

RGB^TechWriter

6. MODIFYING TW.DAT

When RGB^TechWriter is invoked by executing the program TW.EXE, the first thing it does is to locate the file TW.DAT and read it to obtain the default editing parameters and the codes necessary to drive your printer. RGB^TechWriter searches for the file TW.DAT in the following places, in the following order:

1. In the current directory on the current drive,
2. In the directory \TW on the current drive,
3. In the root directory on the current drive.

The first version found is used. If you have a fixed-disk system with several users, each person can maintain their own customized copy of TW.DAT in their own subdirectory, or can rely on the master version stored in the \TW directory. If RGB^TechWriter can't find TW.DAT in any of the above locations, a set of default values is assumed which corresponds to an IBM-type printer without special character support.

If TW.DAT is found but an error occurs while reading the file, an error message is generated and program execution is aborted. To correct the file TW.DAT you can either use another editor, like EDLIN, or you can follow this procedure using RGB^TechWriter:

1. Using DOS, rename TW.DAT to something else, say TEMP.DAT, by changing to the directory which contains tw.dat and typing
rename tw.dat temp.dat
2. Invoke RGB^TechWriter by typing
tw
3. RGB^TechWriter will not find TW.DAT, so it will use default values.
4. Get TEMP.DAT and correct the file to match the description given in the following sections. Be certain that the line numbers match. Also, make certain that the version number of the printer file is version 2. Earlier configuration file versions will not work with this release of RGB^TechWriter.
5. Save the corrected file as TW.DAT.
6. Delete the old file TEMP.DAT.
7. Exit to DOS.

The next time you invoke RGB^TechWriter, the corrected file will be read successfully and the parameters set accordingly. Note that any changes made in the file TW.DAT do not take effect until the next time RGB^TechWriter is invoked from DOS.

RGB^TechWriter

RGB^TechWriter reads TW.DAT based on specific line numbers. The first 4 lines are ignored, so you can put any comments you want on these lines. After that, every odd line through line 87 is assumed to be a comment and is skipped over. The even lines must have properly formatted data. In each odd/even pair of lines, the odd line labels the data which lies on the following even line. You can modify the odd lines if you wish. The even lines 6 through 32 are editor parameters, described in section 6.1. The even lines 34 through 86 are printer codes, described in section 6.2. Line 87 begins an optional section of alternate character definitions, described in section 6.3.

6.1 Editing Defaults

All of the editing parameters like margins and line spacing can be modified from within RGB^TechWriter using the Margin command (M) from the file screen menu. If you wish, you can modify TW.DAT so that the values you use most often become the initial default values. The file TW.DAT can be read, edited, and revised using RGB^TechWriter. The following discussion assumes you have located TW.DAT, read the file into text memory, and are ready to begin editing.

Line Description

- | | |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6 | Maximum Text Length. This controls how much memory space is allocated to store the text. The default is 62768 characters. Two bytes of memory are required for each character, one for the character itself and one for its color. If there isn't enough free memory available, RGB^TechWriter will warn you, then automatically decrease the maximum text length as needed. If you get this warning often, you will want to set this value to a smaller number which will fit into the amount of memory you have available. |
| 8 | Pitch. This is the number of characters per inch RGB^TechWriter will use when printing the file. The allowed values are 10, 12, and 17. The printer must support the pitch selected for proper operation. |
| 10 | Line Width. This is the maximum number of characters allowed on a line. Any attempt to enter more will cause the last word to wrap around to the next line. This value divided by the pitch determines the length of each line in inches. The value on this line must be an integer value less than 256. |
| 12 | Left Margin. This is the total left margin in inches. It will be rounded to the nearest character column when printing the file. |

