



Model 4

PostMaster Support Utilities

Published and Distributed by

Computer News 80

PO Box 680

Casper, WY 82602

POSTMASTER SUPPORT UTILITIES

For the Model 4 with High Resolution Graphics

Copyright (c) 1992 by David Gobin. All rights reserved.

Reproduction or use of the POSTMASTER SUPPORT UTILITIES program package or manual without express written permission from the author, David Gobin, is STRICTLY prohibited. While every reasonable effort has been taken in the preparation of this manual and software to assure its accuracy, the author and any authorized distributor assumes no liability for loss of revenues, time, money or properties in any form resulting from the use of this product, or for errors or omissions in this manual or software, or from the use of the information contained within.

Published and Distributed by

COMPUTER NEWS 80

PO Box 680

Casper, Wyoming 82602-0680

Printed and produced in the United States of America.

10 9 8 7 6 5 4 3 2 1

Installation of a High Resolution graphics board on the computer using these utilities is required.

Ownership of David P. Miller's POSTMASTER program is required to take full advantage of these programs.

From time to time the author may choose to enhance or improve this product, either through documentation or through the program. The author reserves the right to apply these enhancements and/or improvements without notice.

* NOTE: The names **POSTMASTER** and **PostMaster** are copyright (c) 1990 by David P. Miller. The names **POSTMASTER** and **PostMaster** are used by permission of their author.

INTRODUCTION

This manual describes several Hi-Res programs which support David P. Miller's excellent POSTMASTER program. This support is devoted to icons, fonts, and borders. Icons are the graphic picture cells which can be added to the labels, posters, etc.. Fonts are the character-related graphics which in most cases resemble the letters which you type in, but in various styles and formats. Borders are those graphics used to surround large labels, posters, etc..

Icon-related utilities are: DSPLIB (/ICN file display program), DSPICN (display single icon program), MAKEICN (create icon image file utility), and ADDICN (add icon image to an /ICN library utility). A bonus file is MASTLIB/ICN, which was created using these utilities, either capturing displayed images, or using the special utilities in the MODEL 4 HI-RES UTILITIES package (available from *Computer News 80*) to reduce screen images to be captured for icon use by Postmaster. If you have Postmaster, you can add this file to your collection of icon libraries, as it is fully compatible with Postmaster. You can preview its contents using the DSPLIB utility.

Font-related utilities are: DECDFON, which will decode a PostMaster font file into a text file which you can edit and modify, ENCDFON, which allows you to compile a modified or created font from its ASCII source format into a PostMaster compatible font file, and ADDFONT, which allows you to add a PostMaster font to a specified map file so the font can be used by PostMaster. An added file for your examination is BLOCK/ASC, an ASCII file which is a decoded rendition of a PostMaster font.

Border-related utilities are: DECDBOR, which allows you to decode a PostMaster border file into an ASCII file which you can edit and modify, ENCDBOR, which allows you to compile a border from its ASCII source format into a PostMaster compatible border file, and AADBOR, which allows you to add a PostMaster border to a specified map file so that the border can be used by PostMaster. An added file for your examination is ANTS/ASC, an ASCII file which is a decoded rendition of a PostMaster border.

SPECIAL NOTICE

Be aware that on programs that require entry parameters, that by simply entering the filename *without* parameters, or by entering the command name with a question mark following it, after a space (i.e., ADDICN ?), you will be presented with a syntax help screen.

DSPLIB

Display /ICN Library Utility

DSPLIB/CMD allows you to view one or more Postmaster icon library files by entering their names after the DSPLIB command, separating each by a space, thus to display the contents of the MASTLIB/ICN file included in this package, you could enter:

DSPLIB MASTLIB/ICN

or:

DSPLIB MASTLIB

This command will load the specified file (MASTLIB/ICN in this case) and display its contents. Please note that a default extension of "/ICN" will be added if you do not specify an extension. DSPLIB will process each icon file by displaying up to 15 icon images to a screen. Thus an entire 30 icon file can be viewed on 2 separate screens.

Each displayed icon will have an number assigned to it, shown to the upper left corner of the image. This numbering is handy if you are planning to replace an icon in a file using the ADDICN utility. You can simply define the desired icon number when running ADDICN to replace the current image. Below each image, up to 15 character of its description are displayed (full descriptions are 22 characters long -- too long for this display format).

