	Known Group -----	NoSec Group -----
ProtReqList	KnownReqList	OKFile
ProtAvail	KnownAvail	Avail
ProtAbout	KnownAbout	About
ProtReqLim	KnownReqLim	MaxReq
ProtReqTpl	KnownReqTpl	ReqTemplate
ProtSec	KnownSec	FileSec

The rule is simple, BinkleyTerm will use the configuration statement in the "NoSec" group (provided that you have enabled it) in all cases where no equivalent higher security statement has been enabled

1. OKFile <filespec>

(Also the secured statements ProtReqList and KnownReqList)

Designates a file (the full drive, path and filename should be given) containing a listing of files, in machine-readable form, that you will allow to be sent to remote systems via file request. Wildcards are allowed, and nearly always used. When an incoming request is received, the request is checked against the listing to see if you permit the file to be sent.

It is essential, if you wish to allow File Requests, that the OKFile statement is enabled as it is the "Catchall" statement used by all requests not directed elsewhere. The equivalent Prot and Known statements are optional.

OKFILE.TXT is often used as the filename for the OKFile statement in the NoSec group.

Magic Filenames

You may implement "magic filenames" which are words used to represent a file. For example, you could set-up your OKFILE in such a way as to send BDOS_260.ZIP if someone requests "BINKLEY" from you. This frees the person making the request from having to know the exact filename of the file he wants. "Magic" filenames are generally used by systems which are software distribution points, hubs, and so on.

Password Protection

You may implement password protection for any particular file or group of files.

Faster File Searches

If you run a Maximus BBS then its MAXFILES.IDX can be used for file request searches. Adding "*path\MAXFILES.IDX" in the OKFILE list lets BinkleyTerm search through this database when processing file requests. This is much faster than seeking through the whole of a partition (and it's more "multitasking friendly", too!). The asterisk (*) is required.

You will have to enable the following statements in your BINKLEY.CFG:

```
MaxAreas d:\max\area.dat
```

FileSec n or the equivalent KnownSec n and or ProtSec n where n is the following security level:

0=Disgrace	4=Privileged	8=Asstsysop
1=Limited	5=Favored	10=Sysop
2=Normal	6=Extra	11=Hidden
3=Worthy	7=Clerk	-2= Twit

Each caller's access rights in the Maximus file area will be compared to the level set for NoSec (FileSec n), Known (KnownSec n) and Prot (ProtSec n) as stated above. If his security level is too low, the caller will get a password error.

Search details will be logged if you have a log file implemented.

Sample OKFILE:

```
b:\aprog.ARC
c:\file\stuff\*.*
c:\file\programs\wlc*.txt
c:\file\private\*.* !mypass
@BINKLEY c:\file\dist\bdos_260.lzh
@MYEDIT !outpass c:\file\private\editmax.arc
*c:\max\maxfiles.idx
```

The first line gives an exact filename. The second and third lines show the use of wildcards. The fourth line shows password protection, with an exclamation point (!) followed by the password. The fifth line shows a magic filename, an "at" symbol (@) followed by the magic filename, followed by an exact drive, path and filename designation. The sixth line shows a magic filename with password protection. The seventh line shows the use of the MAXFILES.IDX. The initial asterisk is required. Note that in all cases, a password (if any) is always the second argument in an OKFILE entry.

If you want to send two or more files from one magic filename simply specify the same magic filename for each file.

For example:

```
@GOODSTUFF drive\path\file.zip
@GOODSTUFF drive\path\dif_file.lzh
```

2. Avail <filespec>

Also the secured statements ProtAvail and KnownAvail)

This designates a file containing a list of the files you have available for request, in human readable form. The list will be a flat ASCII text file, giving the name of each file, and usually its size in bytes and a short description. There are utilities available that can construct this file automatically based on your BBS system's internal listings. Of course, it could also be manually created. When someone requests the magic filename "Files" BinkleyTerm will send this file.

It is usual, but not essential, to have the NoSec version of this statement enabled. The Prot and/or Known equivalents are optional.

3. About <filespec>

(Also the secured statements ProtAbout and KnownAbout)

When someone makes a request for the magic filename "About" or when someone makes an invalid WaZOO file request, this file is sent by BinkleyTerm. It normally contains a short advertisement about your system, as well as a notification that the calling system could possibly have made an invalid request.

This is a flat ASCII text file in human-readable form.

It is usual, but not essential, to have the NoSec version of this statement enabled. The Prot and/or Known equivalents are optional.

The About file is sent only on failed WaZOO requests. The file is NOT sent on failed Bark requests.

4. ReqTemplate <filespec>

(Also the secured statements ProtReqTpl and KnownReqTpl)

BinkleyTerm has the ability to generate a response file based on a template that you preset. See page 6. This is optional.

If you configure a template file the About file will not be sent.

5. MaxReq <number>

(Also the secured statements ProtReqLim and KnownReqLim)

This designates the maximum number of files that will be sent during one file request session.

6. FileSec <number>

(Also the equivalent secured statements ProtSec and KnownSec)

Only needed if the MaxFiles.IDX search method is being used. See page 4.

Update Requests

Update requests are a method of obtaining an "update" of a file you already have, from another system you believe to have a newer copy. They differ from file requests in that when you construct an update request, you include a drive/path/file specification that points to an existing file on your system. Generally, wildcards are acceptable. When BinkleyTerm connects with the desired remote, it will compare the date and time stamp on your copy of the file(s) to the date and time stamp of the same-named file(s) on the remote system. If the remote system has a newer copy of a given file, it will be sent. If it does not, no file will be sent.

Note that the drive and path specification DO NOT need to be the same on the remote system; the drive and path you give refer only to YOUR copy of the file. The remote may have any file structure.

Update requests were originally implemented in SEAdog, a mailer from System Enhancement Associates, Inc. In addition to supporting update requests with SEAdog systems, BinkleyTerm implements a WaZOO update request for use with other BinkleyTerm systems, or other mailers that support WaZOO update requests.

Update requests are most commonly used to handle GroupMail, a method of sharing conferenced or group message areas developed by System Enhancement Associates, Inc. GroupMail generally requires update request capability on both ends of the connection.

Like file requests, update requests are kept in .REQ files in your outbound area. In fact, a combination of update and file requests can be contained in the same physical .REQ file. An update request entry contains a filename, as well as a date and time code that corresponds to the date and time stamp of the file on your system. Because the date and time code is in a special format, it is not recommended that you attempt to create an update request entry yourself.

Various utility programs such as Amax are available which support the creation of update requests in the manner that BinkleyTerm requires.

Please note that in order for update requests to work, the remote system must have the file you want updated available for regular file request. You cannot update a file that would not otherwise be available for file request from the remote system.

Request Response Files

If a File or Update Request is unsuccessful, the file designated by the 'About' statement is sent, by default, as a reply (assuming that the About statement has been enabled)

If the 'ReqTemplate' statement is enabled a more informative file will be built during the session based on a template file which you preset (see next section).

If the 'PktRsp' configuration statement is also enabled, use of the 'ReqTemplate' statement will build a message packet instead of a file. This will then appear as regular mail to the calling system. Note that you must have a flags directory specified in your configuration file as this is used when building the packet.

Notes

1. The "About" file is not sent if 'ReqTemplate' is enabled.
2. When a response file is sent, it is counted against the maximum number of file sent in response to a request, as designated by the 'MaxReq' statement in the configuration file.
3. Response files are named similarly to outbound packets or request files. They are named <net><node>.RSP. Both <net> and <node> are four- digit hexadecimal numbers with leading zeroes.
4. The response file is a plain text file. Its content is preset by creating a template file.

Response file Templates

Normally, the response file contains information such as the date and time the request was made, what file was requested, and why the request failed. You may wish to include information such as the times your system accepts requests, what magic filenames you support, and so on. Response files will be generated according to the template under any one of the following cases, called reason codes:

- 1 - File Not Found
- 2 - No Update Necessary
- 3 - Password Missing or Wrong
- 4 - Requested no. of Files exceeded MaxReq
- 5 - Start of No-Requests-Honored Event
- 6 - Requested no. of Bytes exceeded MaxBytes
- 7 - Requested no. of Minutes exceeded MaxTime
- 9 - Successful Request

By default, when a template is designated in the configuration file, BinkleyTerm will always generate a response file for all of the above reasons. It is possible, however, to control or limit what the response file says based on a particular reason code, or to not have a response file generated at all for a particular reason code.

The template file designates exactly what is to be placed in the finished response file. Most options perform simple macro substitution; others allow conditional inclusion of text, or instruct BinkleyTerm what to do with the response file.

BinkleyTerm uses the template file serially, and copies everything found in the template directly to the response file, performing substitution or conditional copies as directed by template file statements. When the end of the template is reached the response file is closed and sent to the calling system (unless an %exit statement is used before the end of the file).

All the allowed statements in the template begin with a percent sign (%), a character which should not be used for any other purpose within the file.

Template Statements:

Statement -----	Function -----
%;	When placed in column 1 (far left), denotes a comment line. When in any other column, indicates that remainder of line should be ignored.
%abort	Don't send response file.
%abort <number>	When <number> matches the reason code, don't send response file.
%bink	Copies BinkleyTerm program name and version into the response file.
%date	Copies the current date into the response file.
%exit	Close the response file and send as-is.
%exit <number>	When <number> matches the reason code, close response file and send as-is.
%line <number> <text>	If <number> matches the reason code, <text> is copied into the response directly. The <text> should not exceed 255 characters in length, and may contain other template file statements such as %system, %date, and so on.
%mynode	Copies your node address into the response file.
%request	Copies into the response file the actual line from the incoming request that prompted the creation of the response file. Normally this is the name of the file that caused the response file to be generated.
%status	Copies text previously defined for the current reason code with the %text statement into the response file.
%sysop	Copies your Sysop name, as given in the BinkleyTerm configuration file, to the response file.
%system	Copies your system name, as given in the BinkleyTerm configuration file, to the response file.

%text <number> <text>	When %status is given later in the template, <text> is copied into the response file if <number> matches the reason code. The <text> should be no longer than 255 characters in length, and may contain other template statements such as %system, %date, and so on.
%time	Copies the current time into the response file.
%yrnode	Copies the node address of the calling system into the response file.

Function Requests

Two additional special entries that operate like magic filenames may be included in the file list specified by the OKFile statement. These function requests allow an incoming request to trigger the operation of a program.

\$ Function Requests

The first style uses a dollar sign (\$) in the first column of an OKFILE entry, followed by a function request name. If the name is matched, then the command after the password (if any) is executed. In addition, three arguments which correspond to the net, node and point numbers of the calling system can be sent if desired by adding "%d %d %d" to the end of the OKFILE entry. For example:

```
$INFO INFO.BAT %d %d %d
```

If an incoming file request is for "INFO" then the program INFO.BAT would be executed, and it would receive three command line arguments that correspond to the net, node and point numbers respectively. For instance, if 141/491.0 requested "INFO" then the following would be issued to DOS for execution:

```
INFO.BAT 141 491 0
```

Another example of such an OKFILE entry would be:

```
$CLEANUP !mypass CLEANUP.BAT %x %x %x
```

In this case, a password is included as the second argument of the line, and must be matched by the incoming request in order for the program to be executed. Note also that in this example, "%x %x %x" was used, and would cause a hexadecimal representation of the net, node and point number to be sent as command line arguments to the program being executed. In our example, if 104/36.10 requests "CLEANUP" with a password of "mypass" then the following would be sent to DOS for execution:

```
CLEANUP.BAT 68 24 a
```

Note that you can use a "%04x %04x %04x" mask to pass the numbers as 4 hex digits; handy for directly building .?LO addresses in a batch file, or a simple program. For example, if you specified:

```
$CLEANUP !mypass CLEANUP.BAT %04x %04x %04x
```

Then using the same example, if 104/36.10 requests "CLEANUP" with a password of "mypass" then the following would be sent to DOS for execution:

```
CLEANUP.BAT 0068 0024 000a
```

Cleanup.bat could then do something like:

```
set addr=%outbound%\%1%2
if %3 == 0000 goto notpoint
if not exist %addr%.PNT\nul mkdir %addr%.PNT
set addr=%addr%.PNT\0000%3
:notpoint
```

REM process the request, generate output file/s,
REM send each with:
echo [^#]d:\path\filename.ext >>%addr%.QLO

See page 10 for additional information

+ Function Requests

A second type of function request is also supported, and is designated by a plus sign (+) in the first column of an OKFILE entry. In this case, if the filename and optional password are matched, then the entire line from the incoming .REQ file is passed to DOS for processing, with the <zone>:<net>/<node>.<point> address appended as individual arguments to the command line. An example of an OKFILE entry might be:

```
+GETREV !mypass
```

If an incoming .REQ file from 1:343/491 contained the line:

```
GETREV !mypass BT 2.60
```

Then the following would be passed to DOS for execution:

```
GETREV !mypass BT 2.60 1 343 491 0
```

A program or batch file called GETREV would be executed, and would need to process or filter the command line as needed.

Note that the <zone:net/node.point> number arguments are always passed as decimal, and cannot be specified, as they can with \$ requests.

Note also that only + function requests pass the caller's zone, so it may be necessary to receive a + request, saving the zone, before any \$ requests intended to build output for other-zone callers. Function requests are convenient for allowing a remote system to have a batch file or utility of some kind (not included) place a particular file on hold for a node for later pickup, to cause the system to send a particular file, to trigger some other sort of function or activity remotely, etc., etc.

If a program called as part of a function request creates a file in the appropriate outbound area called <net><node>.QLO (where <net> and <node> are 4-digit hexadecimal numbers with leading zeroes), BinkleyTerm will treat this file as a legitimate "flow" (file attach) file and send the file(s) listed in it back to the caller as part of the request logic.

Function Request Notes

If a function request triggers a program or batch file to build a file attach (flow) file, then BinkleyTerm will process the file attach during the current session. In other words, if a file attach is generated during a function request, then the file(s) shown in the file attach will be sent during the current session.

Note that the normal logic for the handling of .HLO (Hold file attaches) still applies; i.e., file(s) designated within a .HLO file will be sent only when the destination node polls for pick-up.

BBS INTERFACE

One of the most common uses of BinkleyTerm is as a mail front-end for a bulletin board system, or "BBS". BinkleyTerm offers three different methods for passing control to a BBS. The method used is determined by a configuration file statement.

Method 1

BBS Exit

When a caller wishes to access the BBS, BinkleyTerm will exit to the start-up batch file with an errorlevel of the baud rate divided by 100. For example, a 300 baud caller exits to the batch file with an errorlevel of 3, a 2400 baud caller at errorlevel 24. See the table and notes in the User Manual, under Errorlevels, for higher baud rate exit details.

The batch file then detects the errorlevel, and jumps to a section of the batch file you designate to start the BBS program with the baud rate information.

When the call ends, the batch file should restart BinkleyTerm.

This is the method which the original version of BinkleyTerm used to start a BBS. It is still used in some cases but with the continued advance of technology it presents some problems: ie., the batch file used to control program operation becomes rather unwieldy and now that modems with higher baud rates are in general use dividing the baud rate by 100 does not always provide a unique errorlevel.

Moreover the method does not provide for passing any parameters other than the baud rate and it cannot be used with OS/2 or Win32 operating systems as the connection will be lost when BinkleyTerm exits (ie before the BBS starts).

Notes on Methods 2 & 3

Many BBS programs can accept parameters on the command line which provide extra functionality. (Refer to the documentation of your BBS for details)

When the older 'BBS Exit' option is used it is not possible to pass these parameters, but when the 'BBS Spawn' or 'BBS Batch' options are employed, BinkleyTerm can pass the following five parameters:

%1	The speed of the computer to modem link rate in bps
%2	The caller's connect rate as reported by the Modem
%3	The communications port which is in use
%4	The time to the next Event in minutes
%5	Any extended information given in the modem connect string (/ARQ, etc.)

When using either the 'BBS Spawn' or 'BBS Batch' options you need to create a batch file called SPAWNBBS.BAT (or, if using OS/2, SPAWNBBS.CMD) whose function is to start your BBS program with as many of the above parameters as the BBS program can accept. Any or all of the above parameters may be used in the normal way within the batch file.

The way in which SPAWNBBS is invoked depends on whether the 'BBS Spawn' or the 'BBS Batch' option is to be used, but in either case BinkleyTerm will provide the actual parameter information for the current call and you "get at" that information by using the correct parameter descriptor (%1-%5) in the batch file called SPAWNBBS.BAT (for DOS or Win32) or SPAWNBBS.CMD (for OS/2) which you write to start the BBS.

An example of the parameter information provided by BinkleyTerm for use by the SPAWNBBS file is:

```
SPAWNBBS 9600 2400 1 47 /Arq
```

The information is the DTE link rate in bps, the baud rate, port number, time in minutes to the next non-BBS event and extended information from the connect string. In this case, there is a 9600 link rate, a 2400 baud caller, on communications port 1, with 47 minutes remaining to the next non-BBS event, and a modem information string of /Arq.

Method 2

BBS Spawn

OS/2 and Win32 users should always use this method. With methods 1 and 3, the operating system will drop the carrier during the handover to the BBS, and the connection will fail.

This method can also be used when running under DOS but as BinkleyTerm remains resident, memory shortage may be a problem (depending on the size of your BBS program) unless you also enable the 'SwapDir' configuration statement (see page 50)

When a caller wishes to access the BBS BinkleyTerm will execute "SPAWNBBS" as a child process (or "shell") providing information for the five parameters mentioned above.

When the call ends the child process will terminate and control will pass back to BinkleyTerm.

IMPORTANT NOTE for OS/2 and Win32 (Windows NT/Windows 95) users:

The third parameter provided for SPAWNBBS describes the comm port but OS/2 and Win32 systems use comport "Handles" instead of a fixed Port number. These port handles may change "on the fly" so the Port number should not be hard coded. BinkleyTerm knows this and passes the comport handle as parameter 3 (%3) which is the information expected by Maximus, Lora BBS etc..

Method 3

BBS Batch

This method is not recommended under OS/2 or Win32 systems as the connection is lost when BinkleyTerm exits. For systems running under DOS, the 'BBS Batch' method uses the best of both of 'BBS Exit' and 'BBS Spawn' and makes the most efficient use of memory.

When a caller wishes to access the BBS, BinkleyTerm will physically write a file to the disk. This file will be named BBSBATCH.BAT which will consist a single line, the filename "SPAWNBBS" and 5 parameters. See example above. After writing this file, BinkleyTerm exits with an errorlevel describing the baud rate of the caller (as for 'BBS Exit', see method 1, above).

Passing control to the BBS

Method 1, Errorlevel testing

This method has been described under the BBS Exit heading. With modern modem speeds many errorlevels need to be checked so the batch file becomes complex.

Your main batch file, usually BINKLEY.BAT, will decide what happens when a BBS related errorlevel is received, and it should be set to execute BBSBATCH.BAT which in turn will execute BBSSPAWN which you will have written in such a way that it starts the BBS, passing to the BBS any or all of the five parameters pre-set by BinkleyTerm

When the call ends your batch file should restart BinkleyTerm

Method 2, "If exists"

This method does not depend on checking each BBS related errorlevel and these checks can be omitted from your main program control file (probably called BINKLEY.BAT or BINKLEY.CMD) which will only now need to test for NON BBS related exits such as "mail received" etc..