RGB^TechWriter

- 14 Top Margin. This is the blank space above the top line of text on each page, in inches. The header and page number appear 1/2 inch above the top margin. The top margin must be greater than or equal to 0.5 inch, otherwise there wouldn't be room for the header.
- 16 Text Length. This is the length of text allowed on each page, in inches. This is exclusive of the header and footer, which each add an extra 0.5 inch to the top and bottom. In combination with a given top margin and paper length, this value determines the bottom margin.
- 18 Paper Length. This is the length of the printable region on each piece of paper, in inches. For continuous-form paper this will be the actual length of the paper (usually 11 inches). RGB^TechWriter will send a series of carriage returns to cross the perforation between pages. Therefore, do not enable the automatic perforation-skip mode if your printer has one. The paper length must be set at least one inch greater than the text length for proper operation. Alternatively, you can enter 0 for the paper length. In that case, RGB^TechWriter will issue a form-feed command to your printer at the start of each page. This alternate method is recommended for cut-sheet printers (like most laser printers).
- 20 Line Spacing. This can be any value greater than 0, up through 4, including fractional values like 1.79. Single spacing (1.0) corresponds to 6 lines per inch. Other values are inversely proportional, so that triple spacing prints 2 lines per inch.
- 22 Page Number Location. Three page number locations are supported. Enter 0 to get numbers centered at the top of the page, 1 for numbers centered at the bottom, and 2 for numbers in the upper right corner. Page numbering must be turned on with the appropriate dot command for this to take effect.
- 24 Delay Time. This is an integer that gives the number of milliseconds that warning and advisory messages appear on the screen. A nominal value is 1000 (1 second).
- 26 Stall Time. This is an integer that gives the number of milliseconds RGB^TechWriter waits after a pause in your typing to fully update the screen. A nominal value is 200. If set too low, RGB^TechWriter will interrupt your typing to update the screen, while too high a number will mean an unnecessary delay once you do pause. Fast typists may want to decrease this value.

RGB^TechWriter

- 28 Default Path. This is a string of characters which identifies which disk drive and which directory path RGB^TechWriter should use for file storage when it is first invoked. A drive letter followed by a colon is mandatory. This can be followed by an optional directory path specification. If only the drive letter is supplied, the default directory on that drive is used. Initially set for drive A: when shipped, most users will want to change this value to B: for a two-diskette system, or C: for systems having a fixed-disk. If you always use the same subdirectoy for your RGB^TechWriter files, you will want to add that directory path.
- 30 Minimum Disk Margin. This is a distance in inches. When RGB^TechWriter prints a file to disk, it is usually intended for uploading to another computer system. The printer for the other system has some minimum left margin, such that the character in the first printable column still has some blank space to its left. Typically this value is 0.25 inches. RGB^TechWriter will subtract this value from the desired left margin to determine how many blank spaces to put at the start of each line.
- 32 Maximum Disk Column. This is an integer number. When RGB^TechWriter prints a file to disk, it is usually intended for uploading to another computer system. The other computer will probably impose a limit on the maximum length of each line. A typical value is 255, although some systems may limit lines to 132 characters. During a print-to-disk, RGB^TechWriter will truncate any lines which exceed this limit.

6.2 Printer Codes

Unlike the editing defaults, which can be changed from within the program, the printer codes can only be changed by modifying TW.DAT. Try printing the file GUIDE7.TWF, found on the RGB^TechWriter distribution diskette. This file serves as a test for printing many of RGB^TechWriter's features. If your printer does not reproduce one or more of the features as expected, either your printer is not capable of that function or you are sending it the wrong command codes. If you need to know exactly what codes RGB^TechWriter is sending to your printer, you can have the program redirect the printer output to a special disk file, the contents of which you can examine in detail. This is not the same as requesting the "Print-to-Disk" option. To create this special disk file, name your file PRINTEST.TWF. When you try to print this file to the printer, it will instead go onto your disk with the name PRINTEST.

RGB^TechWriter

Several of the codes in the printer configuration file TW.DAT are in the form of what will be termed "ASCII equivalent strings." These are a series of integer numbers, separated by spaces. The numbers must be in the range 0 through 255. These numbers represent the decimal equivalent of the ASCII characters which will be sent to your printer. Thus a 33 in the list will send a CHR\$(33) to your printer. A list of ASCII numerical equivalents is in your BASIC manual. In TW.DAT, all ASCII equivalent strings are limited to no more than 80 separate values, and all of the values must fit on a single 255-column line. A null string (i.e. don't send anything) can be entered by leaving the line completely blank except for the carriage return marker dot.