Once a screen is displayed, DSPLIB waits for a command from you. Pressing any key will cause operations to resume. If a file contains more than 15 icons, then a second screen will be displayed. Once all icons have been displayed, DSPLIB checks to see if another icon library file is specified. If so then it loads the next file and displays it. If no more files are specified, then control is returned to DOS.

Pressing BREAK during a screen display will abort all other operations and return control to DOS. This is handy if you wish to retain the current image in Hi-Res memory and want to manipulate or capture an image on it.

If a file needs more than one screen to display all of its files, you can press the UP ARROW key from the second screen to redisplay the first again.

If you have more than one icon library file to display, you can list each icon library file after the DSPLIB command and separate them by using a space. Thus to display files such as MASTLIB/ICN and PMARTLIB/ICN, for example, you can use the command:

DSPLIB MASTLIB PMARTLIB

DSPICN

Display Single Icon Utility

DSPICN/CMD will display a single icon from an icon library file in the center of the Hi-Res screen. An ICON parameter allows you to tell DSPICN which icon in the library to display. If no ICON parameter is specified, then the first image in the file is used (handy for checking an image just captured using MAKEICN). This utility was specially designed to allow the user to "recapture" images using MAKEICN when they cannot capture an image centered correctly due to the image being too close to the edge of the screen. DSPICN moves the icon away from the edges so a better job can be done.

Thus, to display the first icon in the MASTLIB/ICN file, you could enter:

DSPICN MASTLIB/ICN

or:

DSPICN MASTLIB

If you wish to display the "BAT" icon in MASTLIB/ICN, which is icon number 9 as indicated by a display given by the DSPLIB command of MASTLIB/ICN, you could use the command:

DSPICN MASTLIB (ICON=9)

or:

DSPICN MASTLIB(I=9

After an icon is displayed (with its description shown beneath it), DSPICN will wait for the user to provide it with instructions. Pressing any key except the function keys will cause control to be returned to DOS.

Pressing the "F1" key will cause the displayed image to be mirrored from left to right. Thus an image facing rightward would afterward be facing left, or vice versa.

Pressing the "F2" key will cause the displayed image to be inverted top to bottom, thus rendering an image upside down (or downside up).

Pressing the "F3" key will cause the displayed icon image to be reversed, switching all set bits off, and all reset bits on, producing a "negative" image of its current form.

MAKEICN

Icon Capturing Utility

MAKEICN/CMD allows you to capture a section of the Hi-Res screen and create an 1-image Postmaster icon file. This file can afterward be merged into another icon library file using the ADDICN utility, or displayed and manipulated using the DSPICN utility (it can also be displayed using DISLIB if you wish).

When you invoke MAKEICN, it activates the current Hi-Res screen and draws an 88 x 52 dot non-destructive box in the upper-left corner of the screen. If you have a Mouse device and a mouse driver installed, such as *MOUSE+* from *Computer News 80* then you can move the box by using the mouse. If you do not have a mouse hooked up, you can use the arrow keys. Use the unshifted keys to perform fine-tuned movement. Use the shifted keys to move at 10X speed. Notice the arrow keys are not recognized if the mouse is active.

In both mouse and keyboard modes, you can use the 3 function keys to specify the size of the capture box. F1 selects the default of an 88 x 52 dot box. F2 selects a 176 x 104 dot box. F3 selects a box which is 352 x 208 dots. Please note that the larger boxes will translate the image down to an 88 x 52 dot picture, which is the size of a Postmaster icon.

Pressing the BREAK key or the middle button on the mouse (both left and right on a 2-button mouse) aborts operations and returns control to DOS.

Pressing the left mouse button or ENTER causes the current image masked by the box (including the data covered by the box's border) to be scaled into an 88 x 52 dot image. You are returned to the normal system screen and asked to enter up to 22 characters to describe the image. You are next asked for a file name. If you simply press ENTER, MAKEICN will try to construct a legal filename from the description, and then prompt you for a drive number to place it on (pressing ENTER again will default to drive :0). A default extension of /ICN will be added to your file if you do not provide one.

Pressing the "@" key or the right mouse button does the same as the left, except that the image captured will be reversed, setting unset bits, and resetting set bits.

Once the file is saved, you are returned to DOS control.

