Why doesn't b match word boundaries correctly?

devblogs.microsoft.com/oldnewthing/20110804-00

August 4, 2011

Raymond Chen

A colleague of mine was having trouble getting the `\b` metacharacter in a regular expression to work. Of course, when somebody asks a question like that, you first have to establish what their definition of "work" is. Fortunately, he provided some examples:

`Regex.IsMatch("foo", @"\b" + @"foo" + @"\b")`	true
`Regex.IsMatch("%1" , @"\b" + @"%1"  + @"\b")`	false
`Regex.IsMatch("%1" , @"\b" + @"\%1" + @"\b")`	false
`Regex.IsMatch("%1" , @"\b" + @"\%1" + @"\b")`	false
`Regex.IsMatch("%1" , @"..")`	true
`Regex.IsMatch("%1" , @"%1")`	true

"The last two entries are just sanity checks to make sure I didn't make some stupid mistake like passing the parameters in the wrong order. I want to search for a string that contains `%1` with word boundaries on either side, something I would normally use `\b` for. Is there something special about the % character? Notice that the match succeeds when I look for the word `foo`." Everything is working as it should. Recall that the `\b` metacharacter matches when there is a `\w` on one side and a `\W` on the other, where the beginning and end of the string are treated as if they were `\W`. The string `%1` therefore breaks down as

virtual `\W`	beginning of string
`\W`	% is not an alphanumeric or _
`\w`	1 is a digit
virtual `\W`	end of string

The only points where `\b` would match are immediately before and after the 1, since those are the transition points between `\w` and `\W` and vice versa. In particular, the location immediately before the percent sign does not match since it is surrounded by `\W` on both sides.

My colleague responded, “D’oh! I keep forgetting that % won’t act like a `\w` just because I want it to.”

Raymond Chen

Follow