Amend your main program control batch file as follows:

BINKLEY.BAT (change all references to ".BAT" into ".CMD" if using OS/2)

:TOP

rem First delete any old BBSBATCH file which exists

if exist BBSBATCH.BAT del BBSBATCH.BAT

rem Now start BinkleyTerm in your usual way. Example:

BT share unattended

rem When Bink exits, check for presence of BBSBATCH.BAT, if it now exists

rem Bink has made it because a caller wants into the BBS

if exist BBSBATCH.BAT goto BBS

rem Now check for NON BBS exits, fro example:

if errorlevel 20 goto mail_in


```

rem  etc., etc..

:BBS
rem  Call BBSBATCH which runs SPAWNBBS.BAT with %1=connect rate
rem  %2=portrate etc..(these params being provided by Bink)
call BBSBATCH
rem  Now loop round and restart BinkleyTerm
goto TOP

```

EXTERNAL MAIL PROGRAMS

BinkleyTerm can be used with external mail programs, such as UUCP. When an 'ExtrnMail' statement is used in the configuration file, BinkleyTerm will answer the phone and wait for a YooHoo or TSYNC (the start of a mail session) or a string you designate with the 'ExtrnMail' option. When the designated string is received from the remote system, BinkleyTerm will send the string associated with the 'MailNote' configuration file option (if any), then will physically write a batch file called MAILBAT.BAT to the disk, and exit the start-up batch file with an errorlevel as given with the 'ExtrnMail' statement, or with errorlevel 99 if none was given. MAILBAT.BAT contains a single line:

```
EXTMAIL %1 %2 %3 %4 %5 %6
```

where:

```

%1 = speed of the computer-to-modem link rate in bps
%2 = caller's connect rate reported by the modem
%3 = the comm port in use (or in OS/2and Win32, the port handle in use)
%4 = time to the next event in minutes
%5 = errorlevel exit in your ExtrnMail statement(default 99)
%6 = extended info in the modem connect string (/ARQ, etc.)

```

Notice the similarity in concept between this and the 'BBS Batch' method of invoking a BBS (as described previously).

When BinkleyTerm exits with the previously defined errorlevel, your batch file must capture the errorlevel and invoke MAILBAT.BAT, which will in turn invoke EXTMAIL.BAT with the various command line parameters shown above.

If you prefer to leave BinkleyTerm resident you can include the configuration statement 'Extrn spawn' and all external mail exits will be spawned in a manner similar to a 'BBS spawn'. That is to say, when the 'ExtrnMail' statement causes an exit

```
EXTMAIL.BAT %1 %2 %3 %4 %5 %6
```

will be executed as a child process, the parameters being as described above.

In either case, EXTMAIL.BAT is then used to invoke the external mail program of your choice, taking the needed command line parameters as supplied by the batch file, and processing and/or using them as needed. Once the external mail program session has been completed, your original start-up batch file for BinkleyTerm should be invoked to place BinkleyTerm on-line again.

NOTES:

1. If you're locking your FOSSIL driver, the link rate and connect rate passed by BinkleyTerm will be the same (unless the connect rate is one of the intermediate rates reported by some newer modems). BinkleyTerm has no way of knowing the port's been locked unless it does the locking itself via the 'LockBaud' configuration statement. For further details, please refer to the detailed link/connect rate table in the User Guide.
2. The entire "Connect" line is treated as a potential external mail string. This allows BinkleyTerm to shell out to an external program for some FAX modems.

3. OS/2 users should use the 'extern spawn' option and set up a EXTMAIL.CMD file.
4. Win32 users should use the 'extern spawn' option.

Uses for the external mail facility:

1. Fax Modems. The facility allows BinkleyTerm to shell out to an external program to deal with Fax.
2. Multiple BBS systems are possible and are described more fully in the next section.

MULTIPLE BBS SYSTEMS

With the external mail facility it is possible for multiple BBS systems to reside at the same telephone number, and to give BBS users the ability to select the desired BBS from a menu. It would be possible to run completely separate systems, to run the same system with different BBS packages, and so on. This section will give an overview of the procedure.

Basically, 'ExtrnMail' statements in the configuration file are used to build the menu options themselves. If you want to exit to the batch file when a user presses "A" on their terminal, then a configuration file entry like this would be needed:

```
ExtrnMail 130 A
```

This would cause BinkleyTerm to write MAILBAT.BAT to the disk, and exit to the batch file with an errorlevel of 130. As described in the previous section on external mail programs, your batch file should invoke MAILBAT.BAT, which in turn invokes EXTMAIL.BAT. EXTMAIL.BAT would be the batch file that acts upon the choice, and physically invokes the desired BBS program.

If you want the choices to be case-insensitive, then two 'ExtrnMail' statements would be needed for each letter, like this:

```
ExtrnMail 130 A  
ExtrnMail 130 a
```

This would cause BinkleyTerm to use errorlevel 130 when either "a" or "A" is received from the user.

The menu presented to users should be kept in a file called BINKLEY.BAN (refer to the "Banner" section below for more information). This file might look like this:

```
Welcome to Multi-System 100. Please choose the desired BBS:  
A - "Garden Central" Using QuickBBS Software  
B - "Margarita Bay" Using Opus-CBCS Software  
C - "Margarita Bay" Using Fido 11w Software
```

Such a menu would allow users three options. Each lettered option would require a separate 'ExtrnMail' statement in the configuration file; two each if case insensitivity is required.

A string "Make your selection by letter..." is added with a 'Banner' statement in the configuration file so that the line is re-displayed each time a user makes an incorrect choice (refer to "Configuration File" section for information).

Once the user has made a selection, and MAILBAT.BAT is invoked and in turn EXTMAIL.BAT is invoked, then the batch file must determine which selection is made and act upon it.

In EXTMAIL.BAT (where the BBS systems are invoked), the top of the batch file should make a determination as to which choice was given by the user. One of the parameters on the command line when EXTMAIL.BAT is invoked by MAILBAT.BAT is the errorlevel which corresponds to the choice the user gave (as designated by an 'ExtrnMail' statement in the configuration file).

A section of the batch file might look like this:

```
if %4 == 130 goto bbs1
if %4 == 140 goto bbs2
if %4 == 150 goto bbs3
```

The fourth parameter given on the command line is the errorlevel value, and is reference by "%4" as shown. For example, if the errorlevel that corresponds to the choice was 130, then batch file execution would jump to the "bbs1" label, and invoke a particular BBS program.

Please note that the default is for a user to press the "escape" key to enter the BBS. One of the BBS systems should be setup as the default system as described in the section "BBS Interface". Should a user press "escape," or if the time limit to make a response should expire, the default BBS would be invoked.

Note also that the configuration file parameter 'Timeout' should be extended to allow a user more time to read and make a selection prior to making a default exit. A 'Timeout' value of 30 to 60 seconds is suggested.

Placing multiple BBS systems on-line at one number takes some work and experimentation in order to operate correctly. Hopefully the tips given here will point you in the right direction.

Using the Banner File

When a file named BINKLEY.BAN is present in the directory that BinkleyTerm is invoked from, the file will be displayed to callers immediately after the BinkleyTerm identification line.

The file is a flat ASCII text file, and may contain extended information for the benefit of callers.

The file can be used in combination with external mail processor definitions to present users with a rich BBS selection menu, or just to give the user something to read while waiting for a BBS to load.

POINTS

Points are essentially a "system under a system" and allow the point operator to have limited access to the network without the normal requirements of keeping a system on-line at certain hours.

For use as a point, check the 'BossPhone' and 'BossPwd' configuration statements. If both of these are enabled then a Nodelist is not needed by a point.

BinkleyTerm is well suited to the task of being a "boss" (or host system) of a point network and BinkleyTerm is also widely used by Points, needing only a very simple configuration file in that use.

Originally, point addressing was not part of the FidoNet standard and certain "kludges" were used in order to handle a point network. A private network was established under the Boss and "fake" node numbers were used. This method is still in use by some systems but is tending to be replaced by proper 4D and 5D addressing now that this is available.

BinkleyTerm uses 5D addressing (zone:net/node.point@domain) unless the configuration statement 'PrivateNet' is included, in which case the older privatenet (or fakenet) method is adopted.

It is suggested that wherever possible the new method should be used by new users. If a description of how to set up a private network is needed please refer to older BinkleyTerm documentation.

To set up the Boss end of a point network take the following steps:

In your private nodelist segment, include points as follows (be SURE that you EXACTLY duplicate the information in the distributed nodelist for your net host and your node):

Host,106,Houston_Area,Houston_TX, Scott Royall,etc,etc
 ,2000,COMM_Port_OS/2,Sugar_Land_TX,Bob_Juge,etc,etc,etc
 Point,1,Point_1,Houston_TX,John_Smith,etc,etc,etc
 Point,2,Point_2,Houston_TX,Mike_Jones,etc,etc,etc
 etc.

Run a nodelist processor which can generate points in Version6 (for small nodelists only) or Version7 format.

In the index to Version6, Points will be listed as -1/pointnumber in entries following the bossnode.

In the NODELIST.DAT file (NODEX.DAT for Version 7), point entries will be shown with node flag bit 12 (hex 1000) set to indicate that this entry is a point, and the hub node field will contain the point number instead of a hub.

Where is the outbound area for points? Let's say you are storing mail for points off of a system whose address is 1:132/491. You would do so by creating a directory 008401EB.PNT in your Zone 1 FidoNet outbound directory. (the hex representation of "132" is "84", "491" translates to hex "1EB", so "008401EB" represents 132/491 in hexadecimal)

If you were in Zone 1 of FidoNet, a crash packet to this system's point 12 ("12" is "C" in hex) would be something like:

```
C:\BINKLEY\OUTBOUND\008401EB.PNT\0000000C.CUT
```

FAX

Is your modem suitable?

Modem Classes

Modems fall into three classes:

Class 1

With this class of modem most of the work is done by the software.

Many modems in the class are not capable of adaptive answering (see below) and so they are *NOT* suitable for use with BinkleyTerm. A few modems in this class do have adaptive answering capability so consult your modem documentation.

Class 1 modems receive data in "Direct bit order".

Class 1 modems when in fax mode operate at a maximum baud rate of 19200

Class 2

This is quite different from Class 2.0

Like class 1, Class 2 modems can only operate for FAX purposes at a maximum baud rate of 19200.

Class 2 modems are capable of adaptive answering (see below) so they can be used with BinkleyTerm for automatic FAX reception

Class 2 modems receive fax in Reverse bit order.

Class 2.0

The official standard, called Class 2.0 to differentiate it from class 2 above, is NOT the same as Class 2. Such modems have several added features, the most important, for fax work, being that they are capable of receiving fax when the DCE rate (modem to computer speed) is set or locked at speeds above 19200 baud.

Class 2.0 modems receive data in direct bit order.

AT+FCLASS=?

If you send the above command to your modem whilst in Terminal mode you may well be able to find out which classes your modem can support.

A response of	0	indicates data mode
A response of	1	indicates class 1 fax capability
A response of	2	indicates class 2 fax capability
A response of	2.0	indicates class 2.0 fax capability
Error		indicates no fax capability.

Adaptive Answer

"Adaptive Answering", or "call selection" as USR chooses to call it, is a feature of Class 2 and Class 2.0 modems and is the modem feature that enables the MODEM to decide if an incoming call is Fax or Data and send out an appropriate response code (such as FAX or +FCO) for a fax call.

In general, Class 1 modems need to be set to receive fax *OR* to receive Data before a call is received. If set to Data they will not receive Fax, and vice versa. That is they do not adaptively answer the call but there are a few Class 1 modems, such as the Supra 144LC with the 1.80 revision ROMs which do support adaptive answering.

Baud Rate

As noted above, Class 1 and Class 2 modems can only operate for FAX reception at a computer to modem rate not exceeding 19200 baud.

This is a big snag with modern high speed modems where the baud rate may normally be locked at either 57600 or 112500.

Unless you can accept locking the baud rate for all transmissions to 19200 it becomes necessary with these modem classes to leave the fossil unlocked and to control the baud rate by other means.

Do not lock the Fossil but instead use the 'Faxbaud' statement to lock at 19200 for faxes and other controls (such as 'LockBaud /Arq') for data.

If the Fossil is not locked then BinkleyTerm will set the baudrate to 19200 on a FAX connection unless you set 'FaxBaud X' , in which case your specified rate will be used.

.

Class 2.0 modems can operate with a fossil locked at the speeds used for normal BinkleyTerm operations

Fax Commands

First of all note that BinkleyTerm translates some characters and it so happens that these may be required in the Fax initialization strings. In particular the "." (period) is translated into a comma by BinkleyTerm unless you precede the period with a backslash . For example some modems want "AT+FCLASS=2.0" as part of their initialization, but this needs to be entered in the configuration as "AT+FCLASS=2\.0" to prevent BinkleyTerm translating the . (period) into a comma.

The characters that BinkleyTerm translates are listed in the Dial Translation section on Page 22

The command strings vary from Class to Class. There is very good documentation on Fax in the docs of BGFAX (a shareware program which is available from many sources).

AT+FCLASS=?

All Classes. Requests Class information (see above)

AT+FAA=1

Class 2.0 command. Put modem in adaptive answer mode.

Modem will then issue it's fax response string (see below) when the connection is fully established.

Some modems may respond earlier with "FAX" when they hear a fax calling tone.

Some modems forget the +FAA command once another command has been issued to the modem so it is advisable to put the command in the answer string (i.e., use "AT+FAA=1;A" instead of "ATA").

AT+FAE=1

Class 1 command. Enable adaptive answering if the modem is capable of this.

AT+FCR=1

Class 2 and Class 2.0 command Gives the modem permission to take faxes. It is not actually needed in Class 2.0 as it is a default setting in that class.

AT+FLID="xxx-xxx-xxxx"

Class 1 command. Sets fax ID string. Max 20 characters. should only be space, +, or 0 through 9.

Letters should not be used and may cause problems with some fax machines.

AT+FLI="xxx-xxx-xxxx"

Class 2 and Class 2.0 command. Same rules as for +FLI above.

AT+FDCC=1,3,0,2,0,0,0,0

Class 2 command. Sets the resolution, fax speed, compression type etc. Can be shortened to

+FNR=1,3. Using 1,5 would allow 14.4 fax rate working but many fax modems and most standalone fax machines only work well at 9600 so 1,3 is preferred.

The command is not needed on either Class 1 or Class 2.0 modems.

When commands are "stacked" one after the other this command must come at the end of the stack.

AT+FNR=1,1,1,1

Class 2.0 command. Report details of the current call to a rear end fax program, such as BGFAX.

AT&B1+FCLASS=6

ZyXEL modem command. Fully explained in the ZyXEL manual.

AT+FBO=1

This toggles the bit order of a received Fax between "Direct Bit Order" and "Reversed Bit Order" See Section below about "Handling faxes internally with BinkleyTerm"

Modem Fax Response String

Some modems return "FAX" as soon as they receive a fax CNG tone, and follow this with another response when the connection is fully established. Most modems simply return a single string which must be made known to BinkleyTerm.

Known response strings are:

ZyXEL : Connect FAX followed by ZyXEL

Some Supra's FAX followed by +FCON

Most modems have a single response, for example:

FAX Hayes, Supra, Zoom and most Rockwell based chipsets

+FCON Intel, GVC, PPI and some other class 2 modems

+FCO USR v everything and other class 2.0 modems.

Handling the incoming faxes internally with BinkleyTerm

BinkleyTerm can handle the fax reception internally producing a "Raw fax" file

To do it this way, include a Fax Directory in your Binkley.cfg file, e.g.,

```
FaxInDir C:\Bink\Fax
```

The fax, as received, is in a Raw Fax format which is not viewable so you would then normally include an event in your Binkley.evt file telling BinkleyTerm to exit after receiving a fax (using the EF=nn flag) and pick up the Errorlevel nn to go to a section of your batch file which can process the raw fax data.

When fax is received internally the program outputs raw fax files from all three classes of modem (Classes 1,2 and 2.0) in Reverse bit order whereas in fact the output from Class 1 and class 2.0 modems should be in Direct bit order. To correct this, put an instruction "+FBO=1" in the configuration file for class 1 and class 2.0 modems. Note that this only applies to INTERNAL fax reception.

If you wish, you can read the raw fax with the View.exe program which comes with BGFAX (BGFAX is one of the standalone programs which are available).

Handling Fax Reception with a Rear-end program

Rear end fax reception is possible using a spawned child process.

With either operating system, arrange command or batch files so that the handover takes place as quickly as possible. Some users advocate the use of a Vdisk for faster working but this has not been found necessary using the Extern Spawn method with a 386DX40 machine.

The configuration file is arranged much as for Internal FAX reception except that the "FaxInDir" statement must be omitted or commented out as this acts as the semaphore to BinkleyTerm instructing it to use internal reception. Either one or two additional statements should then be added. These are:

```
Extern Spawn  
ExtrnMail <chosen errorlevel> <Modems fax Response string>
```

Used together, these two statements act so that reception of the modem's fax response string causes execution of the command "SPAWNBBBS" as a child or shell process. You write the SPAWNBBBS.COM (or .BAT if you try it in DOS or Win32). Before it is executed, SPAWNBBBS is passed 5 parameters by BinkleyTerm. and you can utilise any or all of these within the batch file.

In the case of BGFAX that program needs the comm handle (port in DOS) passed as parameter 3 (%3) of the SPAWNBBBS command

If the ExtrnMail is used alone, without Extern Spawn, receipt of the modem response string causes BinkleyTerm to exit to the start-up batch file where the errorlevel used in the ExtrnMail statement can be trapped and used to go to a further batch file which would start the rear end fax program.

Fax Transmission

BinkleyTerm does not handle this but it should be simple to arrange a Function key exit with a suitable errorlevel which can start up a standalone program to send the fax and then re-cycle to BinkleyTerm.

DOMAIN ADDRESSING

When is it needed?

Domain Addressing is only needed if you operate in two or more Fido Technology Networks (FTNs) using the same Zone numbers, or you wish to keep different Domains' compiled nodelists separate.

What is it?

Domain addressing is an extended method of designating various FTNs as compared with the zone-only method previously used. Domain addressing adds an additional "layer" to address designation that represents a particular FTN. Within that FTN, zones and nets can be specified without conflicting with identical zones and nets in other FTNs.

How is it used?

To use domain addressing you would:

- 1) Add a domain string to your address statements.
- 2) Include the domain statement in the configuration file with its required parameters, i.e.,

```
Domain <Designator><Abbreviation>[<Nodelist>]
```

<Designator> is the actual Domain name which can be up to about 30 characters long.

Examples:

```
"Fidonet.org" mirrors Internet use
("FidoNet" is a sufficient FTN domain name).
"Alternet.ftn"
"Eggnet".
```

<Abbreviation> is the shortened form of the domain name with a limit of 8 characters. BinkleyTerm uses this to decide which directory to use for non-primary domains.

Examples: "FidoNet", "Alternet", "Eggnet"

<Nodelist> is an optional parameter indicating which compiled nodelist to use. The default is to use the same name as <abbreviation>.

Example: with a V7 FidoNet nodelist you use "Nodex". (V6 used "Nodelist".)

Each FTN can have its own nodelist, but if the zone and net numbers do not overlap, it is possible to combine all the Nodelists, in which case the Domain statements should all reference the combined Nodelist/s.

- 3) Include DomainKludge statements:

These enable BinkleyTerm to fill in a domain if addresses are without domain specification, either from local entry or in FidoNet handshaking.

If no DomainKludge specification exists for any zone, it is assumed to be part of your primary domain. Thus, with a FidoNet primary address, there is no need to specify DomainKludges for Zones 1 to 6.

The "DomainKludge" lines must follow the "Domain" lines, and if you set a DomainKludge without having previously defined a domain, it will not be processed.

REMEMBER, DOMAIN KLUDGE LINES *MUST* FOLLOW THE DOMAIN LINES!

How should Outbound Areas be Named when domains are used ?

As always, the outbound area for your primary address (including domain) is given with the 'Hold' statement.

Separate Outbound Areas are needed for each Zone in each Domain These take an identical stem path to the primary outbound, except that the name of the last sub-directory is changed to the <abbreviation> parameter, plus the zone extension.