If you develop a configuration file for a printer not currently supported by RGB^TechWriter, please mail a copy to BA^EL Software. It will be included in future releases, with proper credit to you, of course.

Line Description

- 34 Initial Top Margin. This is a value in inches which tells RGB^TechWriter where the printhead is initially located with respect to the top edge of the paper. On most continuous-form printers, the rollers which guide the paper lie "above" the printhead. In order to locate the printhead at the very top of a page it is necessary to waste the preceding sheet of blank paper. RGB^TechWriter allows you to start with the printhead below the top edge of the page. RGB^TechWriter has no way of knowing where the paper is located with respect to the printhead; it is up to you to set this properly before each printout. It is advisable to set this value to 0 for cut-sheet printers. There are two situations where RGB^TechWriter may have to skip to the top of the next page and thereby waste a sheet of paper anyway. One is if you request a header or top-margin page number on the first page and the initial top margin is set greater than 0.5 inches. The other is if you are selectively printing only part of the text starting after page 1 and the initial top margin is set greater than 0.
- 36 Master Printer Reset. This is an ASCII equivalent string which is sent to your printer prior to printing the file. You can use it to clear any previously set functions to ensure a fresh start. Some printers are slow to reset or may require a minimum delay after reset before proceeding. It may be best to leave this line null in such cases.
- 38 Printer Vertical Resolution. This is a value which represents the resolution your printer allows for positioning the printhead on the page in the vertical direction. This value is given in lines per inch. Smoothest results are obtained if your printer's resolution

RGB^TechWriter

is an even multiple of 72. Other values are accepted, even fractions, but it may be necessary to set line 86 to compensate. This value is used to calculate the vertical motion index (VMI).

- 40 Vertical Motion Index Prefix. This is an ASCII equivalent string which is sent as the first part of each attempt to change the vertical motion index (VMI), which is the distance your printer moves the paper when a line feed is sent. The VMI is set at the beginning of the printout and whenever the line spacing changes, including lines with superscripts, subscripts, underlining, or alternate characters.
- 42 Vertical Motion Index Digits. This is an integer value which defines the number of ASCII digits used to convey the new VMI setting to your printer. If this value is 0, RGB^TechWriter will send a single byte whose value is the VMI setting. Thus if the vertical resolution is 144 lines per inch and we want to move 6/144" down the page, the printer will be sent "006" if this line were set to 3, or CHR\$(6) if it were set to 0.
- 44 Vertical Motion Index Decimals. This is an integer value which specifies the number of digits which may be used to the right of the decimal place when specifying the VMI. Continuing the example from line 42, if we actually wanted to move down 6.176/144", and this value was set to 4, the printer would be sent "006.1760" as a part of the VMI-setting command. If this line is set to zero, not only are no decimal digits sent, but neither is the decimal point itself. If line 42 is set to 0, this entry has no meaning. The sum of lines 42 and 44 cannot exceed 20.
- 46 Vertical Motion Index Offset. This is an integer value which is added to the actual number of VMI lines wanted prior to sending the request to your printer. This is necessary because some printers, upon receiving a value of n, will set the VMI to n-1.
- 48 Vertical Motion Index Suffix. This is an ASCII equivalent string which is sent following the numerical VMI data. Some printers require control codes after the VMI data to activate the new value.
- 50 Printer Horizontal Resolution. This is a value which represents the resolution your printer allows for positioning the printhead on the page in the horizontal direction. This value is given in dot columns per inch. A typical value is 120. This value is used only by the right-justification routine to set the horizontal motion index (HMI). If your printer does not support HMI movements, enter 0 on this line. RGB^TechWriter will then add whole spaces between words to justify the right margin.

