

## TURN OFF THAT FILTER

Rich Regent (K9GDF, our ARRL Section Manager), sent me a "puzzlement" recently with a request for an explanation. The situation involves sending email messages with attachments. According to Rich, attachments are files such as WordPerfect documents or Lotus documents that ride "piggyback" along with an email message. The recipient removes the attachment from the message, and then loads the attachment with the appropriate program for viewing or further processing. The problem is that some folks get the attachment garbled, while others have no problem. Rich claims that there is something in the vendor's communications that encrypts or scrambles the attachment.

While the vendor's (Exec-PC or other source of your email messages) communication program could certainly be the culprit, it is not likely. Rather, your own communication program is probably the source of the problem.

When email is coming down the phone line into your modem, your communication program examines each character as it is received. If the character is a letter, number or common symbol (a-z, A-Z, 0-9 or a punctuation mark), there is no problem. These are the standard ASCII (American Standard Code for Information Exchange) characters, numbered in the decimal system between 32 and 127. The characters below 32 (0 through 31) are standard ASCII control codes, which signify such things as a carriage return, form feed, and so on).

Lets examine a couple of the standard ASCII characters (only a few, though, because a complete table would take too much space). Here they are, in decimal and binary:

ASCII CHARACTER	DECIMAL VALUE	BINARY VALUE 87654321	DEFINITION
^G	7	00000111	Rings a bell or beeps.
space	32	00100000	Same as touching spacebar.
,	44	00101100	Comma.
7	55	00110111	The number seven.
?	63	00111111	Question mark.
A	65	01000001	Capital A.
a	97	01100001	Lower case a.
z	122	01111010	Lower case z.
~	126	01111110	Tilde mark.
del	127	01111111	Same as touching delete key.
Ç	128	10000000	An extended ASCII character.
ü	129	10000001	An extended ASCII character.
ƒ	135	10000111	An extended ASCII character.

Look especially at the binary representation, which as I have said in previous articles, can be thought of as a series of 8 on/off switches, numbered as shown in the column head. Each character is represented by 8 zeros and ones. A one can be thought of as a switch that is ON, while a zero is a switch thrown to the OFF position. Note that not one of the standard ASCII characters (those up to decimal 127) has a one in the 8th (leftmost) position of the binary representation. They all have zeros in that position. Only the extended ASCII characters (beginning at decimal 128 and up) have the 8th "switch" in the ON position. Note also that Control-G (^G, bell) and decimal character 135 are exactly the same, except that 135 has the 8th bit on.

Many BBSs that communicate only in standard ASCII tell you to set your communications software parameters to 7 data bits, no parity, one stop bit (7,N,1). What that means is that the 8th data bit will be ignored or always treated as a zero. Given those settings, if the BBS sends you a decimal 7 ASCII character followed by a decimal 135 extended ASCII character, the eighth bit will be considered a zero and you will hear two beeps from your computer. Since the eighth bit is considered zero, your communications program will interpret both characters as a decimal 7, the control character known as BEL because it rings a bell (actually, it causes a beep). On the other hand, if your settings were 8, N, 1, you would hear a single beep and the funny looking character ç would appear on your screen. In this case, the eighth bit has not been ignored and the two characters are interpreted as two different characters.

Well now, how the heck does that apply to a garbled word processing or spreadsheet file? The key is that all true word processors (WordPerfect, WordStar, Microsoft Word, etc.) and all spreadsheet programs use the 8th bit for their own purposes. For example, in WordStar, the lower case letter d is represented normally, unless it is the last letter in a word. In that case, it is represented by decimal character 228, which comes out on the screen as the Greek letter sigma ( $\Sigma$ ) when viewed from DOS. However, when viewed in WordStar, it looks like a perfectly fine letter d at the end of a word (as in the d at the end of the word: word).

Has the light bulb gone on in your head? Of course! Any communication program that filters out that 8th bit will turn a word-processed document or spreadsheet document into gibberish. How to fix it? Make sure your communications parameters are set to 8, N, 1 - not 7, N, 1. Some communications programs also have questions in English that ask something like: "Strip the 8th bit?" Answer NO. Others have what they call a translation table, which translates a character with the 8th bit on to another, standard ASCII character. Make sure you turn the "Translation Table" to OFF. All or one of these steps should cure your garbled attached files.

Thanks to K9GDF for asking a question that was the basis of this month's article. Maybe YOU have a question that might lead to the same result. Let me know. Happy computing!