For example, if your 'Hold' statement designates C:\BINKLEY\OUTBOUND for the outbound holding area (and you are in FidoNet), Alternet (zone 7) outbound mail would be held in the C:\BINKLEY\ALTERNET.007 directory instead. Note that outbound areas for domains other than your primary will ALWAYS have a zone extension, and that zone extensions are always specified in Hexadecimal, up to .FFF (4095)

As an example, assume that your BINKLEY.CFG includes the statements:

```
Hold c:\bink\outbound
Address 1:104/501@fidonet.org
Address 89:555/66@Alternet.ftn
Address 99:444.33@eggnet
Domain fidonet.org fidonet nodex
Domain alternet.ftn alternet anetlist
Domain eggnet.ftn eggnet egglist
```

* note: if all the nodelists were combined then the domain statements could be:

```
Domain fidonet.org fidonet nodex
Domain alternet.ftn alternet nodex
Domain eggnet.ftn eggnet nodex
```

* If used, DomainKludge statements must come after Domain statements.

```
DomainKludge 7 alternet.ftn
DomainKludge 99 eggnet
```

What else needs to be done?

You need to tell your packer that you have different Outbounds for the Zones in your other Domains. For example, using "Squish" as your packer, this would mean adding extra Outbound keywords in Squish.cfg., as follows:

```
Outbound C:\bink\outbound ; default outbound primary zone
Outbound C:\bink\Alternet.89
Outbound C:\bink\Eggnet .99
```

Note the zone no. parameter required for outbounds other than the primary.

The outbound holding areas (for Zone 1 FidoNet) would then be as follows:

c:\bink\outbound	(Default Outbound)
c:\bink\outbound.002	(FidoNet Zone 2)
c:\bink\outbound.003	(FidoNet Zone 3)
c:\bink\outbound.004	(FidoNet Zone 4)
c:\bink\outbound.005	(FidoNet Zone 5)
c:\bink\outbound.006	(FidoNet Zone 6)
c:\bink\alternet.007	(Alternet Zone 7)
c:\bink\alternet.059	(Alternet Zone 89)
c:\bink\eggnet.063	(Eggnet Zone 99)

If you are updating an older Configuration you should beware that the unofficial "EE" version used reversed zone and domain name references and also accepted zone number ranges after the domain

statement. BinkleyTerm 2.60 ignores any extra zone numbers but does not like the reversed references!

DIAL TRANSLATION

When BinkleyTerm is in modem command mode, several dial translations take place automatically.

ASCII	Char.	Name	Action
----	----	----	-----
032		Space	Stripped
045	-	Hyphen	Stripped
046	.	Period	Translated to a Comma
094	^	Caret	Raise DTR Line
096	`	Backquote	1/20th Second Delay
118	v	Lower Case V	Lower DTR Line
124		Pipe, Split Bar	Carriage Return Sent
126	~	Tilde	1 Second Delay

A comma is not translated but is handled by your modem settings, usually a preset delay of 1-8 seconds (must not be less than 4 secs in the UK). Note, a comma does not produce a delay in BinkleyTerm, only in the MODEM i.e., as part of a dial string. For example, to delay sending an answer command you would use "~ATA" and not ",ATA"

The escape character "\" can be used to escape (i.e. send unchanged) any subsequent character that otherwise would be interpreted by BinkleyTerm's dialer; for example "\" to send a space, "." to send a period. To send the "\" backslash character itself, use "\\".

FLAGS OR SEMAPHORE FILES

BinkleyTerm has built in support for flag files, which are also known as semaphore files.

BinkleyTerm's flag support is mainly intended for multi node or multi line use, i.e. to prevent two nodes of the same system both calling another system at the same time, but flags can also be useful in multi-tasking situations.

To enable flag support:

A. The Flags <directory> statement in your BINKLEY.CFG file should be enabled and point to an existing directory. This directory:

- a) Holds the "TASK.#" file. see below.
- b) Is the fallback directory for the *.BSY files mentioned below.
- c) Is used for the BTRESCAN.yy and BTEXTITxx.yy flags

Nothing to do with flags, but the flags directory is also the "staging area" for the PktRsp statement.

PktRsp won't work without a flags directory.

B. The TaskNumber <#> must be enabled and set to a different number for each node's BINKLEY.CFG. and each task number must be greater than zero.

Note however that with 'BTEXTIT' or 'BTRESCAN' methods mentioned below, the task number may be zero for single line systems that do not use the TaskNumber statement.

How are the directory and the task numbers used?

1. BinkleyTerm creates a "TASK.#" file (where # is the number stated in the TaskNumber statement) in the flags directory whenever BinkleyTerm has carrier and a mail session is in progress. This can be used to determine if any given BinkleyTerm task is currently running.
2. BinkleyTerm will also create a file named "xxxxyyyy.BSY" whenever it is transacting business with a system, for each AKA address presented by that system. In each case xxxx is the net number in hex of that system and yyyy is the node number of the system, also in hex.

The file is placed in the outbound directory for the Domain and Zone in question or, if there is no such directory, then in the flags directory if one has been set up.

These files serve as flags to other tasks or compatible programs so that the other tasks can be instructed not to handle any mail for that node (or its AKAs) until the *.BSY flags have been cleared.

Additionally, BinkleyTerm will not send any mail to a node for which there is a .BSY flag.

Note that inbound FTS-0001 sessions will have to accept a packet before the flag can be tested, as the packet is used to indicate the address of the system.

3. An external program (for example, one of your batch or .cmd files) can place a flag in the flags directory. This flag should be in the form:

```
BTEXTIT££.@@
```

(where ££ is an exit errorlevel in hex and @@ is the task number, also in hex)

BinkleyTerm scans the flags directory frequently and if it finds such a file with the task number matching that in the BINKLEY.CFG file, then it will exit with the errorlevel specified in the flag file. for example:

```
BTEXTIT10.B0
```

will make a BinkleyTerm with tasknumber 176 exit with errorlevel 16

```
BTEXTIT20.00
```

will make a BINKLEYTERM with no task number exit with errorlevel 32

4. In a similar way to that shown above an outbound mail rescan can be forced from an external program by creating a file in the flags directory in the form:

```
BTRESCAN.@@ (where @@ is the tasknumber in hex)
```

for example:

```
BTRESCAN.40
```

will cause a BinkleyTerm with task number of 64 to rescan its outbound areas.

5. If the forcexit <errorlevel> statement in BINKLEY.CFG has been enabled BinkleyTerm will also periodically check the flags directory for a file called FORCEEXIT (or FORCEEXIT.@@ if a TaskNumber is set) When found BinkleyTerm will delete the file and exit with the errorlevel specified in the Forcexit statement.

BI-DIRECTIONAL PROTOCOLS

Bi-directional protocols were introduced in version 2.40 of BinkleyTerm, using "JANUS", a protocol developed by Rick Huebner.

Janus is a bi-directional, full-duplex protocol, and generally requires a full-duplex modem to work properly. Under ideal conditions, Janus allows you to transfer mail simultaneously in both directions with Zmodem- like performance.

Hydra is another bi-directional protocol developed by Arjen Lentz and Joaquim Homrighausen. Support for this protocol has been added in the DOS and OS2 versions of BinkleyTerm 2.60

Naming of Bi-Directional Configuration file statements

Originally, Janus was the only protocol supported and the configuration file statements used to control its operation were named JanusBaud and JanusOK.

Now that there is support for two bi-directional protocols, the old statement names are not entirely appropriate (though they may still be used if you wish) and new names, BiDiBaud and BiDiOK are often used in place of JanusBaud and JanusOK.

BiDiBaud is synonymous with JanusBaud. Use whichever you prefer

BiDiOK is synonymous with JanusOK. Use whichever you prefer

For simplicity, within this section of the documentation reference will be made to the BiDiBaud and BiDiOK statements, but remember that in each case, the equivalent "Janus" statements (JanusBaud and JanusOK respectively) could be used

Two additional statements are available:

NoJanus to explicitly disable Janus
NoHydra to explicitly disable Hydra

Note that if the other system with whom you are working offers both bi-directional protocols, Janus will take priority unless you specifically disable it.

THE JANUS MAIL PROTOCOL

NOTE: There is no assurance or guarantee that Janus will work for you in your environment. PLEASE READ THIS SECTION IN ITS ENTIRETY IN ORDER TO UNDERSTAND AND PROPERLY IMPLEMENT JANUS. FAILURE TO DO SO WILL YIELD UNPREDICTABLE RESULTS!

An Introduction to Janus

Janus is a file transfer protocol specifically designed for WaZOO mail transfers. It can achieve Zmodem-like performance and reliability, with the major added benefit of being able to do transfers in both directions simultaneously. It is implemented as dual independent state machines; one for sending files, and one for receiving files. Both state machines run simultaneously, sharing the phone line. All data is exchanged in variable-length packets, with either a 16 or 32 bit CRC. Under ideal conditions, you can expect Zmodem-class performance going both ways at the same time.

The bottom line is that when two machines exchange mail using Janus, the smaller of the two nodes' batches of files will be sent for free while the larger batch is being sent. If you poll your EchoMail feed nightly, and usually only send a couple of kilobytes while picking up a couple hundred, you

won't get any real benefit from using Janus. But if you normally send 100kb and pick-up 200kb, you'll see significant time and money savings.

However, even if you don't usually need to send and receive much data at once, and don't get much benefit from the full-duplex file transfers, you should get Zmodem-level performance even on one-way transfers, and so shouldn't have any penalty for using Janus rather than ZedZap (Zmodem). The idea is for the benefits to be there when you need them, without costing you anything extra if you don't.

What You Should Be Aware Of

There are some important considerations that you should be aware of before using Janus. Most importantly, Janus is a FULL DUPLEX protocol, and works best with a full-duplex connection.

It's not a good idea to try and shove streaming, non-stop, full-duplex data through a half-duplex connection, since the modems will end up constantly flip-flopping the line around and retraining, and you'll get terrible performance.

What this means is that normal low speed (1200 and 2400 baud) connections should work fine with Janus, and that high speed links that use V.32 should also work fine. High speed connections with high speed in one direction and a low speed back channel (such as an HST) will not allow Janus to work well at all.

There are some other situations where Janus will not work very well. Many systems, modems and networks just don't seem to have the sustained bandwidth to allow full-duplex streaming data to be transferred accurately.

Telenet's PC Pursuit service is a good example. The service has great difficulty with Janus, frequently dropping and mangling data. Janus simply saturates the capabilities of the packet switched connection.

Some modems and even some serial port hardware doesn't seem to bear up very well under the tremendous load Janus can put on them either. This is probably to be expected, since full-duplex streaming protocols are rare, and the hardware has probably never been tested under this type of extreme, sustained load.

The bottom line is that Janus may or may not work for you with your hardware, and if it does work, it may not work well on all connections. Its implementation in BinkleyTerm has been tested thoroughly, and works very well in many cases. You'll need to enable Janus (by default, Janus is not enabled) and test it yourself in your own environment to know whether Janus is a workable solution for you.

Configuring Janus

DOS Fossil driver buffer configuration.

Your first consideration before using Janus is your FOSSIL driver buffer configuration. Janus really needs at least a 2kb receive buffer (or better yet, 3kb) and no more than about a 1kb transmit buffer (including whatever transmit buffering your modem does).

The reason for the large receive buffer may be obvious; the largest packet Janus ever sends is 2kb, and while it's sending 2kb, you can obviously receive 2kb, which Janus won't be able to read from the FOSSIL until it's done sending. So up to 2kb of data can easily be received before Janus gets a chance to read any of it, and possibly a bit more in some situations. If your receive buffer is too small, you'll constantly lose incoming data and require massive numbers of retransmissions.

The send buffer should be kept fairly small, because of the way the packets for the two directions are interleaved on the phone lines. If I'm receiving a large file from you while I'm sending you lots of small files, every time I finish sending you a small file I have to wait until I've received your entire send buffer's worth of data before I see the acknowledgment that says you received my file okay. If your send buffer is more than 1 or 2kb, this can end up wasting a lot of time.

Configuration File Statements for Bi-directional protocols

BiDiBaud <max_connect_rate>

IF YOU HAVE A LOW SPEED MODEM (2400 baud or under), then you will generally use the 'BiDiBaud' statement in your configuration file. Set 'BiDiBaud' to your highest supported baud rate.

IF YOU HAVE A HIGH SPEED MODEM (9600 baud or above), 'BiDiBaud' should be set to the highest FULL DUPLEX baud rate you support.

Note: The maximum setting for <max_connect_rate> is no longer limited to 32767, as it was in earlier versions.

If your high speed modem is single-standard, then the 'BiDiBaud' statement is all you will probably need to use.

BiDiOK <extended_connect_info>

Multi-standard modems require 'BiDiOK' statement(s) in order to take full advantage of high speed full duplex (V.32) connections when they occur.

Your multi-standard modem should be set to return extended modem connect information. This information is appended to the end of the modem connect message, and can be used to determine what type of connection has been made in conjunction with the 'BiDiOK' statement.

In the case of a USRobotics HST Dual Standard, you will normally want to use a bi-directional transfer on V.32 connects, but not on HST connects. For example, these connect strings would indicate connections where a bi-directional protocol can be used:

```
CONNECT 9600/Arq/V32
CONNECT 9600/V32
```

But a connection like this would not take advantage of a bi-directional protocol :

```
CONNECT 9600/Arq/Hst
```

NOTE: The extended connect information is not shown in all upper case, since BinkleyTerm will convert everything but the first character to lower case for display, automatically.

You would permit a bi-directional protocol only on the desired connects, like this:

```
BiDiOK /Arq/V
BiDiOK /V
```

(you may not wish to use a bi-directional protocol on a non-secure connect, if not then omit the BiDiOK /V line

When using 'BiDiOK', make certain that you also set 'BiDiBaud' no higher than 2400, or your 'BiDiOK' statements may not have the desired effect.

In addition, you will probably want to dial out in V.32 mode if you frequently connect with multi-standard modems of the same make and model. This will allow a bi-directional protocol to be enabled on the maximum number of connections.

If your multi-standard modem is not an HST Dual Standard, consult your modem manual for details extended connect information for your particular modem make and model.

NOTE: In testing, it has been found that the "PEP" mode of the Telebit Trailblazer series modems can handle the bi-directional protocols, even though PEP mode is not full duplex. Throughput may not be as high as it would be on a true full duplex connection.

HYDRA MAIL PROTOCOL

This is fully described in the documentation available with the Hydra program.

"Hydra (mythology), in Greek mythology, nine-headed monster that dwelled in a marsh near Lerna, Greece. A menace to all of Argos, it had fatally poisonous breath and when one head was severed, grew two in its place; its central head was immortal. Hercules, sent to kill the serpent as the second of his 12 labors, succeeded in slaying it by burning off the eight mortal heads and burying the ninth, immortal head under a huge rock".

"The term hydra is commonly applied to any complex situation or problem that continually poses compounded difficulties."

"Hydra (mythology)," Microsoft (R) Encarta.
Copyright (c) 1993 Microsoft Corporation.
Copyright (c) 1993 Funk & Wagnall's Corporation"

Hydra is only supported in the OS/2 and DOS versions of BinkleyTerm at present. No doubt this will change shortly.

CALL COSTING

BinkleyTerm allows two call costing systems to be used.

The American Costing System

This is the original system used in earlier versions and it is the default unless you enable the 'Eurocost' statement. Using the US system, the Nodelist "cost" field is set to indicate the cost of a call in cents per minute.

The Eurocost System

In Europe calls are priced by "call units", the current cost of a unit in the UK being 4.935 pence. Each unit buys you a certain amount of time, the time depending on whether the call is a Local, Long Distance (within the country) or International call.

The time per unit also depends on the time of day. i.e., whether Cheap /standard /peak rates apply. Binkley cannot yet handle the variations of cheap/standard and peak rates but you can set up your event file to call out only at times when cheap rates apply.

To enable the European Costings put the following three statements in your BINKLEY.CFG:

EuroCost	This will establish the revised costing method.
CostLog <filespec>	Provides a very neat log of call costs (choose a filename).
CostUnit <number>	Defines the cost of one call unit in your currency. Cost is just a number which can represent pence, pfennigs or anything else (in the UK where a unit costs 4.935 pence use 5).

To make this work BinkleyTerm expects new information in your NODELIST. It will look at the "Cost" fields and expect to find the *TIME* allowed for one unit, this time being expressed in tenths of a second. This means altering the configuration file of your nodelist compiler and recompiling the list, but the alterations are not difficult. Just bear in mind that BinkleyTerm will expect a "time per unit" wherever it you would previously have put a "cost per second".

If you previously entered the call cost in cents (for example this is how Fastlst and ParseLst want the information) then enter the time allowed for one unit measured in tenths of a second. (Example.. a local call in the UK at cheap rate gives you 220 seconds per unit. You would enter 2200 in the COST fields)

If you previously entered the call cost in dollars (as BONK expects), then you would enter 22.00 for a UK cheap rate call.

Note: You **will** need to revise your event file because European calls get cheaper as the L parameter gets larger! Refer to the L flag under Event File for more information.

EMSI

EMSI is a handshake protocol credited to Joaquim H. Homrighausen which allows maximum flexibility in session startup and control.

To enable EMSI you will need to ensure that there is no NOEMSI statement in the configuration file (or that it is commented out) and that all four of the following statements are included:

```
MyListFlags MyLocation MyMaxBaud MyPhone
```

Note: This information should be given in Nodelist form as it appears that other mailers may refuse to connect if the items in the MyListFlags entry are not valid nodelist flags.

THE CONFIGURATION FILE

The BinkleyTerm configuration file, by default named BINKLEY.CFG, is the place where you communicate information about your system to BinkleyTerm.

LAYOUT

Statements are made one per line, with any necessary parameters. Statements are not case sensitive. The keyword of the statement must start in column one, i.e. if there is a blank at the start of the line BinkleyTerm will report "illegal statement".

Any line having a semi-colon (;) as its first character is deemed to be a comment line. A sample configuration file comes with the BinkleyTerm distribution package.

FULL ALPHABETICAL LIST OF CONFIGURATION STATEMENTS

About <filespec>

This tells BinkleyTerm the name of the ABOUT file, a special file used with incoming file requests. <filespec> is a complete drive, path and filename designation. Refer to the section "Controlling File Requests" for more information.

Address [**<zone>**]:<net>/<node>[.<point>][@<domain>]

This statement (or multiple statements) designates the network address(es) of your system. Although the <zone>, <point> and <domain> parameters are optional, it is recommended that the <zone> parameter ALWAYS be used.

Example of address for node 1:104/501

```
Address 1:104/501 (or 1:104/501.0)
```

Example of address for point off 1:104/501

```
Address 1:104/501.11
```

Example of address with domain for 1:104/501.11

```
Address 1:104/501.11@fidonet.org
```

If you designate a Domain in the 'Address' statement then you **must** also create a corresponding 'Domain' statement.

When the Address statement is used, the older style addressing, such as 'Aka', 'Point' and 'Zone' should NOT be used, nor should 'Boss' be used by point systems (the address statement covers this).

Multiple 'Address' statements, each with a different <zone> parameter, may be used. This allows BinkleyTerm to identify itself differently to different zones, thereby making multi-net operation somewhat more simple. You may have up to 25 address statements including the primary one.

Note that if you are connected to a zone for which an 'Address' statement does not exist, that the 'Address' statement that appears first in the configuration file will be used as a default.

The zone given in the 'Address' statement that appears first determines your "default" outbound area, as given by the 'Hold' statement.

Mail for all other zones is stored in distinct outbound areas for each zone.

AfterCall <number> <modem_string>

If the Configuration file contains this statement (in which <number> is the number of responses expected from the modem and <modem_string> is an AT command) then BinkleyTerm will send the string to the modem after any call, incoming or outgoing, and will note up to <number> of response lines in the log file.

For example, with a USR Courier or an ATI PM14400FXMT
"AfterCall 16 AT16|" works well.