RGB^TechWriter

- 52 Horizontal Motion Index Prefix. This is an ASCII equivalent string which is sent as the first part of each attempt to change the horizontal motion index. When justification is turned on and line 50 is greater than 0, the HMI is set at the start of each word and after switching between character sets. If line 50 is zero, this value has no meaning.
- 54 Horizontal Motion Index Digits. This is an integer value which defines the number of ASCII digits used to convey the new HMI setting to your printer. If this value is 0, RGB^TechWriter will send a single byte whose value is the HMI setting. Thus if the horizontal resolution is 120 dot columns per inch and we want to have the character columns spaced by $11/120$ ", the printer will be sent "011" if this line were set to 3, or CHR\$(11) if it were set to 0.
- 56 Horizontal Motion Index Decimals. This is an integer value which specifies the number of digits which may be used to the right of the decimal place when specifying the HMI. Continuing the example from line 54, if we actually wanted to space the columns $10.53/120$ ", and this value was set to 4, the printer would be sent "010.5300" as a part of the HMI-setting command. If this line is set to zero, not only are no decimal digits sent, but neither is the decimal point itself. If line 54 is set to 0, this entry has no meaning. The sum of lines 54 and 56 cannot exceed 20.
- 58 Horizontal Motion Index Offset. This is an integer value which is added to the actual number of HMI dot columns per character column prior to sending the request to your printer. This is necessary because some printers, upon receiving a value of n, will set the HMI to n-1.
- 60 Horizontal Motion Index Suffix. This is an ASCII equivalent string which is sent following the numerical HMI data. Some printers require control codes after the HMI data to activate the new value.
- 62 Quality Character Select. This is an ASCII equivalent string which is sent at the beginning of the printout and after each string of alternate characters, provided you selected Quality mode for the printout. If there is no separately distinguishable high-quality character font, leave this string null. In that case, you will not be prompted at print time to select between data and quality modes, and the Data Character Select code will be used. If a zero is encountered in the ASCII equivalent string, it is interpreted as a request to reset the pitch.

RGB^TechWriter

- 64 Data Character Select. This is an ASCII equivalent string which is sent at the beginning of the printout and immediately after each string of alternate characters, provided you selected Data mode for the printout. If your printer has only one character set, you may not need to have anything on this line. If a zero is encountered in the ASCII equivalent string, it is interpreted as a request to reset the pitch.
- 66 Alternate Character Select. This is an ASCII equivalent string which is sent prior to any group of consecutive alternate-set characters. If your printer has a separate alternate character set, you will want to put the code to invoke it here. See section 6.3 for more information about alternate characters. If your printer does not have an alternate character set, you can leave this line null. If a zero is encountered in the ASCII equivalent string, it is interpreted as a request to reset the pitch.
- 68 10 Pitch Select. This is an ASCII equivalent string which is sent to the printer at the beginning of the printout (right after the master printer reset) to select 10 characters per inch. It is also sent when a zero is encountered in the Character Set Select code.
- 70 12 Pitch Select. This is an ASCII equivalent string which is sent to the printer at the beginning of the printout (right after the master printer reset) to select 12 characters per inch. It is also sent when a zero is encountered in the Character Set Select code.
- 72 17 Pitch Select. This is an ASCII equivalent string which is sent to the printer at the beginning of the printout (right after the master printer reset) to select 17 characters per inch. It is also sent when a zero is encountered in the Character Set Select code.
- 74 Minimum Margin. The leftmost column on any printer still leaves some blank space to its left. This value is that distance in inches. It is used in conjunction with the requested total left margin to determine the number of columns to skip at the left edge of each printed line.
- 76 Printer Width. The maximum printable width on your printer, in inches. A typical value is 8.00. RGB^TechWriter will truncate any lines which extend beyond this value.
- 78 Start Bold. RGB^TechWriter prints bold characters by making an additional pass over the line and only printing the bold characters. This ASCII equivalent string is sent prior to the start of the extra pass. This string can be used either to turn on the printer's offset/overstrike mode, or to perform a small horizontal offset directly.

