There is a bug in Ventura 2.0's WordPerfect 5.0 load module that causes two problems:

1) Ventura strips extended ASCII characters 128-168 when loading a WP 5.0 text file.

2) When saving back to WP 5.0, Ventura saves extended ASCII characters 128-168 in WP 4.2 format rather than 5.0 format.

You can get around the first problem by replacing the extended ASCII characters with the corresponding Ventura international characters (e.g., change ¥ to <157> in WP 5.0). While this is a nuisance -- especially if you use a variety of those characters -- you will only have to do it once. A simple macro may be in order.

The second problem is a bit more nasty. If you don't want to concern yourself with the technical details just copy the files FIXWP50.BAT and 42-50.COM to an the directory or floppy containing the damaged text files, and issue the following command:

FIXWP50 *filename*
(where *filename* is the WP file that has been saved by VP 2.0).

For example, if the file is NEWS.WP, the command would be:

```
FIXWP50 NEWS.WP
```

The original NEWS.WP will be renamed NEWS.BAD (for backup), and the new NEWS.WP will be the repaired version.

What's happening

To explain what's going wrong, and what this simple program does, let's suppose that we've created a file NEWS.WP in WordPerfect 5.0, and that there is a yen symbol (¥) in the middle of it.

How WP 5.0 stores ¥: WP 5.0 stores ¥ as the four byte sequence (in hexadecimal) **C0 0C 04 C0**. The **C0** bytes at the beginning and end of the sequence delimit the character data; the **04** byte says WP character set #4; the **0C** byte says character number 12 (hex 0C = decimal 12).

How WP 4.2 stores ¥: WP 4.2 stores ¥ as the three byte sequence (in hex) **A1 9D A1**. The **A1** bytes at the beginning and end delimit the character; the **9D** byte says "extended ASCII character 9D" (hex 9D = decimal 157).

After saving NEWS.WP, we exit WordPerfect 5.0 and enter Ventura Publisher. We then load or create a chapter that pulls in NEWS.WP. As Ventura reads in the text, its WP 5.0 load module can't understand the **C0 0C 04 C0** sequence, so it just strips it.

Noticing that the character has been stripped, we key it in again in Ventura (by holding the Alt key down and typing 157 on the numeric keypad). Then we save the chapter and exit Ventura.

As Ventura is saving NEWS.WP in WP 5.0 format, it comes across the ¥. Again, the WP 5.0 load module doesn't know the proper byte sequence for it, so Ventura does a very strange thing: It saves ¥ as **A1 9D A1** (the WP 4.2 code sequence!).

Now we enter WP 5.0 and retrieve NEWS.WP. WP pulls in all of the text until it gets to **A1** (the first byte of the 4.2 ¥ code), and then stops. If we cursor down to that spot, WP does strange things on the screen.

Worried, we exit WP 5.0 (without saving) and go back into Ventura to find the missing text. As Ventura retrieves NEWS.WP, it comes across the **A1 9D A1** sequence it saved earlier, but it can't understand it. Ventura either goes into an endless loop or stops loading NEWS.WP, truncating it at that point.

At this point, we decide to write a letter to Xerox, encouraging them to fix the bug. In the meantime, we exit Ventura and run FIXWP50, which replaces the errant 4.2 code (hex **A1 9D A1**) with the proper VP international code (ASCII **<157>**). We can then load NEWS.WP into either Ventura or WP, although subsequent saves by Ventura will require another run through FIXWP50.