ADDICN

Add Icon To Library Utility

ADDICN/CMD allows you to add the image from a 1-image icon file created with MAKEICN to a library file. You can create a library file by simply naming one of your 1-image icon files to whatever you choose to call your library, since these 1-image file are stored in standard PostMaster Icon Library file format. The single image stored in it will be considered icon # 1 in your library file. Of course, you could also capture an image and name the file whatever you want to call the library.

You add icons to a library file using the command:

ADDICN source TO libraryfile

where "source" is the icon file which contains the icon you wish to add to "libraryfile". Note that the word "TO" is entirely optional, but is helpful in keeping your "directions" right. Using this command, the new icon will be added into the library file in the first free icon slot. Notice that an icon library file is limited to only 30 slots. A default extension of /ICN will be added to all your files if you do not provide one.

You can add more than one source file by listing them one after the other after the ADDICN name. Just remember to type the library file name as the last in the list. For example, to add ICON1/ICN, ICON2/ICN, DRAGON/ICN, and AUTOCLUB/ICN together into a file called MYLIB/ICN, you might choose a strategy like this:

Supposing that MYLIB/ICN does not yet exist, you could create it by using:

RENAME ICON1/ICN MYLIB

or:

COPY ICON1/ICN MYLIB

and then enter the command

ADDICN ICON2 DRAGON AUTOCLUB TO MYLIB

ADDICN will report loading the library file MYLIB/ICN, and then will load each of the other files: ICON2/ICN, DRAGON/ICN, and AUTOCLUB/ICN. Once all add files are loaded, MYLIB/ICN will be updated to disk with the new information. ADDICN will lastly report the number of free icon slot still open in the library file before returning control to DOS.

You can now use DSPLIB to examine your new library.

If you want to replace a pre-existing icon in a library with another icon, you must first know the number of the icon to replace (shown by DSPLIB). Suppose we wanted to replace the DRAGON icon previously stored in MYLIB/ICN with an icon from a file called ANGEL/ICN. Knowing that the DRAGON icon is icon # 3 in the MYFILE/ICN file, we would issue the command:

ADDICN ANGEL TO MYLIB (REPLACE=3)

or:

ADDICN ANGEL TO MYLIB(R=3

The library file will now have its dragon text and icon overwritten by the angel text and icon.

Note that the REPLACE parameter can be abbreviated to just one character.

Also be aware that you can replace *consecutive* icons by listing more than one source file before the library file name. Replacement will begin at the specified REPLACE icon number, and increment for as many new icon files listed.

DECDFON

Decode PostMaster Font Utility

DECDFON/CMD allows you to decode a PostMaster font file, which is a file ending with a /FON file extension, to an ASCII file of the same name, but with an /ASC file extension. ASCII files can be edited by a number of programs, such as Mark Reed's ED-IT, or my own SCRIPT (both from *Computer News 80*).

A PostMaster font file consists of all characters with decimal code from 33 through 126 ("!" through "}"). You can examine these characters and corresponding codes in the Appendix of your DOS manual, or on pages A-6 and A-7 of MOD 4 BY CHRIS from *Computer News 80*.

Using DECDFON is easy. For example, suppose you wanted to decode the PostMaster font file BLOCK/FON. You would enter the command:

DECDFON BLOCK/FON

Please be aware that DECDFON assumes the font file to already contain a /FON extension, and so the command can be shorted to:

DECDFON BLOCK

You can also add a drive number specification. Additionally, if you wanted the destination ASCII file to be created on a drive other than the first available, you can also specify the drive you wish to have it created on. For example, suppose that BLOCK/FON existed on Drive :1 and you wanted the ASCII file (called BLOCK/ASC, by the way) to be created on Drive :2, you could enter the command:

DECDFON BLOCK:1 :2

Feel free to examine the file BLOCK/ASC included on this disk. This is a direct conversion of the BLOCK/FON file found in PostMaster, converted to ASCII format by DECDFON. If you wish to examine it in detail, if you have continuous (fanfold) paper in your printer, enter the command:

LIST BLOCK/ASC (P)

and a hardcopy will be printed of the file. You may need it to fully understand what is required for compiling a legal font file.

The purpose of DECDFON is to decode a PostMaster font for editing and recompiling by ENCDFON, described next in the manual. Refer to that section for more specific rules on font source file formats.