RGB^TechWriter

- 80 Stop Bold. This ASCII equivalent string is sent at the conclusion of the extra pass made for bold characters.
- 82 Start Underline. RGB^TechWriter makes a separate pass of the line to do underlining. This ASCII equivalent string is sent at the beginning of each occurrence of underlining within the line. If your printer does not support continuous underlining, or it does but it doesn't underline spaces, leave this line null. RGB^TechWriter will then send underscore characters to generate the underline.
- 84 Stop Underline. This ASCII equivalent string is sent at the end of each group of underlined characters.
- 86 Underline Offset. This is an integer value, which determines how far down RGB^TechWriter offsets the printhead between the normally printed line and the extra pass to do underlining. This value must be in the range of zero through 4. This value should be set to insure that the underlining will not run through character descenders. Printers with vertical resolutions in line 38 which are not multiples of 72 and which do not allow for a decimal representation of fractional VMI in line 44 may need a particular value in this line to insure that the correct number of lines are printed per page. Specifically, if the resolution is R and the underline offset is U, the requirement is:

$$2*\text{round}(4*R/72)+\text{round}(U*R/72)+\text{round}((4-U)*R/72) = \text{round}(12*R/72).$$

6.3 Alternate Character Codes

Line 87 is the title line which signifies the beginning of a series of optional additional lines which can be used to specify translation codes for the alternate character set. The basic format for each line starting with line 88 (both even lines and odd lines) is:

a 12 14 84 124 5

The first column contains the letter which is to be translated. The case illustrated is invoked whenever an alternate letter "a" is encountered. This is followed by an ASCII equivalent string (translation string) which contains the series of codes which will be sent to the printer. The sequence of events is as follows. First the alternate-character-set select code from line 66 is sent. This may include a zero, which is a request to send the pitch reset code from line 68, 70, or 72. If right justification is turned on and line 50 is greater than zero, this is followed by the HMI-setting command from lines 52 through 60. After that, the translation string for the particular alternate character is sent. If justification is turned on and line 50 is greater than zero, this is followed by the HMI-setting command

RGB^TechWriter

from lines 52 through 60. This is followed by the primary character-set select code from line 62 or 64, which may include a request to reset the pitch. If justification is turned on and line 50 is greater than zero, the HMI-setting command is sent one last time before printing the next character.

There is no limit to the number of alternate-character translation lines which can be added to the end of TW.DAT. The end of the list is determined when either a blank space is found in the first column or the end of the file is encountered. If more than one translation is provided for the same alternate character, only the last entry is active. Any alternate characters which are not found in the translation table will be sent to the printer as the character itself. That is, when an alternate "a" is encountered, CHR\$(97) is sent to the printer if no entry for "a" is found.

The translations can be quite involved, but must be limited to 80 bytes of data, and must fit on a 255-column line. For example, a single alternate character could be translated to print one character, backspace, then print another character on top of the first. Or, codes could be sent to put the printer in dot-graphics mode, followed by the series of codes necessary to "paint" the character on the paper dot-by-dot. This is particularly useful for creating special symbols on printers which do not directly support an alternate character set. If you do this, it is advisable to start out the string with a space character (32). This properly locates the print location at the right-hand edge of the character cell, taking into account the pitch and HMI setting. Before painting the character (say it is n dot columns wide), send the command necessary to back up the print location to the left by n dot columns. Note that laser printers tend to not move the print "cursor" location when dot graphics are sent, in which case the space character should be sent at the end of the string, instead of the beginning. You can not assume anything about what the vertical motion index is set to prior to your alternate character. If you need to move vertically, set the VMI first. RGB^TechWriter will reset the VMI before printing the next line. Nothing prevents you from translating a single alternate character into an entire word or phrase, but to do so would destroy RGB^TechWriter's match between columns on the screen and columns on the printed page. Some suggested alternate character definitions are given in the documentation file "GUIDE7.TWF". It is a good idea to test your alternate character definitions by printing this file at different pitch settings and both with and without right justification.

Don't forget that you can redefine any of the alternate characters from within a file using the .ALT printer dot command. And remember that any changes you make to TW.DAT using RGB^TechWriter will not take effect until the next time you invoke RGB^TechWriter from DOS.

