Why is the OEM code page often called ANSI?

devblogs.microsoft.com/oldnewthing/20051027-37

October 27, 2005

Raymond Chen

It has been pointed out that the documentation for the cmd.exe program says

> /A Causes the output of internal commands to a pipe or file to be ANSI

even though the output is actually in the OEM code page. Why do errors such as this persist?

Because ANSI sometimes means OEM.

The "A" versions of the console functions accept characters in the OEM code page despite the "A" suffix that would suggest ANSI. What's more, if you call the `SetFileAPIsToOEM` function, then "A" functions that accept file names will also interpret the filenames in the OEM code page rather than the ANSI code page.

"There are two types of people in the world: Those who believe that the world can be divided into two types of people, and those who do not."

There are those who mentally divide the world of characters into two groups: Unicode and 8-bit. And as you can see, many of them were involved in the original design of Win32. There are "W" functions (Unicode) and "A" functions (ANSI). There are no "O" functions (OEM). Instead, the OEM folks got lumped in with the ANSI folks.

There are also those who realize the distinction, but out of laziness or convenience often use "ANSI" as an abbreviation for "an appropriate 8-bit character set, determined from context". In the context of console programming, the appropriate 8-bit character set is the OEM character set.

The person who wrote the online help for cmd.exe clearly meant ANSI to mean "That thing that isn't Unicode."

> /A Causes the output of internal commands to a pipe or file to be ANSI
> /U Causes the output of internal commands to a pipe or file to be Unicode

I'll leave you to decide whether this author belongs to the "Everything is either Unicode or ANSI" camp or the "just being casual" camp.

Related: [Keep your eye on the code page](#).

[Raymond Chen](#)

Follow