Be aware that decoding a font file, especially a quite large one, can fill up a sizable portion of a single sided double density disk. This possible large size also will make it extremely difficult to edit. You will need to use a file splitting utility, such as the freebee BREAKUP/BAS utility included on the POSTMASTER SUPPORT UTILITIES disk to break the file up into several parts which can be edited individually. Of course you will also have to move blocks from one file to another so that one character definition is left intact with all its other parts. Also be sure to include linking commands at the end of each file if another one is required to follow it.

ENCDFON

Encode A PostMaster Font

ENCDFON/CMD allows you to take an ASCII file following the format shown in BLOCK/ASC, which is included on the *POSTMASTER SUPPORT UTILITIES* disk.

Once a legal ASCII file is created (rules described shortly), Using ENCDFON is easy. For example, suppose you wanted to encode the source code for the PostMaster font file BLOCK/FON, which is contained in the BLOCK/ASC file. You would enter the command:

ENCDFON BLOCK/ASC

Please be aware that ENCDFON assumes the font file to already contain an /ASC extension, and so the command can be shorted to:

ENCDFON BLOCK

You can also add a drive number specification. Additionally, if you wanted the destination font file to be created on a drive other than the first available, you can also specify the drive you wish to have it created on. For example, suppose that BLOCK/ASC existed on Drive :1 and you wanted the compiled font file (called BLOCK/FON, by the way) to be created on Drive :2, you could enter the command:

ENCDFON BLOCK:1 :2

As stated previously in the DECDFON section, feel free to examine the file BLOCK/ASC included on this disk. This is a direct conversion of the BLOCK/FON file found in PostMaster, converted to ASCII format by DECDFON. If you wish to examine it in detail, if you have continuous (fanfold) paper in your printer, enter the command:

LIST BLOCK/ASC (P)

and a hardcopy will be printed of the file.

You may need it to fully understand what is required for compiling a legal font file.

Now examine the first character (on the next page):

L(nextfile)

Do not confuse this link command with the leading blank command, which must always follow the B(x); command. Used in this format, it indicates that the filename entered within the parentheses (replacing "nextfile") is to be immediately linked to, abandoning all further compiling on the current file and focusing its attention on the specified file. Next create a file by the name specified within the parentheses and continue your font definition.

If you have a special interest in creating you own PostMaster font files, be sure to follow the format shown in the example BLOCK/ASC file. Try using ENCDFON on other PostMaster font files in your collection and note how characters are marked. Using Mark Reed's ED-IT, or SCRIPT (both from *Computer News 80*) allows you to also see how each line is ended by displaying the end of line markers.

Also notice that some large fonts cannot be easily edited due to their being too large for some editors. You will need to break these files up between character definitions into two or more files. Be sure to edit in a LINK command at the end of the first file to allow linking into the next file.

You can create fonts that use as many lines or columns as you want. Just be sure that each individual character definition has a uniform line width. Also be aware that PostMaster uses the width of the dash "-" character to define the width of a SPACE character, and the width of the period "." character to define the spacing between characters. Keep this in mind when creating your own fonts.

ENCDFON explicitly expects a consecutive order in the characters you type in between the parentheses on the B() and E() commands, expecting that all characters are defined in actual numerical order (33, 34, 35...). The use of a character, which MUST be used because ENCDFON expects something there, is also for YOUR reference. This is especially handy when you are defining a special symbol set that will not actually display the typed characters. Also be aware that you should not add blank spaces to the right of a line, and that each line must have EXACTLY the same width (except for blank lines). Notice that ALL codes from 33 through 126 MUST be defined. Failure to do so will result is improper operation of the ENCDFON file.

Failure to comply with these rules can result in a faulty font file if compiled using ENCDFON. Following these simple rules will result in trouble-free program operation.

ADDFONT

Add PostMaster Font File to a Library Utility

ADDFONT/CMD allows you to add PostMaster font files (created by ENCDFON or those gathered from The File Cabinet) to a list of font files which can be later be accessed and used from within PostMaster. By default, ADDFONT creates or accesses a file called PMFONTS/MAP. This is the default file used by Version 1 of PostMaster. The next version will allow the existence of more than one map file to exist on a disk by designating font maps with a /FMP file extension. ADDFONT gives you the capability to support that format once Version 2 of PostMaster is released.