AfterMail <command_line>

If used, BinkleyTerm will invoke a Command Shell and execute the <command_line> after receiving mail. It is suggested that <command_line> designate a batch file, rather than a specific program. The batch file would contain command line(s) for the program(s) that will actually unpack and/or toss incoming mail.

NOTE! If this statement is used, no E2= or E3= exits during event schedules should be used, since they take priority over this statement. Refer to the section "Event Files" for details.

See 'Packer <command_line>' on page 45 and 'Cleanup <command_line>' on page 33 for related information.

Aka <net>/<node>

[This Statement is Obsolete]

NOTE: This statement is supported for backward compatibility only. To specify alternate addresses, use multiple 'Address' statements as described previously.

AltNumber <primary number> <alternate number>

This statement defines an alternate number which BinkleyTerm will dial if a call to the <primary number> was unsuccessful.

The alternate number must have the same node number as the <primary number> or have the same node number as an aka.

If a system you frequently call has 2 or more nodes that share the same node number, you can now call the alternate numbers if the primary number (nodelisted number) is busy or doesn't answer.

For Example:

```
AltNumber 555-1111 555-2222
```

If 555-1111 was busy, BinkleyTerm would automatically call 555-2222. If there was more than one alternate number they would be specified as follows:

```
AltNumber 555-1111 555-2222
AltNumber 555-1111 555-3333
```

As in the first example, if 555-1111 was busy, a call would be made to 555-2222. If that number was busy, BinkleyTerm would attempt to call 555-3333.

This type of setup, multiple nodes sharing the same node number, or having the same node number as an aka, is most often used by, but not limited to, mail distributors. The only limitation of the number of AltNumber statements you can specify is the amount of memory you have.

If the various modemxxxx statements (see "Modem result code array") are NOT included in your configuration file, then BinkleyTerm defaults to 'ModemRetry BUSY' and alternate dialing will occur if a BUSY response is received from the main number.

If the Modemxxxx statements are in your config, you must include 'ModemRetry BUSY' to make busy responses eligible for alternate number dialing.

Answer <modem_string>

When this statement is used, BinkleyTerm assumes that the modem has been set NOT to answer the phone automatically (by the modem initialization string, or the modem's DIP switches).

When BinkleyTerm receives a response string of "RING" from the modem, it sends the <modem_string> command to the modem to answer the phone.

The advantage is that BinkleyTerm must be "alive and well" before the modem will answer a call. If for some reason BinkleyTerm is not available, yet the modem still has power, no calls will be answered.

NOTE! Some modems DO NOT like commands to be sent while they're sending response strings, similarly, some modems prefer command echo off (ATE0). In testing, this feature DOES NOT work on all modems. Only by trying it will you be able to determine if it works with your modem.

AnswerBack <text> [Terminal Mode Only]

In Terminal Mode, when an ENQ (ASCII decimal 5, hex 5) is received, <text> will be sent in response. Normal BinkleyTerm character translations are available. Many BBS packages send this character immediately prior to requesting a user name.

Application <app_name> [<parm> <parm> ...]

Allows addition of application dependent data to the configuration file. Any 'Application' statement is ignored by BinkleyTerm entirely. <app_name> is the name of or reference to a specific application, such as a message editor or outbound maintenance utility that uses the BinkleyTerm configuration file. Zero or more application specific parameters, shown as <parm> in the example, may follow <app_name>.

Autobaud

In Unattended Mode, provided the Fossil is not locked, this forces BinkleyTerm to call out at the baud rate specified by the 'Baud' statement, regardless of the baud rate associated with a given nodelist entry. This assures connects at the highest possible baud rate. If the Fossil is locked, BinkleyTerm cannot change the rate.

Avail <filespec>

This designates the name of the file to be sent to a remote system that file requests "FILES" from your system. The <filespec> identifies the file, and may contain an optional drive and path designation.

Banner <string>

The line designated by <string> is sent to callers immediately following the BinkleyTerm identification line, and before the line, "Press <Escape> to enter BBS." Generally, this is the name of your BBS or something else of interest to callers.

Baud <max_baud_rate>

In DOS, valid <max_baud_rate> settings are 300, 1200, 2400, 4800, 9600, 19,200 and 38,400. but the 'ExtBaudRates' statement described on page 37 can be used to add higher rates when used in conjunction with Ray Gwinn's X00 FOSSIL Driver v1.53a.

The Baud statement does not support the "three-quarter rates" i.e. 7200, 12000 and 14400. Set your modem to a higher full rate e.g. 19200 or 38400.

In OS/2 and Win32 systems, 57600 baud and 115200 baud can also be set. Note that for this to work with the OS/2 version, you must have version 2.5 or above MAXCOMM.DLL or any version of SIOCOMM.DLL.

BBS <exit_option>

This designates the method to be used to access your BBS software when a human caller dials your system. Valid options for <exit_option> are 'Batch,' 'Exit' and 'Spawn.' Refer to the "BBS Interface" section for more information on the options, and how to use them.

BBSNote <string>

After a human caller presses Escape to access the BBS program, or after the number of seconds designated by the 'Timeout' statement, BinkleyTerm will display the <string> to the caller. Generally, this is a notification that the BBS software is loading.

BBSSound <filename.wav>

Play a sound on exit to a BBS (OS/2 and Win32 machines only at present)

BiDiBaud <max_connect_rate>

This will enable both bi-directional protocols (Janus and Hydra) and set the maximum baud rate to be used with such transfers.

The action of this statement may be modified by one or more BiDiOK statements, q.v..

Note: The <max_connect_rate> is not limited to 32767 as it was in earlier versions of BinkleyTerm.

See also 'NoHydra', 'NoJanus' and the details in the "Bi-Directional Protocol" section on page 24

'BiDiBaud' has exactly the same action as 'JanusBaud'

BiDiOK <extended_connect-info>

Modifies the action of the 'BiDiBaud' statement by only allowing bi-directional transfers for those connections where the modem returns extended connect information which matches the <extended-connect-info> parameter.

The BiDiOK statement is not required with a single standard modem.

A multi-standard modem should be set to return extended modem connect information appended to the end of the modem CONNECT message. If the modem returned CONNECT 9600/Arq/V32 then a statement: BiDiOK /Arq/V32 will enable the bi-directional protocols for that connection but the protocol would not be enabled if the Modem returned CONNECT 9600/Arq/Hst as the strings do not match.

It is thus possible to differentiate between HST and other connections.

NOTE: The extended connect information is not shown in all upper case, since BinkleyTerm will convert everything but the first character to lower case for display, automatically.

Multiple BiDiOK statements may be used to cover the various modem result strings where Bi-directional transfers are to be allowed.

See also 'NoHydra', 'NoJanus' and the details in the "Bi-Directional Protocol" section on page 24

'BiDiOK' has exactly the same action as 'JanusOK'

BlankWait <number>

Sets the number of seconds BinkleyTerm will wait before blanking the screen when the 'ScreenBlank' configuration statement is enabled.

Boss <net>/<node>

[This Statement is Obsolete]

NOTE: This statement is supported for backward compatibility only. It is not needed when the 'Address' statement is used, as described previously.

This specifies a FidoNet node address. For regular FidoNet nodes, place your assigned address here. For Point systems, place the address of your Boss node here.

BossPhone <phone_number>

This statement, used in Point installations, contains the telephone number of your Boss node for use with the Alt-Y command in Terminal Mode. This statement is optional in all cases except Point installations that do not wish to use a nodelist.

BossPwd <password>

Used for Point installations, this statement designates the password to be used for session-level passwording with your Boss node. Refer to the section "Session Passwords" for additional information. The Boss node must also have session passwording implemented, using this password.

When this statement AND 'BossPhone' are BOTH implemented, a nodelist is NOT required for a Point system.

BoxType <number>

When full-screen mode is used (default), this tells BinkleyTerm what type of boxes to use for the various on- screen windows. Legal values are from 0 to 4. They produce the following results:

- 0 = Hatches (Non-IBM)
- 1 = Single Rule
- 2 = Double Rule
- 3 = Single Top, Double Sides
- 4 = Double Top, Single Sides

Busy <modem_string>

While BinkleyTerm performs certain functions, and when you exit the program from Unattended Mode, BinkleyTerm sends the <modem_string> to the modem. Normally, this is a short set of modem commands, as shown in the sample configuration file, to take the phone off-hook to prevent incoming calls. Callers will hear a busy signal.

If you lower DTR instead, using a lower-case letter "v", this would cause callers to hear ringing, but with no answer.

CaptureFile <filespec>

[Terminal Mode Only]

If used, this statement tells BinkleyTerm the name of a file to use for session capturing in terminal mode. The Alt-L command toggles session capture on and off. If this statement is not used, then BinkleyTerm will prompt for a name each time Alt-L is pressed. When activated, all communications session I/O will be echoed to this file.

Carrier <hex_carrier_mask>

This tells DOS BinkleyTerm which FOSSIL status bit it should use to determine whether or not carrier is present. A value of 80 (hexadecimal) is nearly always correct. Some modems do not support CD (carrier detect) and other signal lines may be used.

NOTE: This value is in HEXADECIMAL (base 16). Other systems, such as Opus-CBCS, ask for this value in DECIMAL (base 10). Normal setting is 80 hex, which equals 128 decimal.

Cleanup <command_line>

If used, BinkleyTerm will execute <command_line> at the beginning of each event, but prior to the 'Packer' statement's command line (if used). This might be used to unpack any previously packed outbound mail for later repacking, or to perform minor outbound area maintenance, etc. It is suggested that <command_line> designate a batch file that would contain the command line(s) for the program(s) actually used to unpack mail and/or perform maintenance.

See 'AfterMail <command_line>' on page 30 and 'Packer <command_line>' on page 45 for related information.

Colors <brdr> <set> <tday> <pndg> <actvty> <trnsfr> <rev> <popwin>

The first 6 parameters refer to parts of the window display, <rev> is the reverse video used in the pending outbound window and <popwin> refers to the color of pop up windows, e.g., the popup which appears when Alt-G is pressed.

Specify the color you want for each section, for example:

Colors 112 9 10 11 14 12 48 7 (This combination once won a competition)

Remember that you need a Video Fossil installed for color displays if you are running under DOS. See page 59

The default colors are black background with white foreground for all areas.

The following chart may assist you in determining your desired color attribute values. The numbers listed under "Foreground" will yield the given color on a black background. By adding the value shown under "Background Value" to the foreground value will yield a background of the selected color.

For example, to get yellow characters on a blue background, add the foreground color for yellow, 14, to the background value for blue, 16 - use a color attribute of 30. Note that gray, bright colors and yellow cannot be used for the background.

Color	Foreground	Background
-----	-----	-----
Black	0	0
Blue	1	16
Green	2	32
Cyan	3	48
Red	4	64
Magenta	5	80
Brown	6	96
White	7	112
Gray	8	N/A
Bright Blue	9	N/A
Bright Green	1	N/A
Bright Cyan	11	N/A
Bright Red	12	N/A
Bright Magenta	13	N/A
Yellow	14	N/A
Bright White	15	N/A

On IBM-compatible monochrome displays and using a black background, colors 9 through 15 yield high intensity characters, 1 through 7 normal intensity characters. 0 and 8 yield invisible characters. 1 gives underscored white and 9 will yield underscored bright white. You may find that you need to have a Video fossil installed, even with a mono screen, to get these responses. See Page 59

See also the 'Mark_Kromm' statement on page 41 which installs a popular color set without further effort.

The easiest way of all to set the colors to your choice is to obtain one of the available color adjunct programs, for example SBC (a DOS tool), which allows you to set the colors on the fly.

CostLog <filespec>

Include this statement to produce a neat Cost log. When using the Eurocost option, the log has an error field but this is not at present implemented. It is left at zero.

CostUnit <number>

This is for use with the Eurocost option and defines the cost of one call-unit in your local currency. The costings will then be shown in that currency, i.e., if you define the unit cost in pence then the costings (shown as tariff in the log) will represent the cost of that call in pence.

Curmudgeon

NOTE: This statement is not for general use.

If used, this will cause your system to refuse mail connections from any "Unknown" systems, i.e. systems not within your compiled nodelist. Note that in FidoNet, systems using this keyword must have the "LO" flag in their nodelist entry, and run a full-world nodelist (to avoid refusing any valid FidoNet mail connections).

CursorCol <column>

For use with multi-tasking systems, this tells BinkleyTerm the column number to place the cursor in, after screen writes. For DESQview, <column> should be set to 1. The default value is 80.

CursorRow <row>

For use with multi-tasking systems, this tells BinkleyTerm the row number to place the cursor in, after screen writes. For DESQview, <row> should be set to 1. The default value is 23.

Debug

NOTE: This statement is not for general use.

Provided that you set LogLevel to 5 (see "LogLevel <log_level_number>" on page 41), then use of the Debug statement will cause BinkleyTerm to send additional information about the progress of its operations to your log file.

Dial <match_string> <new_prefix>/[<new_suffix>]

This option allows for real-time telephone number translation. BinkleyTerm will look at the telephone number it is to send to the modem. If the prefix of the telephone number matches that shown in <match_string>, then the prefix will be changed to <new_prefix>, and <new_suffix> (optional) will be added to the end.

In most cases, this will be used to strip-off "1-" and/or area codes for systems in a local exchange.

Example:

```
DIAL 1-603-888 888/
```

The result of the above statement will be that a number in the nodelist as 1-603-888-8179 would be changed to simply 888-8179 and dialed.

An AT command to your modem can be included e.g.,

```
DIAL 1-303-555 555/&M0
```

A number in the nodelist as 1-303-555-1234 would be changed to 555-1234 and dialed with the "&M0" being sent to the modem.

There is a maximum of 20 characters each for <match_string>, <new_prefix> and <new_suffix>.

Another use for this feature would be for dialing scripts:

```
DIAL 1-404 "GA_PCP.SCR"404/
```

This line could be used to invoke a PC Pursuit script, for example, for the state of Georgia. The script would be used for all outgoing calls to area code 404.

For permanent translations it is probably more efficient to perform these translations with your nodelist processing program (i.e., Fastlst, ParseLst, XlaxNode). Refer to your nodelist processing software documentation for more information.

The escape character "\" can be used in a phone number to escape subsequent chars that otherwise would be interpreted by BinkleyTerm's dialer. To send the "\" backslash character itself, use "\\".

DoingMail <string>

If used, this statement will cause BinkleyTerm to send <string> to a caller when an event without a "B" flag is active, indicating that BBS access is NOT allowed. This replaces the default string "Processing Mail. Please hang up". See page 51 for more information on event flags.

Domain <Designator> <abbreviation> [<nodelist>]

<Designator> is the actual Domain name. FidoNet would be "Fidonet.org" and "Altnet" would use "altnet.ftn".

<Abbreviation> is the shortened form of the domain name. This is limited to EIGHT (8) characters. In most cases this will be the common name of the network (i.e. "FidoNet" and "Altnet" in the examples).

<nodelist> is an optional parameter which indicates the nodelist to associate with this domain. If not given, the default is to use the same name given as <abbreviation>.

Refer to the "Domain" notes in this Manual for examples

DomainKludge <ZoneNumber> <domainName>

This will fill in a domain if addresses are without domain specification, either from local entry or in FidoNet handshaking.

These lines must follow the "Domain" lines, and if you set a domain kludge without having previously defined a domain, it will not be processed. Refer to the "Domain" notes in this Manual

Downloads <path> [Terminal Mode Only]

This tells BinkleyTerm where to place files downloaded while in Terminal Mode. The <path> is a complete drive and path designation. This path has no effect on mail transfers made in Unattended Mode.

DTRHigh

If used, BinkleyTerm will leave the DTR (data terminal ready) line to the modem "high" whenever it is exiting. By default, BinkleyTerm takes the DTR line "low" when exiting. The use of 'DTRHigh' has no effect when doing an Alt-J shell escape which will leave DTR low. 'DTRHigh' should be used with modems that go back on-hook when DTR is lowered.

EnterBBS <string>

If used, this statement will cause BinkleyTerm to send <string> to a caller when the BBS is available, during events with a "B" flag, indicating that BBS access is allowed. This replaces the default string "Press <Escape> to enter BBS." Note that <string> must not exceed one line. See page 51 for more information on event flags.

ErrLevelShell <errlevel> <shell command>

When <errlevel> matches an exit errorlevel specified in the event file BinkleyTerm will spawn a shell instead of exiting. The spawned shell will run the command specified by <shell command>

For example, if the event file had the following:

```
Event ALL 08:00 12:00 E4=91,TIC A=60 T=3,10
```

...and the configuration file had the following:

```
ErrLevelShell 91 DOTICK
```

Instead of exiting with an errorlevel of 91 when a TIC file is received, DOTICK.CMD (or .BAT) will be executed. If you are using the OS/2 or Win32 version of BinkleyTerm (or the DOS version on Windows 95) and specify:

```
ErrLevelShell 91 Start /I /C "Tick Processing" DOTICK
```

...DOTICK.CMD (or .BAT) will be started as a separate session.

Unlike the exit procedure, where if more than one E4-E9 exit applies the first one is taken, if more than one E4-E9 has a matching ErrLevelShell that applies, then each shell that applies will be taken.

If an E4-E9 exit applies and does not have a matching ErrLevelShell then that exit will occur after the shells have been taken.

For instance, if there was an error level specified for E4 through E9, and if each applied at the end of a session, and an ErrLevelShell was specified for E4, E7, and E8, the 3 shells would be taken, and then BinkleyTerm would exit because of E5.

Note however that if E3 is an ErrLevelShell then any E2 exit will be ignored on the assumption that the E2 (packet) stuff will have been processed by the shell for E3 (which handles compressed and other mail).

Note also that this statement only applies to errorlevels specified *in the event file*. It is not (yet?) available for such things as function key exits, which are not in the event file.

EuroCost

This enables the European style of costing in which the charge for a "call-unit" is fixed and the time allowed for each call-unit varies for local, national and international calls, and also depends on the time of day.

See 'CostLog <filespec>' on page 34, and 'CostUnit <number>' on page 34 or refer to the section on Call Costing for more information.

If Eurocost is not enabled, BinkleyTerm defaults to the original US cost per minute tariff calculation.

Event <event_flags...>

NOTE! Normally the 'Event' statement is used only in the event file, BINKLEY.EVT. Events should NOT be scheduled in the configuration file.

Due to the depth of this topic, it is covered in the section "Event Files and Schedules"

ExtBaudRates

Used by the DOS version of BinkleyTerm only. Under DOS the baud rate cannot normally be set to exceed 38200, but a recent version of the X00 FOSSIL driver (ver 1.53a) supports additional baud rates. This is not yet a standard so BinkleyTerm needs to be told to use this extension by means of the ExtBaudRates statement. BinkleyTerm can then support up to 115200 baud.

Notes:

1. Only use this statement if your FOSSIL supports this function, or you will regret it!
2. Put this statement in your configuration file *BEFORE* any line that deals with baud rates.
3. This statement is not required in OS/2 or Win32 versions, which can directly support 115200 baud.

Extern spawn

This causes each external mail exit to be a "spawn".

Read the comparable 'BBS spawn' description for details.

On spawning BinkleyTerm will execute

```
EXTMAIL.BAT %1 %2 %3 %4 %5 %6
```

The parameters to EXTMAIL are exactly the same as the exit case, so you can find the "errorlevel" from the command line if you need it.

EXTMAIL.BAT is a file that you create which can use the above parameters.

Note: if you enable this option, all external mail is spawned.

ExtrnMail [<errorlevel>] <string>

This is used in conjunction with BinkleyTerm's external mail program feature. Refer to the section "External Mail Programs" for information. The [<errorlevel>] parameter is optional, the default value is 99.

<String> is something sent by the remote system which is to trigger the errorlevel exit. When used for an external mail program, <string> should be relatively long, without too many repeating characters, to assure accuracy. When used with multiple BBS functionality, <string> may be only one letter.

