

# The great thing about regular expression engines is that there are so many to choose from

 [devblogs.microsoft.com/oldnewthing/20090806-00](http://devblogs.microsoft.com/oldnewthing/20090806-00)

August 6, 2009



Raymond Chen

Back in the days before perl ruled the earth, regular expressions were one of those weird niche features, one of those things that everybody reimplements when they need it. If you look at the old unix tools, you'll see that even then, there were three different regular expression engines with different syntax. You had `grep`, `egrep`, and `vi`. Probably more. The `grep` regular expression language supported character classes, the dot wildcard, the asterisk operator, the start and end anchors, and grouping. No plus operator, no question mark, no alternation, no repetition counts. The `egrep` program added support for plus, question mark, and alternation. Meanwhile, somebody went back and added repetition counts to `grep` but didn't add them to `vi`; somebody else added the `<` and `>` metacharacters to `vi` but didn't add them to `sed`. POSIX added repetition counts to `awk` but changed the notation from `{n,m}` to `{n,m}`. And so on. No two programs use the same regular expression language, but they overlap sufficiently that you can often get by with the common subset and not have to worry about which particular flavor you're up against.

Until you wander into the places where they differ.

**From:** John Jones

**Subject:** Problem with regular expression

I'm trying to write a regular expression to match blah blah blah.

**From:** Jane Smith

**Subject:** RE: Problem with regular expression

I think this will match what you want: `^Z@1&*B*!34`

I just ran my hand randomly over the keyboard to generate that fake regular expression. The scary thing is, at first glance, it is not obviously *not* a regular expression!

**From:** Chris Brown  
**Subject:** RE: Problem with regular expression

Try `$(#\$C)*#`

**From:** John Smith  
**Subject:** RE: Problem with regular expression

Thanks, everybody, for your suggestions, but I can't get any of them to work. For example, I can't get any of them to match against this string: blah blah blah blah.

At this point, people chimed in with other suggestions, confirming that John doubled the backslashes, that sort of thing. John posted his test program, and then the reason was obvious.

**From:** Jane Smith  
**Subject:** RE: Problem with regular expression

Oh, you're using `CAt1RegExp`. In that class, `\w` doesn't match a single character; it matches an entire word. You want to use `\a` instead.

Raymond Chen

**Follow**