To add a font, such as BLOCK/FON to the default PMFONTS/MAP file (this file will be created if it does not already exist), you would use the command:

ADDFONT BLOCK/FON

Since ADDFONT expects all font files to have an extension of /FON by default, this can be shortened to:

ADDFONT BLOCK

You can also specify drive numbers, and include more than one file. For example, suppose you had the files BLOCK/FON, MARTIN/FON, and EDITOR/FON on Drive :1, you could use the command:

ADDFONT BLOCK:1 MARTIN:1 EDITOR:1

To add them to the PMFONTS/MAP file.

If you wanted to create or add the files to the PMFONTS/MAP file on another drive, say drive :2, you could use the command:

ADDFONT >PMFONTS/MAP:2 BLOCK:1 MARTIN:1 EDITOR:1

Notice that the /MAP extension in this case is *required*, as otherwise a default extension of /FMP will be added to it, which is the extension for the font maps in new version of PostMaster. You could "cheat" and just use the disk drive designation ":2" to also create PMFONTS/MAP on Drive :2, since ADDFONT will then fall back onto its default PMFONTS/MAP file, in the form **ADDFONT >:2 BLOCK:1 MARTIN:1 EDITOR:1**.

Using the ">" designator specifies a destination library file. This file can be named anything you choose as long as the filename follows the rules set down in the DOS manual for legal filenames.

PLEASE NOTE that you can add only EIGHT (8) fonts to a font map file.

By default, ADDFONT will create a sample font image using the name of the font file, the first character capitalized, and the others in lower case. If you wish to alter this order, you can do so by following the filename with a quoted string, containing the desired text. For example, suppose you wanted to add a file called HITECH/FON to your library, but you wanted it to be shown in the font selection menu within PostMaster using the format: *Hi-Tech*, you could use the command: **ADDFONT HITECH"Hi-Tech"**. Finally note that you can use up to 22 characters within the quoted string.

DECDBOR

Decode PostMaster Border Utility

DECDBOR/CMD allows you to decode a PostMaster border file, which is a file ending with a /BOR file extension, to an ASCII file of the same name, but with an /ASC file extension. ASCII files can be edited by a number of programs, such as Mark Reed's excellent and versatile ED-IT, or my own SCRIPT (both from *Computer News 80*). Please be aware that a border file can have up to 12 borders combined within it, and so can be quite large. You may need to create or use a utility to break them up into separate files if you wish to edit and recompile them (try the BASIC program BREAKUP/BAS included on this disk).

To use DECDBOR is easy. Suppose you had a border file called ANTS/BOR. To decode this file you could use the command:

DECDBOR ANTS/BOR

By default, DECDBOR expects a file extension of /BOR, and so you do not need to include it. Thus you could have used:

DECDBOR ANTS

You can also add a drive number specification. Additionally, if you wanted the destination ASCII file to be created on a drive other than the first available, you can also specify the drive you wish to have it created on. For example, suppose that ANTS/BOR existed on Drive :1 and you wanted the ASCII file (called ANTS/ASC, by the way) to be created on Drive :2, you could enter the command:

DECDBOR ANTS:1 :2

Be aware that decoding a border file which has a full 12 border files defined in it can fill up a single sided double density disk. This also will make it extremely difficult to edit. You will need to use a file splitting utility, such as the BREAKUP/BAS utility included on the POSTMASTER SUPPORT UTILITIES disk to break the file up into several parts which can be edited individually. Of course you will also have to move blocks from one file to another so that one border grouping is left intact with all its other parts. You may further want to save each border grouping into its own individual file, as this will be what shall be required to recompile them using ENCDBOR/CMD described later.

Once you have decoded a border file, notice that each set of borders is composed of 8 separate parts, defining the four corners and all four sides. Refer to the file format description covered in the ENCDBOR section, covered next.

Feel free to examine the file ANTS/ASC included on this disk. This is a direct conversion of a PostMaster border, converted to ASCII format by DECDBOR. If you wish to examine it in detail, if you have continuous (fanfold) paper in your printer, enter the command:

LIST ANTS/ASC (P)

and a hardcopy will be printed of the file. Please note that since some lines can be over 80 characters wide

POSTMASTER SUPPORT UTILITIES

that you may want to first set your printer up for 12 characters per inch (Elite) or denser.

You may need a listed copy of the file to fully understand what is required for compiling a legal font file.

The purpose of DECDBOR is to decode a PostMaster border file for editing and recompiling by ENCDFON, described next in the manual. Refer to that section for more specific rules on border source file formats.