Up to 16 'ExtrnMail' statements that each use [<errorlevel>] may be used.

Note: If the Extern Spawn statement is also used then ALL external mail exits are spawned.

ExtSession <mask> <ProgramName>

Using this statement, BinkleyTerm can have "external mail sessions". <mask> should be a hex mask which corresponds to a "modem type" in the nodelist. When BinkleyTerm attempts to "call" a system, it will use the external mail facility if the modem type exactly matches the mask. BinkleyTerm will then turn off everything (close log files, busy the modem) and, for every file to send to this system (packets, non-zero length attached files), it will call:

```
programname <full-address> <tasknumber> <filename>
```

This designates the name of the response file template used for est response file construction. The <filespeer is defined) and that filename is the full path and filename, in the same case as it appears in the file attach. Binkley only calls programname for listed files that actually exist (not logging missing files).

When BinkleyTerm regains control it will look for a file in the current directory named "programname.tasknumber". If this file exists it will continue the mail session; if this file does not exist it will interpret this as a session failure and will handle it like any other session failure (bad flags, etc.). When the session ends, one way or the other, BinkleyTerm will restart itself and proceed. Example: Assume point 2 on the system is defined as having modem flag 64 (0x40).

(With a private list, I could do this by putting a "UGATE" flag on the node, then with XLAXNODE, "MODEMTRANS 7 UGATE").

create the file "points.bat" containing these commands:

```
echo %0 %1 %2 %3 >> points.log
if "%1%" == "1:343/491.2@fidonet" copy %3 m:\point2
if errorlevel 1 goto end
touch points.%2%
:end
```

Now include the statement 'ExtSession 40 points' in the Binkley.cfg

The result will be that all sessions with point 2 put the mail into m:\point2.

Note that this mechanism is designed mostly for callout. However, you can poll for mail by creating a dummy packet and "sending" it, and your batch file can copy files into your inbound.

EXTSound <filename.wav>

Play a sound on exit to an external (UUCP) mailer (OS/2 and Win32 machines only at present)

FaxBaud <number>

Various fax modems behave differently with respect to baudrate after a fax connection. This version of BinkleyTerm will leave the baudrate alone by default after a fax connect but will set it to a user-specified rate of <number> if 'FaxBaud' is enabled.

FaxInDir <path>

Sets a directory for receipt of FAX. This tells BinkleyTerm to receive incoming Fax using its own internal Fax reception routine. For now we've left the code as is but expect to change to PCX format.

Do NOT include this statement if you intend to use an external program to receive the fax.

FaxSound <filename.wav>

Play a sound on FAX exit (OS/2 and Win32 machines only at present)

FileSec <number>

Part of the implementation of faster file searching using Maximus information. See page 4

FileSound <filename.wav>

Play a sound on E3 or user specified mail exit (OS/2 and Win32 machines only at present)

Flags <path>

Specify the path and directory name to be used to hold task identification files created by BinkleyTerm. The filename is TASK.# where # is the task number. These files can be used to determine if any given BinkleyTerm task is currently running. The files are created whenever BinkleyTerm has carrier present and a mail session is in progress.

ForcExit <errorlevel>

This will cause BinkleyTerm to check in the flags directory periodically for a file called FORCEEXIT or FORCEEXIT.@@ if a task number has been set. (@@ represents the task number in hex). When found BinkleyTerm will delete the file and exit with the errorlevel specified in the ForcExit statement.

For further information refer to the section on Flags and semaphore files in this manual. Refer also to the TaskNumber statement, and the use of the BTEXTxx.yy flag files.

FTS-0001**[Normally Used for Test Purposes Only]**

When used, this statement will force BinkleyTerm to use only base FidoNet protocol (FTS-0001), effectively enabling the equivalent of the following statements: NoWaZOO, NoResync, NoSLO and NoSEALink. Used mostly for debugging and testing of FidoNet policy compliance; not for regular use.

Gong**[Terminal Mode Only]**

This statement is only applicable in Terminal Mode. It causes BinkleyTerm to sound an alarm on connecting with a system it's attempting to dial, or after a download has been completed. 'Gong' also works when doing a manual mail poll operation (Alt-M) in Terminal Mode.

Hold <path>

This specifies the complete drive and path designation for the directory that will be used as your outbound mail holding area. Your mail processing software must place outbound mail in this area for BinkleyTerm to send to or hold for other FidoNet systems.

It should be mentioned that this area should NOT contain any other files of any kind, and that the contents of the directory should NOT be manipulated by you unless you know EXACTLY what you're doing. For all practical purposes, BinkleyTerm maintains this directory for you.

Include <filespec>

If used, this tells BinkleyTerm the name of a file to include while reading the configuration file. The include file must contain additional configuration file statements in the same format as expected for the primary configuration file. When end-of-file is reached, BinkleyTerm will continue reading the main configuration file at the line immediately following the 'Include' statement that initially caused BinkleyTerm to branch.

<filespec> gives the filename and may optionally include a drive and path designation.

Note: Included files may include other files, recursively.

Init <modem_string>

BinkleyTerm sends the <modem_string> to the modem to initialize it, and make it ready for communication. The string is sent to the modem verbatim, with the exception of special dial translation characters. These characters are shown in the section "Dial Translations."

Refer to your modem instruction manual for help in finding a correct init string for your particular modem and configuration.

JanusBaud <max_connect_rate>

This statement is synonymous with 'BiDiBaud', q.v.

JanusOK <extended_connect_info>

This statement is synonymous with 'BiDiOK', q.v.

KnownAbout <filespec>

This tells BinkleyTerm the name of the ABOUT file, a special file used with incoming file requests received from "known" systems. <filespec> is a complete drive, path and filename designation.

KnownAvail <filespec>

This designates the name of the file to be sent to a "known" remote system which file requests "FILES" from you. The <filespec> identifies the file, and may contain an optional drive and path designation.

KnownInbound <path>

Used with secured inbound areas, this statement designates the path to the inbound file area used for mail received from "known" systems.

KnownMaxBytes <number>

See 'MaxBytes <number>' on page 41 for information.

KnownMaxTime <number>

See 'MaxTime <number>' on page 42.

KnownReqLim <number>

Limits the maximum number of files that will be sent to "known" systems in response to incoming file requests during any one mail session. Regardless of whether the incoming requests has wildcards, or whether multiple file requests are sent in one mail session, the maximum number of files that will be sent is <number>.

KnownReqList <filespec>

This designates a file, specified with full drive, path and filename.ext, similar to 'Okfile <filespec>' on page 45 but which holds details of files available to "Known" systems.

KnownReqTpl <filespec>

This designates the name of the template file used to construct response files generated for "known" systems. The <filespec> is a complete drive, path and filename designation.

KnownSec <number>

Optional part of the implementation of faster file searching using Maximus information. See page 4

LineUpdate **[Used for Test Purposes Only]**

Updates BinkleyTerm's log file on a line by line basis, so that if a system crash occurs the logged information will be as up to date as possible.

LockBaud [<lock_rate>]

The FOSSIL should NOT be locked if this option is used.

The <lock_rate> parameter tells BinkleyTerm the baud rate at which locking should occur.

This is useful for modems that allow a floating baud rate up to a certain speed, then are locked at higher connect rates. Most ROM revisions of the USRobotics Courier V Everything allow this type of operation by setting the modem for &B0 which floats the baud rate at 2400 bps or lower. Bit 7 (and perhaps Bit 6) of the S27 register also need to be set appropriately. This would give improved interactive performance on low speed connects.

Example: When using a Courier modem, To LOCK at 38400 and FLOAT at slow speeds, use the 'Baud 38400' and 'LockBaud 4800' statements in your configuration file, and set the modem for &B0 and S27=192.

'LockBaud 0' (or just 'LockBaud') will lock the baud rate at all connect speeds. Even with a locked FOSSIL port, this may be used to have Binkley pass a constant rate (say 38400) as its parameter to the BBSbatch or ExtMail batch files.

Lockbaud <ConnectSuffix>

The <ConnectSuffix> is that part of the connect string from the modem that identifies an error-free connection. The FOSSIL should NOT be locked if this option is used.

The modem must also have the ability to set a floating/locked baud rate.

See the detailed explanation in the "High speed and Error correcting Modem" section of the User Manual.

If you have a modem with more than one response code which indicates an error-free connection, you can use multiple "LockBaud" lines (up to 16).

LogLevel <log_level_number>

This tells BinkleyTerm how verbose to make the status log. Acceptable values for <log_level_number> are from 1 to 5, 1 indicating minimal information, 5 maximum information. See 'Statuslog <filespec>' on page 49 for additional information.

Each log entry is preceded by a symbol or a blank, indicating the importance of the entry.

LogLevel	Symbols That Precede Included Entries
-----	-----
1	! *
2	! * +
3	! * + :
4	! * + : #
5	! * + : # and blank (no character)

Macro <number> <macro_string>**[Terminal Mode Only]**

This allows the sending of predefined macros from within Terminal Mode. Macros are typically used to send your name, user ID, or passwords while on-line.

The <number> parameter is a digit between 1 and 9, which corresponds to the F1 through F9 keys. While in Terminal Mode, you send a macro by pressing Alt-Fx, x indicating the macro number you desire to send.

The <macro_string> is sent verbatim. A carriage return is indicated by the pipe symbol (|). No other translations take place.

MailNote <string>

Used in conjunction with BinkleyTerm's external mail program feature. When the string designated by the 'ExtrnMail' statement is received, <string> is sent to the caller as notification that the external mail program is being loaded. Refer to the "External Mail Programs" section for information.

MailSound <filename.wav>

Play a sound on E2 mail exit (OS/2 and Win32 machines only at present)

Mark_Kromm

Installs a color set judged best in a competition
This uses colors 112, 9, 10, 11, 14 and 12.

MaxAreas <filespec>

Users of MAXIMUS can implement faster searches using their MAXFILES.IDX for file request searches. See Page 4.

MaxBytes <number>

Together with the 'ProtMaxBytes' and 'KnownMaxBytes' statements, you can control file requests by number of bytes sent. These three statements work in the same manner as the other file request limiting statements detailed in the documentation.

MaxPort <quantity>

BinkleyTerm is capable of supporting up to 32 communications ports, far more than any current FOSSIL driver is capable of supporting. Normally the <quantity> will be 1 or 2, depending on your FOSSIL.

Refer to the documentation for your FOSSIL for information about the number of ports it is capable of supporting. For Unattended Mode, the port number in use is set by the 'Port' statement. In Terminal mode, you may change the port in use (dependent on the number of ports your FOSSIL can support and the hardware you have available).

MaxReq <quantity>

Limits the number of files that will be sent during one mail session in response to an incoming file request. Regardless of whether the incoming requests has wildcards, or whether multiple file requests are sent in one mail session, the maximum number of files that will be sent, including any response file, is <number>.

MaxTime <number>

Specifies the maximum time in minutes allowed for this file request session. This statement can be used in combination with the file request size limiters (MaxBytes, KnownMaxBytes, ProtMaxBytes) as well as the file request quantity limiters (MaxReq, KnownMaxReq, ProtMaxReq).

The next 7 statements allow users to create their own Modem Response Array. If none of these 7 statements is used then BinkleyTerm will use the original hardcoded defaults.

**** Note that if *any* of these statements are used then the user must create a complete array of all the Modem Response strings he needs to use by means of multiple statement lines. Ordering of the statements is very important as the first correct prefix match is taken.**

Be sure there are no trailing spaces or invisible characters in any of these strings.

As a guide to producing your own array, the order of the statements in BINKLEY.CFG, and the modem response strings which would produce the defaults are:

```

ModemIgnore RINGING
ModemIgnore RING RESPONSE
ModemRinging RING
ModemConnect CONNECT
ModemIgnore RRING
ModemRetry BUSY
ModemFailure VOICE
ModemFailure ERROR
ModemFailure OK
ModemFailure NO CARRIER
ModemIncoming NO DIAL
ModemIgnore DIALING
ModemFailure NO ANSWER
ModemIgnore DIAL TONE
ModemFax +FCO

```

ModemConnect <modem response>

Put in <modem response> any prefix (such as Connect) issued by your modem and which you wish BinkleyTerm to regard as a CONNECT instruction. The remainder of the modem response string is then parsed to determine connection speed etc..

ModemFailure <modem response string>

Put in <modem response> any string returned by the modem which you wish BinkleyTerm to regard as a failure to connect when dialling out.

ModemFax <modem response>

Put in <modem response> any string returned by the modem that you wish to identify as an incoming Fax connect.

ModemIgnore <modem response string>

Put in <modem response> any string returned by the modem that BinkleyTerm should ignore.

ModemIncoming <modem response string>

Put in <modem response> any string returned by the modem which the outdialer should interpret as a collision with an incoming call

ModemRetry <modem response>

This is a special statement to be used with the AltNumber statement. It could also be used (with caution) in place of other Modemxxxx statements.

ModemRinging <modem response>

Put in <modem response> any string returned by the modem which identifies an incoming ring. Don't add the Caller-ID or equivalent lines -- only those which identify an incoming ring. Usually RING is all that should be defined.

ModemTrans <number> [<prefix>]/ [<suffix>]

This instructs BinkleyTerm to dynamically select the modem prefix and suffix strings based on the modem type field found in the nodelist, or in a NODELIST.EXT file (see the section "Nodelist" in the User's Manual).

The [<prefix>]/ [<suffix>] have the same purpose and usage as they do in conjunction with the 'Dial' statement described previously. Refer to that statement for more information.

The value of <number> corresponds to a given modem type. If this type is matched, then the given <prefix>/ [<suffix>] values are used. Possible values currently are:

<number> -----	Nodelist Modem Flag Set -----
1	HST
2	PEP
3	Either HST or PEP
4	Not Currently Used by BinkleyTerm

BinkleyTerm matches modem types exactly rather than using a bitwise AND as in version 2.50. This allows lots more modem types, but if you were using this feature with BinkleyTerm 2.50 or below, you must change your nodelist generation and configuration stuff.

Callout to nodes with a particular modem type can be disabled. This is achieved using 'ModemTrans <number>' with no prefix or suffix. This allows sharing BinkleyTerm between lines with particular modem types.

MultiLink

Used by the DOS version of BinkleyTerm only. Include this if the MultiLink multi-tasker is installed, to release time-slice to the multi-tasker during certain non-processor-intensive operations. This increases efficiency for systems using MultiLink.

The next four statements are all needed in order to enable EMSI (See note about EMSI on page 29)

MyListFlags

Nodelist flags in nodelist format for EMSI

MyLocation

Defines your location in nodelist format for EMSI

MyMaxBaud

Max baud rate in nodelist format for EMSI

MyPhone

Give your phone number in nodelist format

NetFile <path>

This specifies the complete drive and path of the directory that will hold files being sent to your system via FidoNet. Incoming mail is stored here prior to processing.

If secured inbound areas are being used (see "Secured Inbound File Areas" on page 1), then this statement designates the inbound file path for mail received from systems not in the nodelist and not password protected.

NetMail <path>

This specifies the complete drive and path of the directory where your netmail will be placed.

BinkleyTerm will now scan a Squish style netmail message base to see if there is any unread netmail pending. To specify a Squish style netmail area, you must prefix the path with a "\$" and add the name of the message base (no extension). The message will remain until all mail is moved out of the netmail directory.

Example:

```
Netmail $C:\Bink\Netmail
```

The 'NetMail' statement was originally used by BTCTL to build a MAIL.SYS file produced for the benefit of certain older mail processing utilities. BTCTL is not now supplied but if needed it can be found in archives of older BinkleyTerm Versions.

NewNodeList**[This Statement is Obsolete]**

An older way of telling BinkleyTerm to use a Version 6 nodelist. Use the Version6 statement (q.v.), if using V6.

NoCollide

By default, if an incoming call is detected while preparing to make an outgoing call outgoing call will be aborted. This feature is called "call collision detection," and may not work on all modems. Using 'NoCollide' disables this feature entirely.

Nodelist <path>

This specifies the complete drive and path where processed, compiled nodelist files can be found.

NoDietIfna**[Used for Test Purposes Only]**

Do not allow the DietIFNA handshake method for mail sessions.

NoEMSI

disables EMSI (see the note about EMSI on page29)

NoFilter <ConnectSuffix>

Defines connect suffixes for which MNP filter can be disabled. 'NoFilter /Arq' will disable filtering for any "Connect xxxxx/arq".

Up to 16 of these suffixes can be defined, one per line. If you have a lot of strings starting with /ARQ, you only need the one /ARQ line.

Without NoFilter /Arq, Binkley makes no assumptions about what YOUR modem can do. It therefore watches the incoming data stream to see if your modem is passing through protocol negotiation things that it (the modem) doesn't handle, and filters them out. This can make a connection start slower. If your modem can negotiate with the other modem and settle on a protocol (it doesn't have to be ARQ) without letting any of that stuff leak through then this is a waste of time and effort. NoFilter lets you specify connect strings that guarantee that no stuff leaked through, so Binkley can expect "real data" right from the start.."

NoFullScreen

By default, a full-screen interface is used when in Unattended Mode. When this statement is used, the line-by-line screen write mode employed by BinkleyTerm Version 1.10 and earlier will be used. This statement has no effect in Terminal Mode.

NoHydra

Explicitly disables Hydra protocol

NoJanus

explicitly disables Janus protocol

NoPickup

During mail sessions, deliver mail, but do not pickup any mail that may be waiting for your system.

NoRequests

Refuse incoming file requests at all times.

NoResync

[Used for Test Purposes Only]

Do not allow SEALink Resync (an ability to restart failed SEALink mail sessions).

NoSEALink

[Used for Test Purposes Only]

Do not allow SEALink mail sessions at all, including SEALink extensions to base FidoNet protocol (FTS-0007) and WaZOO/DietIFNA sessions.

NoSharing

Disables file sharing calls in networked environments.

NoSize

Disables the calculation and display of queued file sizes for the pending outbound window display. If you see a big performance problem associated with the queued file size display, try adding this statement. When in force, the "Q=nnn" schedule flag is also disabled.

NoSLO

[Used for Test Purposes Only]

Do NOT employ SEALink "Overdrive" (an ACK-less variety of the SEALink protocol) for any SEALink network mail transfers.

NoWaZOO

[Used for Test Purposes Only]

NOTE: Not for normal use, performance with other WaZOO-capable mailers (BinkleyTerm, Opus, D'Bridge, FrontDoor, etc.) may be adversely affected.

Forces BinkleyTerm to be strictly a EMSI or FTS-0001 mailer by disabling WaZOO functionality.

NoZedZap

[Used for Test Purposes Only]

Do not allow ZedZap (Zmodem) transfers during a WaZOO session. It is primarily intended to force BinkleyTerm to use DietIFNA mode (SEALink) during WaZOO sessions.

NoZones

[Used for Test Purposes Only]

Tells BinkleyTerm to handle zones in the same manner as version 1.50 and previous versions. This essentially means that multi-zone support is turned off.

Okfile <filespec>

This designates the name of the OKFILE, a special file used with incoming file requests. <filespec> is a complete drive, path and filename designation. See the File Request section of this manual.

Overwrite

Allow existing files to be overwritten when receiving a file in Unattended Mode, or when downloading a file in Terminal Mode. Use with care. BinkleyTerm by default will NOT allow overwriting of an existing file if you're receiving a file with the same name. Instead, the name of the file being received will be slightly altered to differentiate it from the existing file.

Packer <command_line>

If used, <command_line> will be executed at the beginning of each event, but after the "Cleanup" statement's command line (if used). This might be used to pack any pending outbound mail for sending. It is suggested that <command_line> designate a batch file that would contain the command line(s) for the program(s) actually used to scan and/or pack mail.

See 'AfterMail <command_line>' on page 30 and 'Cleanup <command_line>' on page 33 for related information.

PickUpAll

When a connection has been made using EMSI this enables the pick up of mail for all your system addresses (or akas) that match the domains/zones of the other system, in the one call.

PktRsp

Build and send a packet back instead of an .RSP file. Note that you must have a flags directory for this feature to operate, as the flags directory is the "staging area" for the packet as it is being built. See the "Request Response File" section.

Point <net>/<node>**[This Statement is Obsolete]**

NOTE: This statement is supported for backward compatibility only. To specify a system address, use the 'Address' statement described previously.

This statement specifies a FidoNet node address. For regular FidoNet nodes, this is your assigned node address. For Point systems, this address is the one assigned to you by your Boss node Sysop.

PollTries <number>

This controls how many call attempts will be made when calling a system using:
Alt-D keypress in Terminal Mode, or,
A command line invocation of BinkleyTerm using the keyword "POLL"

Port <port_number>

Indicates which communications port your modem is connected to or configured as. The <port_number> corresponds to the COMM. port number, 1 for COM1, 2 for COM2, etc. Most FOSSIL drivers support COM1 and COM2, some support more. Refer to your FOSSIL documentation for information on port support and installation information. Note that in Terminal Mode, it is possible to temporarily override this setting (it will revert when you exit terminal mode).

PreDial <modem_string>

This statement will override BinkleyTerm's default predial string which is:

```
v^.....
```

The <modem_string> designates a string, which will have standard modem translations performed upon it, that is to be sent to the modem BEFORE the dial string (designated by the 'Prefix' statement, or the prefix part of a ModemTrans statement, if used) is sent.

Some modems give responses when DTR is lowered, and others require an extended period of time to resync. By using 'PreDial', you can provide an outward dialing situation better suited to your modem. In testing, a single backquote (`) has been used with the USR HST high speed modem to provide extremely fast outward dialing responsiveness.

Prefix <modem_string>

The <modem_string> is a modem command to cause the modem to dial. To dial a system, BinkleyTerm sends the <modem_string>, followed by the telephone number, to the modem. Normally, for touch-tone systems, "ATDT," is used. For pulse dial exchanges (usually older, rotary-dial lines), "ATDP," would be used.

Refer to your modem manual for more information on control options.

PreInit <modem_string>

BinkleyTerm has a default 'PreInit' string of:

```
|v^.....|^.....
```

In the PreInit statement, <modem_string> designates a string, which will have standard modem translations performed upon it, that is to be sent to the modem BEFORE the modem initialization string (designated by the 'Init' statement) is sent.

The default string is optimized to be suitable for a wide variety of modems. However, many modems may be able to work with a shorter string, which would yield faster modem initialization sequences.

In testing, this 'PreInit' string has proven fast and effective:

```
|v^^^
```

Note that 1/2 second is still added at the end of the 'PreInit' string for timing purposes, and therefore, 1/2 second is the fastest initialization that could be realized by using your own 'PreInit' setting.

PrivateNet <fakenet>

This tells BinkleyTerm the net number of a private network for which you serve as a gateway, if any. This statement is also used by Point systems to designate their private net number.

ProtAbout <filespec>

Defines the special file used with incoming file requests that are received from "protected" systems. The <filespec> is a complete drive, path and filename designation.

ProtAvail <filespec>

This designates the name of the file to be sent to a "protected" remote system which file requests "FILES" from you. The <filespec> identifies the file, and may contain an optional drive and path designation.

ProtInbound <path>

Used with secured inbound areas, this statement designates the path to the inbound file area used for mail received from "protected" systems.

ProtMaxBytes <number>

See 'MaxBytes <number>' on page 41 for information.

ProtMaxTime <number>

See 'MaxTime <number>' on page 42 .

Protocol <filespec>**[Terminal Mode Only]**

Enables the use of an external file transfer protocol. The <filespec> is a complete drive, path and filename that points to an Opus-CBCS compatible external file transfer protocol program. Refer to the User's Manual section "External Protocols" for more information.

Note that the first letter of <filespec> will be the letter used to access the protocol from the upload and download menus in Terminal Mode. Because the first letter of the filename may conflict with another external protocol, or with a hard-coded protocol, you may need to rename the executable file for the external protocol to begin with a letter that is not currently in use.

Note also that external protocols are available in Terminal Mode ONLY - they cannot be used for mail session in Unattended Mode!

ProtReqLim <number>

Limits the number of files that will be sent during one mail session in response to an incoming file request received from a "protected" system. Regardless of whether the incoming requests has wildcards, or whether multiple file requests are sent in one mail session, the maximum number of files that will be sent is <number>. If a response file is to be sent this will be included in the file count.

ProtReqList <filespec>

This designates a file, specified with full drive, path and filename.ext, similar to 'Okfile <filespec>' on page 45 but which holds details of files available to "protected" systems.

ProtReqTpl <filespec>

This designates the name of the template file used to construct response files generated for "protected" systems. The <filespec> is a complete drive, path and filename designation.

ProtSec <number>

Sets the protection level applicable to systems which are "protected"
Optional part of the implementation of faster file searching. See Page 4.

QuickNodeList**[This Statement is Obsolete]**

This tells BinkleyTerm to use a QuickBBS 2.0x nodelist. This nodelist must not be produced by QuickBBS' Qnode program, as it WILL NOT work correctly for BinkleyTerm as of this writing. A current version of ParseLst is the recommended nodelist processor for use with BinkleyTerm, however, other nodelist processors may be able to produce a QuickBBS nodelist of the required format. Refer to the User's Manual section "Nodelist" for more information.

Note: The above support is not compiled into BinkleyTerm by default, you will have to recompile the source code to use it.

Reader <command_line>

Using the Alt-E command in Unattended Mode causes <command_line> to be sent to COMMAND.COM for execution as a child process. This is typically used to invoke your local console message base reader/editor.

RecentActivityLines <number>

Used to define the total number of lines you wish to have scrollable in the Recent Activity window. Use with care under DOS due to the amount of memory this will use. See further description under heading of Recent Activity Window.

ReqOnUs

When this statement is used, incoming file requests will be filled, even if your system initiated the call. Otherwise, incoming requests "on your dime" will be refused.

ReqTemplate <filespec>

This designates the name of the response file template used for request response file construction. The <filespec> is a complete drive, path and filename designation.

Rev3**[Obsolete and was used for Test Purposes Only]**

Used by the DOS version of BinkleyTerm only. In normal use, the revision level of the FOSSIL driver you use is determined automatically. Using 'Rev3' forces the assumption that a revision 3 FOSSIL is installed., this was only used on systems which were using developmental FOSSIL drivers that did not yet fully support a higher revision.

RingTries <number>

Limits the <number> of unanswered rings allowed before hanging up on an outbound call. Your modem must be able to identify and report "RINGING" for this feature to work. default is 4.

Note: many modern modems can return a number of other responses before the "CONNECT" response; Binkley will count these unknown responses as if they were "RINGING" responses, and thus on the 4th response (which may be the "CONNECT"), Binkley will hang up, reporting "No Answer". For these modems, setting RingTries to a higher value, say 7, will be necessary.

RingWait <number>

Causes the program to wait <number> rings before answering. Default is 1 ring. This can be used to get caller-ID information into the log. Most Caller-ID systems will work with RingWait 2

SameRing

'SameRing' is used when your modem reports "RING" on BOTH incoming and outgoing calls (most modems reports "RING" on incoming and "RINGING" on outgoing), and partially disables call collision detection (see NoCollide).

ScreenBlank [<method>]

If 'ScreenBlank' is used, and 10 minutes pass without any activity (incoming call, outgoing call), then the screen will be blanked. The screen will remain blanked until the user presses the space bar. Once the space bar is pressed, the next time BinkleyTerm writes to the screen, the screen will be reactivated.

The optional <method> parameter may be "Key" or "Call". "Key" tells it to unblank upon a keypress (and is the default); "Call" tells it to unblank when a call comes in or is placed.

'ScreenBlank' works only if a Video Fossil is installed. See page 59

Also see 'BlankWait <number>' on page 32.

ScriptPath <path>

Where BinkleyTerm should look for outward dialing scripts (refer to the section "Scripts" for more information). <path> is a standard DOS path line, with optional drive designation.

Serial <number>

By default BinkleyTerm identifies itself as "UNREGISTERED".

You can include the 'Serial' statement, with a number of your choice, and the "UNREGISTERED" notices will disappear. This is OK with the authors as long as you are complying with LICENSE.260 (which basically says that unless you're a pretty seriously commercial situation, you are licensed free of charge).

The number should NOT include any punctuation characters.

Server

NOTE: This statement is not for general use.

Instructs BinkleyTerm to assume that a connection has been made immediately upon startup.

BinkleyTerm will act as if carrier detect is always high.

This is useful when establishing a null modem session between two copies of BinkleyTerm.

Shell <number> <command_line>

This allows you configure up to 9 keystroke accessible Command Shells to run programs while BinkleyTerm stays memory resident. Shells work only in Unattended Mode. While in Terminal Mode, the same keystrokes work as macros keys (see 'Macro <number> <macro_string>' on page 41 for information).

The <number> parameter is a digit between 1 and 9, which corresponds to the F1 through F9 keys.

While in Unattended Mode, you invoke a shell by pressing Alt-Fx, x indicating the shell number you wish to invoke.

The <command_line> is sent to COMMAND.COM verbatim for execution. If the program you're invoking uses command line parameters, include them in <command_line> as you would at the DOS prompt.

SlowModem

Using 'SlowModem' causes the insertion of a 1/10th second delay between each character sent to the modem while in command mode. This is for use with modems that may otherwise have a hard time keeping up with BinkleyTerm's modem commands. Most modern modems have no need for this option.

SmallWindow

During mail transfers that use the SEALink protocol, a default run ahead, in blocks, of the baud rate divided by 400 is used. This statement will limit the run ahead no more than 6 blocks. This option is used primarily with high speed modems.

Snoop

For OS/2 use only. This is used in conjunction with a program called PMSNOOP. You can then run BinkleyTerm minimized and monitor its activity/progress in Snoop's PM desktop window.

To use, include the snoop statement with a pipe name, for example:

```
Snoop \pipe\line1
```

Run PMSNOOP with the same pipe name.

(SNSERVER.DLL is also required but is usually provided with PMSNOOP)

If you use Maximus then both BinkleyTerm and Maximus (V2.xx) can use the same pipename and display to the same PMSNOOP window. Note: this support is not built into BinkleyTerm 2.60 by default. You can use the supplied source code and appropriate makefile to compile a snoop-enabled executable. Addenda: a compiled version of BT32.EXE with snoop support is now available under the filename BOS2S260.ZIP and can be found on Bob Juge's BBS and also on his FTP site.

StartBlkLen <number>

Allows adjustment of the starting Zmodem session block size. The start value can be set in the range 64 bytes to 2048 bytes. Communications on noisy lines often benefit from use of a smaller initial block size.

StartSound <filename.wav>

Play a sound at start of Unattended mode (OS/2 and Win32 machines only at present)

Statuslog <filespec>

Gives the location and name of the log file. The <filespec> is a complete drive, path and filename. See 'LogLevel <log_level_number>' on page 41 which shows what information will be recorded.

Suffix <modem_string>

NOTE: Unlike some communications packages such as Telix, normally you DO NOT need a 'Suffix' with BinkleyTerm.

BinkleyTerm will send the 'Prefix' string, followed by the phone number, followed by a carriage return to the modem for the purpose of dialing a number.

If for some reason you need to put characters immediately after the phone number BUT PRIOR TO THE RETURN CODE, use the 'Suffix' field.

SwapDir <path>

Used by the DOS version of BinkleyTerm only. This will enable "memory swapping." The <path> designated will be used for storage of a swapfile when spawning subtasks, such as jumping to DOS or invoking a packer. BinkleyTerm will swap itself out of memory except for about 5k to 8k of code. If <path> points to a RAM disk (you will need about 384k of space available), BinkleyTerm exits and reloads very quickly. The <path> parameter may designate a directory path, or both drive and directory path.

Note: If sufficient XMS (extended) or EMS (Expanded) memory is available to BinkleyTerm, it will use that for it's swap file. This is faster even than using a ramdisk.

Sysop <sysop_name>

The WaZOO method of mail transfer (originally designed by Wynn Wagner for Opus-CBCS, and supported by BinkleyTerm) and the EMSI method, designed by Joaquim Homrighausen, send a variety of information during session negotiation. Among the information is the Sysop name, normally your name. This information is not passed during an FTS-0001 mail session, and is not used in Terminal Mode.

System <system_name>

Sends the system name, normally whatever name you have given to your BBS or Point system, during WaZOO or EMSI session negotiation.

TaskNumber <number>

See the "Flag Files and Semaphore" section for details.

TaskView

Used by the DOS version of BinkleyTerm only. Indicates that the TaskView multi-tasker is installed, so that processor time-slices will be given up when the system is idle and while certain functions are being performed. This increases system efficiency.

TBBSList

[This Statement is Obsolete]

This tells BinkleyTerm to expect and use a TBBS style nodelist. To fully utilize this option, you must create the nodelist files with ParseLst 1.01 (or above), making sure that the proper adjunct nodelist files are available in addition to NODELIST.DOG.

Note: The above support is not compiled into BinkleyTerm by default, you will have to recompile the source code to use it.

TermInit <modem_string>

Identical in structure to the 'Init' statement, this statement provides a modem initialization string for use in Terminal Mode. The string will be sent to the modem if BinkleyTerm is initialized in Terminal Mode, or when switched to Terminal Mode from Unattended Mode. The string will also be sent when an Alt-I command is issued in Terminal Mode.

NOTE: Even if the 'PreInit' statement has been enabled it will NOT be used in conjunction with this initialization string.

Timeout <seconds>

Wait this length of time for a caller to press Escape or for a mail session to begin before assuming that the incoming call is for your BBS. The default value is 20 seconds. Note that <seconds> cannot be set lower than 20.

TopView

Used by the DOS version of BinkleyTerm only. Indicates that the TopView multi-tasker is installed, and that processor time-slices should be given up when the system is idle and while certain functions are being performed. This increases system efficiency.

Unattended

By default, BinkleyTerm is in Terminal Mode when invoked. When this statement is used, the program will be in Unattended Mode when invoked from DOS. This option should be used on systems where BinkleyTerm's primary purpose is as a FidoNet mail interface.

Version6

[This Statement is Obsolescent]

NOTE: This statement is not recommended for new users. The full nodelist's NODELIST.IDX file is now too large for BinkleyTerm to load, so nodes near the end of the list will not be found. If a Version 6 nodelist is used, BinkleyTerm will expect to find the files NODELIST.IDX and NODELIST.DAT compiled and ready for use.

Version7

Enables support for the Version 7 compiled nodelist format. This format offers a 40% savings in file size compared to the older Version 6 format

Make sure your nodelist compiler can handle Version 7 and be aware that the files needed are named NODEX.DAT and NODEX.NDX. The "sysop name lookup" feature in Version7 uses the SYSOP.NDX file, not FIDouser.LST as in Version6.

WinFossil

If you have installed Bryan Woodruff's WinFOSSIL program, this statement will cause the Win32 version of BinkleyTerm to use it and so be able to pass a com port handle to a DOS application.

If WinFOSSIL is not used, BinkleyTerm/Win32 uses NT or Windows95 communications which do not make provision for passing the port handle.

WinSlice

Used by the DOS version of BinkleyTerm only. Use Windows' timeslice rather than the MSDOS (int 28) timeslice.

When using BinkleyTerm with Windows 95 The WinFOSSIL program is often used. (You can use BinkleyTerm/32 to frontend a Dos based Bulletin board on Windows 95 using WINFOSSIL 1.10 or above and the WinFOSSIL command. Contact Bryan Woodruff at 1:343/294 for details)

Under MS Windows 3.1 a comm driver replacement is usually required. Try WFXCOMM (free from Delrina) used with CHCOMB which is a buffer (virtual com port) replacement. There are also two commercial programs, Turbocom and Kingcom which do the job of both the above programs.

If you are using Workgroups for Windows 3.11 then no additional Comm programs are necessary, since its standard communications driver works well.

Zone <zone_number>

[This Statement is Obsolete]

NOTE: This statement is supported for backward compatibility only.

Use the 'Address [<zone>:]<net>/<node>[.<point>][@<domain>]' statement discussed on page 29 to designate a zone.

EVENT FILES AND SCHEDULING EVENTS

It is essential that there is an event file in your system. Without it BinkleyTerm will just sit and wait for ever!

The purpose of an event file is to tell BinkleyTerm precisely when and how it should operate, for example:

- . to make calls
- . to allow housekeeping
- . to ensure that Zone Mail Hour is observed
- . to allow access to BBS users (if you have a BBS system) at certain times but exclude them at other times (such as ZMH)
- . to restrict long distance calls to times when they are cheap
- . to send high priority mail as quickly as possible

Moreover, you can do all this on a presetable minute by minute, daily, weekly or monthly basis . Many other options are possible as explained below.

EVENT FILE FORMAT

Although events may be placed in the configuration file, they should be contained in a special file named BINKLEY.EVT. In this way, small changes to the configuration file will not cause BinkleyTerm to re-run the current event; this will only happen if the BINKLEY.EVT file is edited.

By storing event schedules in a flat text file, events can be easily edited at any time with a standard ASCII text editor. No special utilities are required.

Note that each time an edit is made to BINKLEY.EVT (or to the configuration file if events are listed there), the files BINKLEY.SCD and BINKLEY.DAY should be deleted at a time when BinkleyTerm is not running. When restarted, BinkleyTerm will re-build its binary schedule file, which will in turn cause BinkleyTerm to re-run the current event. This is normal operation, and is necessary to allow BinkleyTerm to properly register the schedule changes.

Note: To avoid all Forced events (such as daily maintenance, or scheduled polls) being rerun when BinkleyTerm restarts, after deleting BINKLEY.SCD as above, you should restart Binkley with the NOFORCE command line parameter, then exit and restart it as usual. This procedure will prevent the rerunning of all such Forced events.

Each event is on a separate line in the BINKLEY.EVT or configuration file. Here is a sample of such a line:

```
Event All 03:00 04:00 L=10 N B E1=10 E2=20 E3=30
```


PARAMETERS AND FLAGS FOR EVENT FILES

The syntax for the entries is:

Event <day> <start> [<stop>] [<string>] <flags/options>

Details of these parameters are as follows:

<day>

This tells BinkleyTerm which days this event line applies to. This is a REQUIRED parameter.

Options are:

All	Every day of the week
Week.	Weekdays, Monday through Friday only
WkEnd	Weekends, Saturday and Sunday only
Sun	Sunday only
Mon	Monday only
Tue	Tuesday only
Wed	Wednesday only
Thu	Thursday only
Fri	Friday only
Sat	Saturday only

Several <day> parameters can be linked with the pipe character (|) to indicate more than one option. For example, "Mon|Wed|Fri" would indicate that the event applies to Monday, Wednesday and Friday only. No spaces may be used between the parameters.

<start>

This tells BinkleyTerm what time to start the event, in 24 hour "military" time, in the format hh:mm, where hh is the hour and mm is the minute. Note that <start> must NOT be greater than <stop>, i.e., events may NOT stretch through the midnight hour. The <start> parameter is REQUIRED.