ENCDBOR

Encode a PostMaster Border File

ENCDBOR/CMD allows you to take an ASCII file following the format shown in ANTS/ASC, which is included on the *POSTMASTER SUPPORT UTILITIES* disk.

Once a legal ASCII file is created (rules described shortly), using ENCDBOR is easy. For example, suppose you wanted to encode the source code for the PostMaster border file ANTS/BOR, which is contained in the ANTS/ASC file. You would enter the command:

ENCDBOR ANTS/ASC

Please be aware that ENCDBOR assumes the font file to already contain an /ASC extension, and so the command can be shorted to:

ENCDBOR ANTS

You can also add a drive number specification. Additionally, if you wanted the destination border file to be created on a drive other than the first available, you can also specify the drive you wish to have it created on. For example, suppose that ANTS/ASC existed on Drive :1 and you wanted the compiled font file (called ANTS/BOR, by the way) to be created on Drive :2, you could enter the command:

ENCDBOR ANTS:1 :2

As stated previously in the DECDBOR section, feel free to examine the file ANTS/ASC included on this disk. This is a direct conversion of a PostMaster border file, converted to ASCII format by DECDFON. If you wish to examine it in detail, if you have continuous (fanfold) paper in your printer, enter the command:

LIST BLOCK/ASC (P)

and a hardcopy will be printed of the file.

You may need it to fully understand what is required for compiling a legal font file.

This file consists of only one group of graphics which comprise the four corners and four sides of a border. Each graphic is defined in a uniform order of Top-Left (TL), Top-Right (TR), Bottom-Right (BR), Bottom-Left (BL), Top-Bar (TB), Bottom-Bar (BB), Left-Bar (LB), and Right-Bar (RB).

The TL, TR, BR, and BL definitions all have the same resolution definitions -- 48 dots horizontal by 26 dots vertical. The TB and BB definitions have resolutions of 88 horizontal dots by 26 vertical dots. The LB and RB definitions have a resolution of 26 horizontal dots by 52 vertical dots.

With this in mind, examine the first definition (on the next page):

B(TL); L(0)

```

.....
.....
.....
.....OO...OO.....
.....O...O.....
.....O...O.....OO
.....OO...O...O...
.....OOOO.....OO...O...O...
.....OOO...O...O...OOO.OO...
.....OO.O.....OO...OO...OOOOOO.O...
.....OO...OO...OO...OOOOOO.O...
.....OO.O...OO...O...O.OOO...
.....O...OOO...OO...OOOOOO.....
.....OO.....OOO...OOOOOO...OOOO.....
.....OO.....OOO...OO...OOOO...
.....OO.O.....OOOOO..O...OO.....OO...
.....OOOOOOOOO..OO...OOO...O...
.....OO..OOOOOOO..OO...OO...OO..
.....O..OOOOOOO.....OO...OO...OO
.....O..OOOOOOO.....OOO...OO.....
.....O.OOOOOOOO.....OO...OO.....
.....OOOOOOO.....OO...OO.....
.....OOOOO.....OO...O.....
.....OO...OO.....OO.....
.....O.....
.....OO.....

```

E(TL)

Hold it at a distance to get a clearer view of the ant crawling on the page. Now notice the heading. It begins with B(TL);. This tells you that we are beginning the definition of the "TL" graphic (Top Left corner). The next part of the header, L(0), tells us that there are no "leading" blank rows. This second part is handy, as we can specify the number of rows of blank lines without having to actually type them in. For example, specifying L(5) would indicate that there are 5 blank lines before the displayed data actually begins. But always keep in mind the dimension restrictions for each graphics definition listed previously.

Next the data begins. Usage of upper case or lower case OHs indicates a bit which is to be set ON, or SET. Periods indicate bits that are to be OFF, or RESET. Notice that blank lines can be entered without any typed data, but can in fact be marked by simple carriage returned (ENTER codes).

Once the graphic is completely defined, the trailer command E(TL) is used. Blank lines can be added for separation before definition of the next character.

SPECIAL NOTES

Unlike font files, you cannot define LINKS. The border must contain only 8 graphic definitions for one border set, and it must be contained entirely within a single file, which is easily handled by most text editors.

Some text editors have restrictions on line width. You will need to use one which can handle wider lines, such as SCRIPT from *Computer News 80*.