The <start> parameter may optionally take date information as well. Allowable additional information is the month in numeric form (1 for January, 2 for February and so on, ending with 12 for December) and a day of the month. This additional information is used IN ADDITION to the <day> parameter previously described. A field that includes date information would be in the format:

hh:mm,month,day

Where hh is the hour, mm is the minute, month is the numeric month, and day is the day of the month. You may also leave out the day parameter, like this:

hh:mm,month

If you wish to leave out a month parameter, simply place a zero into that field, indicating that the event is to take place on all months, like this:

hh:mm,0,day

Using the date options with the <start> parameter offers significant scheduling power. For example, if you wanted something to occur on the second Friday of each month, you can do so without a lot of manipulation by creating seven events with the same errorlevel, each corresponding the possible date values for the second Friday of each month, like this:

```
Fri 23:00,0,8 23:00 F E1=131
Fri 23:00,0,9 23:00 F E1=131
Fri 23:00,0,10 23:00 F E1=131
Fri 23:00,0,11 23:00 F E1=131
Fri 23:00,0,12 23:00 F E1=131
Fri 23:00,0,13 23:00 F E1=131
Fri 23:00,0,14 23:00 F E1=131
```

In any given month, the event required on the second Friday of the month would cause an errorlevel 131 exit (which has been previously associated in your batch file with the function you wish to occur). If you wanted something to happen every leap year, you could create an event to cause an errorlevel 132 exit on February 29, like this:

```
All 23:15,2,29 23:15 F E1=132
```

Note that if you want a specific exit on a specific date regardless of the day-of-the-week, you should use "All" for the <day> parameter for that event.

[<stop>]

This tells BinkleyTerm what time to stop the event. This parameter is OPTIONAL, and defaults to 60 minutes after the start time should the parameter be omitted. This is given in the same format the <start> parameter, as military time, hh:mm where hh is the hour and mm is the minute.

[<string>]

This parameter is OPTIONAL. If used, <string> designates a string of characters to be added to the command line of the configuration file parameters 'Packer,' 'AfterMail' and 'CleanUp.' The string should be enclosed in quotation marks. For example:

```
Event All 00:00 01:00 "-sA -c"
```

The <string> is appended to the command lines given for these options in the configuration file, and should be ignored by those that do not need it. It is suggested that batch files be used with the above mentioned configuration file options, and that the batch file(s) filter out unneeded information given in <string> before calling a program that might "cough" because the command line is wrong.

Up to 32 extra characters can be added with the <string> parameter.

<flags/options>

These provide information about the event. The various flags and options should be separated by a space.

Flag	Description
---	-----
\$	This flag causes all bad-call files in the outbound areas to be deleted.
A=	Controls the idle time between outgoing call attempts. The format is "A=x" where x is the number of seconds desired, which can be a number between 0 and 1800 (1800 seconds = 30 minutes). The average wait between calls is based on +/- 50% of the number specified, i.e., A=60 would yield a wait time in the range of 30 to 90 seconds, 60 being the average. Should the A= parameter not be used, the default value is 120, for an average wait time of between 60 and 180 seconds.
	This flag does NOT operate when a forced (Alt-M) poll is made. In that case repeated calls are made until a connection is established.
B	Indicates that BBS operation is allowed during this event. If this flag is NOT present, callers will be greeted with the message "Processing mail. Please hang up." Use this option at all times (if you run a BBS) except during mail schedules, such as Zone Mail Hour.

Flag	Description
---	-----
C	Means call out during this event only if there is mail marked as Continuous Mail. The call is still subject to any other restraints such as cost.
D	Indicates that the event is dynamic. Dynamic events continue until there is no longer any mail of the specified type to be sent. For example, if the dynamic event specifies that local Continuous Mail is to be sent, the event will continue until there is no more local Continuous Mail to be sent, or until the event ends, whichever happens first. When the dynamic event ends, the non-dynamic event scheduled for the same time slot will take over. If no such event exists, the system will accept mail, but will dial out to no one. Note that dynamic events must be started before or at the same time as non-dynamic events, if the dynamic event is to overlap a non-dynamic one. <ul style="list-style-type: none"> • Notes on Event exits E2, E3, E4-9, and AfterMail: • The order of exit precedence is as follows: • E3, E4-E9, E2, AfterMail. • Where more than one E4-E9 exit applies, the lowest one is taken. If the ErrLevelShell statement is enabled then each of the shells which apply will be taken, one after the other. If any E4-E9 exits also apply, the one with the highest priority will be taken after the shells.
E1=	Causes an exit with the given errorlevel at the beginning of the event. 'E1=10' would cause an exit to the start-up batch file with errorlevel 10 when the event begins. This is a good method of executing functions once daily, for example, message base maintenance software, and so on. Once the E1= exit has been made, it will not occur again until the next time the event is scheduled.
E2=	Causes an exit with the given errorlevel after mail is received. The E2= exit is only executed if the incoming mail does not meet the criteria for an E3= exit, or if an E3= exit does not exist. Using your batch file, the errorlevel set for the E2= option should invoke mail unpacking software to merge the incoming mail with your message base.
E3=	Causes an exit with the given errorlevel after compressed mail is received. If mail is received during the event, and compressed mail is not a part of the mail received, then an E2= exit is performed. If compressed mail was received (even in conjunction with other mail or files) then the E3= exit is performed. Using your batch file, the errorlevel you set for this option should invoke mail unpacking software that can handle compressed mail and merge it with your message base.
E4= to E9=	Event exits E4-E9 can now be followed by a three character string. If that string is matched in the file extension of a received file, then the designated errorlevel exit will be taken. For example: E4=100,TIC (Received .TIC Files Cause Exit 100) E5=100,FLE (Received .FLE Files Cause Exit 100) E6=110,REQ (Received .REQ Files Cause Exit 110) E7=120,MO? (Received .MO? Files Cause Exit 120)
EF=	Causes an exit with the given errorlevel on Fax reception
F	Indicates that the event should be "forced" so that it will occur at the first possible moment. USUALLY YOU DO NOT NEED TO USE THE F FLAG. BinkleyTerm will execute the current event anyway if for some reason the start time is bypassed (but before the stop time passes). If you do use this option, use it only on zero-length events; those events for which the <start_time> and <stop_time> are the same.
H	"High-Priority Mail" - BinkleyTerm will send Continuous flavored mail IMMEDIATELY, no matter what the cost. All other mail flavors are sent according to cost or other constraints imposed by the current event. This behavior mirrors that of Crashmail under Opus 1.1x+ with one exception - BinkleyTerm makes calls at normal intervals during an H event, rather than forcing a repetitive poll as with Alt-M.
K	Means "Do not to send to any nodes marked in the nodelist as #CM (accepts Continuous Mail) during this event".
L	The L flag is concerned with call costs and is usually used to stop calls being made at times when call costs are high. There are two alternative methods of using this flag: 1) The default method is the original US system where L refers to the COST of a call in Cents per minute as indicated in the Nodelist Cost field. 2) The Eurocost method (See "CALL COSTING" on page 28 for more details) where L refers to the TIME allowed for one "Call Unit". To use this system statements are added to the BINKLEY.CFG and the Nodelist cost field is used to hold TIME values measured in tenths of a second. Note: In the original US system LOW values of "L" indicate cheap calls whereas with Eurocost calls

Flag	Description
---	----- get cheaper as the "L" value rises.
L	Can be used alone, without any parameter, and is then equivalent to "L=0". The L flag used in this way where local calls are free is sometimes called the "Local" flag.
L=nn	Means: "Only send mail in this event if the figure in the nodelist cost field is equal to OR LESS THAN nn"
L<nn	Means: "Only send mail in this event if the figure in the nodelist cost field is less than nn"
L>nn	Means: "Only send mail in this event if the figure in the nodelist cost field is greater than nn"
M	The "M" flag indicates that the event is a "mail" event, and that it is okay to send mail to anyone in the nodelist, regardless of their #CM designation. This flag is normally used during local mail schedules, and during Zone Mail Hour.
N	Means "Do not accept inbound file requests during this event".
Q=	Inhibits BinkleyTerm from calling out with less than nnn bytes of data for a node (?LO + ?UT sizes). You should probably have at least one event with Q=0 (the default if none is specified) in order to get the mail out.
R	Indicates that the event is "Receive Only" so outgoing calls will not be made to send mail, however, the mail will be sent if a poll is received.
S	Designates an event which is "send only" meaning that BinkleyTerm will continue to send normally, but will not answer the phone (or if the modem is in auto-answer mode, BinkleyTerm will not respond). Note that you can use "R S" for events where BinkleyTerm should neither dial out nor answer the phone.
T=	This option allows you to control the maximum number of call attempts and failed connects that will be acceptable to BinkleyTerm. The T= option accepts two parameters, and is used in the format "T=x,y" where x is the maximum number of failed connects (carrier established, session fails - a chargeable call in toll situations), and y is the maximum number of call attempts (no answer, no session - generally NOT a chargeable call in toll situations). Generally, set the x parameter very low, so as not to rack up charges on your phone bill should the call be long distance or another toll call. Set the y parameter high, since these calls are usually not charged for. The default x parameter is 3, the default y parameter is 10,000
X	The X flag is now obsolete for the reasons given below. Many users will have this flag in their event file and these notes may explain why it may not work as expected. The X flag was intended to prevent a call out being made for stand alone .REQ files, but, since version 2.40, BinkleyTerm's behavior has been altered and calls are never made for standalone .REQ files. The X flag never prevented BinkleyTerm from calling out if other mail (such as a packet) existed. It follows that because most utility programs nowadays create a packet on which to piggyback the .REQ, BinkleyTerm may still call out for a file request even if the X flag is set.

Many of the above options can be used with one another, and in fact usually are. Constructing a working event schedule can be a time consuming process requiring a certain amount of trial and error. Since the event schedule plays a very important role in deciding when mail will be sent, it should be manipulated VERY carefully to avoid having BinkleyTerm make toll calls during high rate periods.

A very minimal sample event schedule is shown in NEWUSER.EVT, contained in the distribution package.

SCRIPTS

SCRIPT FILE USAGE

A script is a series of instructions used when dialing a particular system. They allow the system to "look" for particular information coming across the line, and act according when the desired information is received within a set time limit. Scripts are essentially a mini programming language, and as such, take study and practice to use effectively.

Once written, scripts are associated with a particular nodelist entry for use each time the given node is dialed.

Possible applications for scripts include accessing packet switch networks, such as GTE/Telenet's PC Pursuit service, that require multiple sets of operations to reach the desired destination.

The use of a script is triggered by the appearance of a file name, inside double quotes, in the phone number field of a nodelist entry. For example, instead of seeing 1-303-555-6789 in the nodelist data file, you might see "MYSCRIPT.SCR"1-303-555-6789

Notice that the name of the script file is IMMEDIATELY (without spaces) followed by the area code and phone number. The field must appear in the format shown, including hyphens. The area code portion of the number can be up to 10 characters long (for use with certain long distance carriers). Refer to the documentation for your nodelist processor for information on inserting information into the phone number field of a nodelist entry.

Script references can also be placed in your configuration file by way of the 'Dial' statement. See 'Dial <match_string> <new_prefix>/[<new_suffix>]' on page 35 for details.

PREPARING SCRIPTS

Scripts are stored in a flat ASCII text file, and edited using any standard text editor.

The following restrictions apply when creating scripts:

1. Any line beginning with anything other than a letter or a colon (:) is ignored as a comment.
2. All lines must begin flush left (no leading spaces or tabs)
3. All statements that take parameters must be followed by exactly ONE space between the statement and the parameter.
4. There should be NO extra characters at the ends of lines. This includes space characters. All characters on a line are significant, including any extra spaces that you may have inadvertently included.
5. Script statements and script labels are NOT case sensitive.

Please note that spaces can cause hard-to-track-down problems. Spaces are significant characters, meaning they are NOT ignored in patterns, etc. Do not use a space unless you intend to, and do not leave any at the ends of lines unless you want them there.

SCRIPT STATEMENTS

:<label>

The colon (:) starts a label. Labels can be up to 20 characters long. Control can be passed to the location in the script identified by a label using the "If" and "Goto" script statements. Up to 50 labels per script are allowed.

Abort [<start_time> <stop_time>]

Allows conditional aborting of script execution based on time of day. If used without parameters, it causes unconditional aborting of execution. If parameters are used, script execution will abort if the current time is between the hours given with <start_time> and <stop_time>. For example, "Abort 8:00 22:00" would make the script abort between the hours of 8:00am and 10:00pm. The hours CAN wrap through midnight, "Abort 22:00 3:00" would be an example of this.

Areacode

Transmits the areacode portion of the phone number to the modem, as shown in the given nodelist entry.

Baud [<baud_rate>]

This statement sets the baud rate for the call to the value given. If no <baud_rate> is given, the baud rate as listed in the nodelist for this node is used.

BPSxxxx

Allows branching or actions based on baud rate.
For example, the statement

IF BPS2400 DO2400

would cause script execution to jump to the previously defined label "DO2400" if the current connection was at 2400 bps (baud).

Break [<duration>]

This causes a "break" signal to be sent, as needed with some types of host systems. <duration> designates the number of hundredths of a second for the break signal to last. If the <duration> parameter is not given, the default duration value of 100 (1 second) will be used.

Carrier

Continue the script if there is carrier, abort if not.

Comm <settings>

Allows setting of the communications parameters. <settings> is a three-character string, consisting of the number of data bits, parity and number of stop bits. For example, a <settings> string of "8N1" would cause the parameters to be set to 8 data bits, No parity, and 1 stop bit. "7E1" would cause 7 data bits, Even parity, and 1 stop bit. "7O2" would cause 7 data bits, Odd parity, and 2 stop bits. Possible values are 7 or 8 for data bits, E (even), O (odd) or N (none) for the parity, and 1 or 2 for stop bits.

NOTE! The string is NOT checked for accuracy. The user is responsible for making sure that it is correct!

Dial

Dial the entire telephone number, and wait for a valid response. Continue if there is carrier, abort if there is not.

DOS <command_line>

Causes the <command_line> to be sent to DOS for execution.
Upon completion, script execution continues. Sufficient memory must exist for any application executed by this command.

Download <x>

<x> is a protocol selection, Z for Zmodem, or S for SEALink

Goto <label>

This statement causes the script processor to jump to the location in the script pointed to by <label>. A colon (:) must have previously been used in the script to identify the <label>. If the label does not exist, the script aborts.

If <pattern_number> <label>

If a match for <pattern_number> was found at the last 'Wait' statement, transfer control to the point in the script identified by <label>. If a match was not found, control continues to the next statement in the script. 'If' can be used at any time prior to the next 'Wait' statement.

NoEMSI

Disables EMSI for the session with the system the script is calling.

NoWaZOO

Forces BinkleyTerm to be strictly an FTS-0001 mailer by disabling WaZOO functionality for the current outgoing session only. This is primarily of interest to coordinators who wish to verify that their nodes are meeting FidoNet compatibility requirements.

There is no benefit to the average user from using this statement. In fact, performance with other WaZOO-capable mailers (BinkleyTerm, Opus, D'Bridge, FrontDoor, etc.) will be adversely affected.

Pattern <pattern_number> <string>

This statement establishes a pattern for the script handler to look for during the next 'Wait' statement. <string> IS case sensitive. The script handler will look an EXACT match for the series of characters in <string> during the next 'Wait' statement. Up to 8 patterns can be used, and they can be reused or reset at will. Up to 20 characters can be used in a pattern. The purpose is to wait for a given string from the host, or a particular modem response string, and to act accordingly.

Phone

Transmits the local portion of the phone number to the modem, as shown in the given nodelist entry. Hyphens are stripped automatically.

Rawxmit <string>

This works in the same manner as the 'Xmit' statement, except that dial translation is NOT performed.

Session

This should be used at the end of a script which has been successful. It tells BinkleyTerm to begin a NetMail session with the remote system.

Speed

Causes BinkleyTerm to send the baud rate divided by 100 as a string. The baud rate used is the rate specified for the node in the nodelist, or the rate specified by a prior call to the 'Baud' statement. For example, if the current connect rate was 2400 baud, the string "24" would be sent when this statement is encountered.

Timer <seconds>

Sets a master countdown timer to <seconds>. If the timer expires, the script will abort. This allows you to set timeouts on any portion of, or the entire script. You may reset the timer by using another 'Timer' statement.

Wait [<seconds>] [<label>]

Wait for a maximum of <seconds> for one of the previously set patterns to be matched. If a pattern is matched, the script continues, otherwise it aborts. If <label> is provided, the script will resume operation at the label specified upon timeout. Note that the default value is 40 seconds. Both <seconds> and <label> are optional. A <label> may be specified without specifying <seconds>. For example:

Wait 40 foo is the same as: Wait foo

Upload <x> <filespec>

where <x> is the protocol selection: Z for Zmodem, S for SEAlink

Xmit <string>

Transmits <string> to the modem. Normal BinkleyTerm translations are valid (refer to the section "Dial Translation" for information).

THE BINKLEYTERM WINDOWED INTERFACE

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

VIDEO FOSSIL

A Video Fossil (VFOSSIL) is a separate driver program used by BinkleyTerm when running in a DOS environment. VFOS_IBM is a video fossil in common use. This is a freeware program likely to be available from wherever you obtained BinkleyTerm.

OS/2 and Win32 users do not require a Video Fossil. Video services similar to those made available in DOS by using the Video Fossil are native to the operating system and are used directly by BinkleyTerm.

When running under DOS, a video fossil is required for color displays. Even if you have a monochrome display it is recommended that a Video Fossil be used as the windowing operations mentioned below can take place much faster than they would without one. Also the 'ScreenBlank' function requires the presence of a Vfossil.

Some more recent Video Fossil implementations may also allow BinkleyTerm to be used on EGA and VGA systems in extended text modes of 132 columns by 43 lines. The mode switching must be performed by the utility software that accompanied your video card prior to invoking BinkleyTerm.

THE DISPLAY SCREEN

There are five information windows displayed on-screen:

CURRENT SETTINGS WINDOW

This window contains various information about your system, including the current date and time, current event, port and current baud rate, status and multitasker type, if any. The current event line features a number, and a list of event flags. The number corresponds to which entry in the Event file contains the line that covers the current event. The first 'Event' statement in the file would be event 1, the second would be event 2, and so on. The flags are letters that are a subset of those used with event scheduling. Here is a list of letters that may be found in this window:

C	Continuous Mail Only Event
L	Local Only Event
N	Do Not Accept Inbound File Requests
R	Receive Only Event
B	BBS Use Allowed
D	Dynamic Event
K	Do Not Send to #CM Nodes

Note that not all event flags will be displayed in the window. For details of all event flags, see "EVENT FILES and SCHEDULING EVENTS" on page 51.

TODAY AT A GLANCE WINDOW

This window contains several lines of information regarding the activity on your system since midnight. Note that the totals given apply only to calls handled by BinkleyTerm's Unattended (mailer) Mode, and do not include Terminal Mode totals.

The first line, "BBS/Mail," lists how many BBS calls and mail calls have come in, separated by a slash. For example, 2/15 would indicate 2 BBS calls and 15 mail calls since midnight.

The second line, "Calls Out," lists the number of dial attempts that have been made by your system, successful or otherwise.

The third line, "Good/Cost," shows how many successful outward connects have been made and the cumulative cost of all the successful calls, separated by a slash. For example, 3/100 would indicate that 3 successful outbound calls have been placed, and that together, they cost 100 units (in the United States, units are cents, which would mean that the cost was 100 cents or \$1.00). The cost is calculated based on the cost shown in your compiled nodelist files; the cost shown there is assumed to be a per-minute value in calculating a per-call cost.

The fourth line, "Files I/O," shows how many files (packets, compressed mail or other incoming files) have been received and sent, separated by a slash. For example, 12/3 would indicate that 12 files have been received, and 3 have been sent.

Finally, the fifth line indicates the type of last call. If the last call was incoming, the display will indicate whether the call was a BBS call, external mail call or incoming FAX, or the node address will be displayed if it was a mail call. If the last successful call was outgoing, the node address will be displayed.

PENDING OUTBOUND MAIL WINDOW

This window displays a list of systems for whom mail is pending, along with related information. The window can be scrolled using PgUp, PgDn, Home, End, Up-Arrow and Down-Arrow keys. The top line of the window lists the node that is being called, or was just called. The second line lists the node to be called next.

For each node, BinkleyTerm lists the number of files which BinkleyTerm needs to send, the total size of the files, and a "status" which is explained below. The file size and file count display can be disabled by means of the NoSize configuration statement.

Pending Outbound Status Information:

In the "status" column, information about the mail for the various nodes is also displayed. Each attribute of the mail is marked by a single letter. The letters and their meaning are:

Letter	Meaning
-----	-----
C	Continuous Mail
H	Hold-for-Pickup Mail
D	Direct Mail
N	Normal Mail
R	File and/or Update Request
S	Status

To the right of the mail attributes, BinkleyTerm will show a single character status which indicates whether it expects to be making any outgoing calls to deliver mail to this node. The values which BinkleyTerm will display and their meanings are:

Symbol	Meaning
-----	-----
*	Outgoing mail will be sent
x	Too many bad connects (considered undialable)
#	Already called, but mail still pending
-	Cannot call this node during this event
!	This node not found in the Nodelist

When BinkleyTerm is making an outbound call, the node which is being called is highlighted in the Pending Outbound Mail window. This allows you to know which node has been called, even after the pertinent information has scrolled out of the Recent Activity window (described below). This feature can also be helpful in determining whether an active mail session was initiated by you, or was started by an incoming call.

Rescan of the Outbound Area

The outbound area is re-scanned for new additions under the following circumstances:

- After a keyboard shell has been invoked
- After 'AfterMail' is run (if designated)
- After 'Packer' is run (if designated)
- After message editor (Alt-E) has been run
- After user manipulates the outbound, sending, requesting files, etc. either from unattended mode or in zoomed outbound window
- After 10 minutes of no activity
- After executing an Alt-I modem Init in unattended mode
- After BinkleyTerm finds a BTRESCAN.@@ file in the flags Directory

Note the 'AfterMail', 'Packer' and 'Reader' (message editor) are designated in the configuration file. Refer to the "Configuration File" section for additional information.

RECENT ACTIVITY WINDOW

This window gives information about what the system is currently doing, or has just done. This includes identification of the remote system, the type of connection, what files were transferred, etc.

The information in this window essentially reflects the information that BinkleyTerm is writing to its logfile.

The recent activity window is scrollable. Set the Max lines to scroll with the RecentActivityLines configuration statement and use CTRL+(uparrow/dnarrow, pgdn/up, home end) keys to scroll. DOS users should beware that this uses a fair bit of memory, and should also note that for some reason the

CTRL/UPAR and CTRL/DNAR do not seem to work under DOS. You should be able to redefine these commands to other keys and recompile your language file to gain access to the functions.

TRANSFER STATUS WINDOW

This window gives particulars about current file transfers that may be taking place if a connection is active. For most types of mail transfers, this includes filename and the number of bytes transferred; for other types of transfers, the number of blocks transferred may be given.

Between mail sessions, the window will display the amount of time until the next event begins and the flags associated with the next event. If your message base uses the .MSG format, BinkleyTerm is able to determine that unread mail is waiting in the NetMail area and an indicator will be displayed in this window. This feature can now be enabled for a Squish type message base by including the 'NetMail' statement in your Binkley.cfg with the associated filename preceded by a dollar sign(\$). For example:

```
NetMail    $c:\bink\netmail
```

The success or otherwise of the "squish Netmail advise feature" depends on how you handle your netmail area. The message advising you of unread netmail will remain as long as there is any unread mail, even outgoing messages, areafix msgs etc., which you would not normally read. No doubt this will be amended in a future release.

BINKLEYTERM LOG FILE

If a log file is defined with the StatusLog configuration statement then information about the program' activities can be logged. The LogLevel statement defines the extent of the logged information.

File request searches can be logged, see the section on Incoming File requests.

If the AfterCall statement is used then details of each call can be logged.

For debugging purposes, the Debug and LineUpdate statements provide additional information to the log.

BinkleyTerm can maintain a separate log of call costs. Enable this by setting the "EuroCost" command, and making "CostLog <file>" and "CostUnit" settings.

US CostUnit seems to be about 2 and European seems to be 23. Adjust as you need.

With the CostLog in use, quite a bit of additional information is available and is logged at the end of each session. i.e.

- Day number for this record
- Number of BBS callers
- Number of mail calls
- Number of outgoing calls made
- Number of outbound call successes
- Number of files received
- Number of files sent
- Type of last call
- Address of last, excl. Domain
- Domain of last
- Time of last outbound session
- Address of next, excl. Domain
- Domain of next
- Cumulative of call costs

Size of files received
 Time of files received
 Errors while receiving files
 Size of files sent
 Time of files sent
 Errors while sending files

COMMAND LINE PARAMETERS

BinkleyTerm offers a selection of command line parameters which each have a unique function. The words that are described below are simply placed on the DOS command line - no hyphens or slashes are necessary. For example:

BT NoForce Unattended Share Config C:\BT\SYS1.CFG

A full list of the command line parameter options is as follows:

Command	Function
-----	-----
NoForce	Don't force events that have already passed.
Mail	For Point system use, this causes BinkleyTerm to attempt connection to the boss upon start-up.
Share	Leaves the FOSSIL driver "hot" (does not de-initialize the driver upon exit) for use with other FOSSIL-based systems.
Dynam	Restart the current dynamic event, if any, even if it may have already executed.
Unattended	Run in Unattended Mode, regardless of whether the 'Unattended' configuration file statement is used.
Config	Allows specification of a configuration file other than the default (BINKLEY.CFG). This parameter must be followed by a single space, and then the name of the configuration file, including drive and path if applicable, for example: "BT Config C:\BT\MYCONFIG.CFG"
Poll	Instructs BinkleyTerm to immediately poll a given node and keep trying until successful. No notice is taken of any event file entry. Upon completion of the poll, BinkleyTerm will exit. Example: "BT Poll 1:132/101"

TERMINAL MODE KEYSTROKES

While in Terminal Mode, several keystrokes are available that allow uploading and downloading of files, changing of communications parameters, and so on.

Keystroke	Function
----- Alt-=	Used to toggle "Doorway Mode" on and off. When in this mode, all keystrokes are sent out the modem as entered (except for the command used to toggle the mode on & off). If a function key is used, a zero followed by the scan code is sent.
Alt-B	This allows you to step the baud rate up to the next higher value. BinkleyTerm supports baud rates of 300, 1200, 2400, 4800, 9600, 19,200 and 38,400. 38,400 baud wraps to 300 baud.
Alt-C	This allows you to set various communications parameters, including number of data bits, parity, and number of stop bits. You are prompted for the information. Note that when 8 bits are set, BinkleyTerm defaults to no parity, and you are not prompted for the setting.
Alt-D	Allows you to dial out. When prompted, you may enter one of three items: a telephone number, a FidoNet node address in the form [<code><zone></code>]: <code><net></code> / <code><node></code> , or a FidoNet Sysop name. Entering a FidoNet address requires the presence of a compiled nodelist, properly referenced in the BinkleyTerm configuration file. Entering a Sysop name requires that your nodelist compiler has created a compatible list, either FIDouser.LST for a Version6 Nodelist or SYSOP.NDX for Version7.
Alt-E	Erases the display screen.
Alt-F1 through Alt-F9	These keystrokes allow you to send user-defined macros. Please refer to the section "Configuration File" for information on Terminal Mode macros.
Alt-F10	This provides a brief help screen, listing the key-presses available to you in Terminal Mode.
Alt-H	Hang-up. This command toggles the modem's DTR (data terminal ready) line, thereby disconnecting any call in progress. The modem init string is sent to the modem after DTR is toggled.
Alt-I	Re-initializes the modem.
Alt-J	Jump to DOS. Invokes COMMAND.COM to allow for "quick and dirty" DOS commands on the fly. Enter "EXIT" at the DOS prompt to return to BinkleyTerm. BinkleyTerm will automatically return to the directory it was run from when returning from DOS.
Alt-L	Toggle session logging on and off. If 'CaptureFile' is enabled in BINKLEY.CFG, Alt-L opens that file (for append if it already exists). If 'CaptureFile' not enabled, this session logging function allows you to designate a file or printer, and have the entire terminal mode session echoed to the selected file or device. When first invoked, this command will prompt you for a file name. Type in the desired name of the log file, including drive and path designation if desired. You may also enter a printer device name, e.g., PRN, LPT1, LPT2, etc. for printer echoing. Invoke the command again to end logging to the file or device.
Alt-M	Poll a node in the nodelist, by node address. This command requires that mail handling be implemented as described in the User's Manual section, "Unattended Mode Overview". You are prompted for a FidoNet node address. Once entered, BinkleyTerm will dial the system (if it exists in the nodelist) and attempt to exchange mail with the system. If there is outgoing mail to the system, or if mail is waiting for you on the remote system, it will be sent during the mail session.
Alt-P	This command allows you to change the communications port in use. Invoking the command shifts the port number to the next higher port. The number of ports supported is determined by your hardware, and by the 'MaxPort' statement in the

Keystroke -----	Function -----
Alt-R	<p>configuration file.</p> <p>This allows you to dial out with a "scan list". Empty elements are designated by 'Null.' Using the function keys (or equivalently numbered numeric keys) you can select the scan list element you wish to enter. Enter a telephone number, node address or Sysop name, just as you would with the Alt-D command. Press "Enter" to "store" your entry for that element. When you have stored all the desired elements, press "Enter" to begin the dialing process.</p> <p>Each number will be dialed in sequence, until a connection is made. After the session is completed, a 'Null' will be stored for that element. Invoke Alt-R again, and press "Enter" to dial the remaining elements. Press "Escape" at any time to abort the dialing process.</p> <p>Note that unless saved, the dialing list is volatile. Once you exit the Terminal Mode, your dialing list will be erased. It is possible to save to and retrieve from the disk media up to 10 separate lists of entries. To save a list, press Shift- Fn, where 'n' is the number of the list to save. For example, Shift-F3 would save the current list of 10 entries to list number 3. To retrieve a previously saved list, press Alt-Fn, where 'n' is the number of the list to retrieve. For example, Alt-F5 would retrieve the 10 entries previous saved as list number 5.</p>
Alt-S	<p>Invoking this command sends a "break" signal via the modem to the remote host. This command requires that a FOSSIL of revision 5 or later be installed.</p>
Alt-U	<p>This command is used in Point and BBS installations to shift to Unattended Mode. BinkleyTerm is put into the mailer mode, ready to accept calls from remote systems. In order to operate, the mailer functions must be implemented as described in the User's Manual section, "Unattended Mode Overview".</p>
Alt-X	<p>Allows you to exit the program and return to DOS. If BinkleyTerm was invoked with a batch file, this command will cause BinkleyTerm to exit to the batch file with an errorlevel of 1. The batch file must trap this errorlevel, and exit accordingly.</p> <p>Note that this function DOES leave the DTR (data terminal ready) line on the modem "high."</p>
Alt-Y	<p>Used primarily for Point operations, this function initiates a mail session with the host. This is designed ONLY to work with the system listed as your Boss node in the configuration file.</p> <p>Issuing this keystroke invokes a temporary mailer mode, and BinkleyTerm attempts to dial the Boss system to exchange mail. Mail handling must be set-up as described in the User's Manual section, "Unattended Mode Overview".</p> <p>If mail is waiting to be sent to the Boss, or if mail is waiting on the Boss system for pickup, it will be sent during the mail session.</p>
PgDn	<p>This allows you to download a file from the host with which you are connected. You are prompted for the desired protocol to use, and for the required file information, if needed (protocol dependent). If you have the 'Download' statement properly implemented in the configuration file, the downloaded files will be placed in the directory specified. Otherwise, the file will be saved to the current directory.</p>
PgUp	<p>This allows you to send a file to the remote host. You are prompted for the desired protocol to use for the transfer, as well as the file information. You may specify a complete drive, path and filename using standard DOS conventions, e.g., C:\PATH\FILENAME.ARC.</p> <p>When a batch protocol is used (SEAlink, Telink, or Zmodem), wildcards are allowed in the filename.</p>

UNATTENDED MODE KEYSTROKES

While in Unattended Mode, several keystrokes are available that perform various functions, such as manual polling, call forcing, Command Shells, and so on.

Keystroke -----	Function -----
F1 through F10	These keys cause BinkleyTerm to exit with an errorlevel of 10 times the function key number pressed. In other words, F1 causes an errorlevel 10 exit, F7 would cause an errorlevel 70 exit. Your batch file is used to capture and process these errorlevels.
Alt-F1 through Alt-F9	These keystrokes invoke user-defined Command Shells, to execute programs or utilities as desired. Refer to the section "Configuration File" for additional information.
Alt-F10	This key causes BinkleyTerm to display a brief help screen, listing the various keystrokes available to you in the current operating mode.
Alt-A	Sends answer string to the modem
Alt-B	Forces a screen blank to occur. Any system activity or a press of the space bar will reactivate the screen display.
Alt-C	Causes information in the Unattended Mode "Today at a Glance" window to be "zeroed" out. The information is NOT stored to the BINKLEY.DAY file prior to being zeroed.
Alt-E	This invokes your message editor/reader, as designated by the configuration file parameter 'Reader.' Refer to the section "Configuration File" for more information.
Alt-G	Allows the generation of an outbound file request. When executed, this command will pop-up a window on the screen where you will be prompted to provide information appropriate to creating a file request. Note: An outbound directory must pre-exist for the Zone (and, optionally, Domain) of nodes you are calling.
Alt-I	Re-initializes the modem (equivalent to Alt-T/Alt-U in previous BinkleyTerm versions). Also causes BinkleyTerm to re-read the outbound mail holding area.
Alt-J	This causes BinkleyTerm to exit to DOS and stay resident (Command Shell). This is probably the most efficient method to perform a quick DOS command "on the fly." Enter "EXIT" at the DOS prompt to return to BinkleyTerm. BinkleyTerm will automatically return to the directory it was run from when returning from DOS.
Alt-K	Allows you to kill all mail for a particular node. When executed, you will be prompted for confirmation.
Alt-M	This keystroke allows you to perform a manual mail polling operation. You will be prompted for a FidoNet node address in the form <net>/<node>. You may also enter a Sysop name if a SYSOP.NDX (or FIDouser.LST for the older Version6 Nodelist) is available to BinkleyTerm in the nodelist directory. Refer to the User's Manual section "Nodelist" for more information. BinkleyTerm will then dial the system, without regard to event schedules, or any flags, (i.e., repeatedly without pauses until successful) and attempt to transact mail with the remote system. This is sometimes termed a "demon" poll.
Alt-P	Poll a node. Pop-up window appears for you to enter the address. This creates an empty Poll attach (.CLO) which causes dialling at the usual intervals set by the 'T' flag, not repeatedly like Alt-M Note: An outbound directory must pre-exist for the Zone (and, optionally, Domain) of nodes you are calling.
Alt-R	This forces BinkleyTerm to restart the first event that should execute at the current time, not including forced events.
Alt-S	Allows the generation of an outbound file attach. When executed, this command will pop-up a window on the screen where you will be prompted to provide information appropriate to creating a file attach.

Keystroke	Function
-----	-----
Alt-T	This is used to shift to Terminal Mode on the fly. By using the Alt-U command of the Terminal Mode, you can flip back and forth between Terminal Mode and Unattended Mode.
Alt-Q	This tells BinkleyTerm to quit the current event, and start the next event that covers the current time period. If there is no such event, Event 0 (the default non-event) will be started. For all practical purposes, nothing happens during event 0, however, BinkleyTerm will still accept incoming mail. The events can be restarted by the Alt-R command.
Alt-W	This command causes the screen to be redrawn with current information. This is handy for situations such as multi-tasking that may have caused the BinkleyTerm screen to become "trashed."
Alt-X	This allows you to exit BinkleyTerm with errorlevel 1. If you are using a batch file, it must capture and act upon the errorlevel properly by terminating the batch file. Refer to the User's Manual section "Control" for additional information.
Alt-Y	Polls the boss node (same as Alt-Y in Terminal Mode).
Alt-Z	Zoom the outbound window. See "ZOOMED OUTBOUND WINDOW KEYSTROKES" on page 69 for details of this window.
C	Means "Call out please", if there is anything waiting to go. If there is something that would not ordinarily be sent during the current event, this command will not cause it to be sent. Use a manual poll for immediate sending to any system under any circumstances. Note that BinkleyTerm automatically checks the outbound area approximately once every 10 minutes; the automated equivalent of this command.
Space	This key may be used to abort polls and to un-blank the screen (when the 'ScreenBlank' option has been enabled). This key is ignored when pressed during BinkleyTerm idle time (waiting for calls)

ZOOMED OUTBOUND WINDOW KEYSTROKES

Keystroke	Function
-----	-----
A	Re-addresses mail by changing the address on the cursor line, using pop-up window for entry.
C D H or N	Change the status of the mail currently shown on the cursor line.
I	Reset the Dial Tries counter for the node shown on the cursor line. You are asked to confirm.
R	Kill requests from node shown by cursor line.
K	Kill all mail to node shown by cursor line.
T	Stop Mail to node shown by cursor line.
G	Get (Request) files from the node shown by the cursor line, allowing requesting multiple files from that node.
Alt-G	Get (Request) files (pop-up window requests the address then filename then password if any)
S	Send files to node shown by cursor line, allowing entry of multiple files to send to that node.
Alt-S	Send files (pop-up window requests address then filename).
P	Create a Poll for the node shown by cursor line.
Alt-P	Poll a node (Pop-up window allows address entry).
Arrow keys	Up/Dn arrow keys move the cursor line.
Page up/dn	Scroll the display.
Home/End keys	Move to start/end of display.
Esc	Cancel the current command if any and return window display to normal.
	Display also reverts after about 1 min if no keys pressed.

MODIFICATION/TRANSLATION OF BINKLEYTERM INTERNAL TEXT

BinkleyTerm now allows you to generate your own translations or modifications of much of the BinkleyTerm internal text.

Enclosed with this release is a file named ENGLISH.TXT, which is the raw English language text created by the authors, and BINKLEY.LNG which is the compiled binary representation of ENGLISH.TXT for BinkleyTerm's use.

To create your own translation of BinkleyTerm's text strings, copy the ENGLISH.TXT file to a file of your choice for editing. Use a standard ASCII flat text file editor (such as DOS EDLIN or your favorite word processor in non- document mode) to edit the file for your needs. The strings should in most cases be self-explanatory. BTLNG now supports '\s' for embedded spaces.

When finished editing the file, use the included language compiler, BTLNG.EXE, to compile the text into binary format for BinkleyTerm's use. The command line for BTLNG.EXE is as follows:

```
BTLNG <source_file> <output_file>
```

For example, to compile the ENGLISH.TXT file, use the following command:

```
BTLNG ENGLISH.TXT BINKLEY.LNG
```

The binary language file must be named BINKLEY.LNG in order for BinkleyTerm to recognize it.

Persons who are qualified (bilingual and sufficiently experienced or educated) to translate ENGLISH.TXT into foreign languages are encouraged to do so and to make your translations available to others through appropriate means.

BinkleyTerm now works internally with function codes, rather than scan codes. This means that ALL keyboard behavior can be modified in the ENGLISH.TXT file. Examples are included in that file to illustrate how this is accomplished. Any key combination you wish remapped should have the original and remapped scan codes preceded by the capital letter "U" for Unattended mode, "T" for Terminal mode, or "A" for Ansi escape sequences. The file should then be recompiled into a language file using BTLNG.EXE

Replace the original BINKLEY.LNG file with your newly compiled revision, and your keyboard remappings will take effect.

A new operating-system conditional switch has been added to BTLNG.EXE, the language file compiler:

?DOS at the beginning of a line will tell BTLNG to compile the line if it's being run under DOS.

?OS2 at the beginning of a line will tell BTLNG to compile the line if it's being run under OS/2.

?WNT at the beginning of a line will tell BTLNG to compile the line if it's the Win32 version.