If you have a special interest in creating your own PostMaster border files, be sure to follow the format shown in the example ANTS/ASC file. Try using ENCDBOR on other PostMaster border files in your collection and note how graphic definitions are marked. Using Mark Reed's ED-IT, or my own SCRIPT program allows you to also see how each line is ended by displaying the end of line markers.

Unlike fonts, you **cannot** create borders that use as many lines or columns as you want. To reiterate: The TL, TR, BR, and BL definitions all have the same resolution definitions -- 48 dots horizontal by 26 dots vertical. The TB and BB definitions have resolutions of 88 horizontal dots by 26 vertical dots. The LB and RB definitions have a resolution of 26 horizontal dots by 52 vertical dots. Also be sure that each individual character definition has a uniform line width.

ENCDBOR expects the graphics definitions to be in an exacting order: TL, TR, BR, BL, TB, BB, LB, and RB. Do not try to deviate from this, or an error will be reported by ENCDBOR and operations will stop. The use of the L(0) designator after the B() command is mandatory, even if you are fully defining a graphic, as has been done with the ANTS/ASC file. However, in creating your own borders, you can make good use of it. This is especially handy when you are defining a special symbol set that will not actually display the typed characters. Also be aware that you should not add blank spaces to the right of a line, and that each line must have EXACTLY the same width (except for blank lines, which can **NEVER** begin a definition). Notice that ALL graphics MUST be defined. Failure to do so will result in improper operation of the ENCDBOR file.

Failure to comply with these rules can result in a faulty border file if compiled using ENCDBOR. Following these simple rules will result in trouble-free program operation.

ADDBOR

Add PostMaster Font File to a Library Utility

ADDBOR/CMD allows you to add PostMaster border files (created by ENCDBOR) to a list of border files which can be later be accessed and used from within PostMaster. By default, ADDBOR creates or accesses a file called PMBORDER/MAP. This is the default file used by Version 1 of PostMaster. The next version will allow the existence of more than one map file on a disk by designating border maps with a /BMP file extension. ADDBOR gives you the capability to support that format once Version 2 of PostMaster is released.

To add a border, such as ANTS/BOR to the default PMBORDER/MAP file (this file will be created if it does not already exist), you would use the command:

ADDBOR ANTS/BOR

Since ADDBOR expects all font files to have an extension of /BOR by default, this can be shortened to:

ADDBOR ANTS

You can also specify drive numbers, and include more than one file. For example, suppose you had the files ANTS/BOR, XMAS/BOR, and THICK/BOR on Drive :1, you could use the command:

ADDBOR ANTS:1 XMAS:1 THICK:1

To add them to the PMBORDER/MAP file.

If you wanted to create or add the files to the PMBORDER/MAP file on another drive, say drive :2, you could use the command:

ADDBOR >PMBORDER/MAP:2 ANTS:1 XMAS:1 THICK:1

Notice that the /MAP extension in this case is *required*, as otherwise a default extension of /BMP will be added to it, which is the extension for the new version of PostMaster. However, if you were to simply specify the drive number ":2", then ADDBOR would fall back onto its default file designation of PMBORDER/MAP.

Using the ">" designator specifies a destination library file. This file can be named anything you choose, as long as the filename follows the rules set down in the DOS manual for legal filenames.

SPECIAL NOTES

You can add only TWELVE (12) borders to a border map file.

When the /MAP file is created, a special /BOR file is also created. In the above examples, when PMBORDER/MAP is created, another file called BORDERS/BOR is also created, which will contain all border files added to the /MAP file (the original /BOR files are left untouched). If you create a /BMP file by specifying a name other than PMBORDER after the ">" symbol (and excluding a file extension), then two files with the specified name, one with a /BMP extension, and the other with a /BOR extension will be created. These are the files which will be used by the new version of PostMaster. Therefore, be careful not to name a library the same as one of your compiled border files.

By default, ADDBOR will create a sample border image in the /MAP file and use the name of the border file, the first character capitalized, and the others in lower case, to be used in the text portion of the border selection menu in PostMaster. If you wish to alter this to make the text in the menu selection reflect a more descriptive name, you can do so by following the filename with a quoted string, containing the desired text. For example, suppose you wanted to add a file called ANTS/BOR to your library, but you wanted it to be shown in the border selection menu within PostMaster using the name: Bugs, you could use the command:

ADDBOR ANTS"Bugs"

You can use up to 22 characters within the quoted string.

