

Expressions and Regular Expressions

In this section:

- [Expressions](#)
- [Regular Expressions](#)
- [Additional Information](#)

Expressions

Expressions are used to match values; \$\$ is used with expressions to indicate a variable. You can enter expressions in the Modify Expression window. A few examples of valid expressions that you could enter in this window include:

Expression	Matches	Example
\$\$==n	a value equal to n	\$\$==1 matches values equal to 1
\$\$<n	a value less than n	\$\$<100 matches values less than 100
\$\$>n	a value greater than n	\$\$>100 matches values greater than 100
\$\$<=n	a value less than or equal to n	\$\$<=550 matches values less than or equal to 550
\$\$>=n	a value greater than or equal to n	\$\$>=1 matches values greater than or equal to 1

Regular Expressions

Regular expressions are used to match strings. Regular expressions are supported by many languages, including Perl, Python, and Ruby. RuleWizard's regular expressions are similar to those supported by Perl. You can enter regular expressions in the Regexp field of the Modify String window. The following guidelines should help you enter regular expressions:

Character /meta-character	Matches	Examples
anystring	an occurrence of the string "anystring"	"soft" matches parasoft, software, soften, etc.
.	exactly one non-null character	".at" matches hat, cat, bat, fat, etc., but not "at" w...ing matches webking, working, but not "what a king" or "wing"
?	0 or 1 occurrences of the preceding character	"j?test" matches either jtest or test
*	0 or more occurrences of preceding character	"a*soft" matches asoft, or aaaaaasoft; ".*ing" matches webking, waning, wing, what was that thing
+	1 or more occurrences of the preceding character	"a+soft" matches aaaaaasoft, or aaasoft, but not asoft
[]	matches one occurrence of any character inside the brackets; ^ inverts the brackets metacharacter	"[cpy]up" matches cup, pup, or yup "rule0[1-4]" matches rule01, rule02, rule 03, rule 04 "[^ch]at" matches all 3 letter words ending with "at" except for cat and hat.
[A-Z]	any uppercase letters from A to Z	"[A-Z]" matches any uppercase letter from A to Z
[a-z]	any lowercase letters from a -z	"[a-z]" matches any lowercase letter from a-z
[0-9]	any integer from 0 to 9	"rule[0-9]" matches any expression that begins with "rule" and ends with an integer
(?)	ignore case	"(?)ParaSoft" matches ParaSoft, PARASOFT, parasoft or paraSOFT
{ }	like *, but the string it matches must be of the length specified in the braces	"a{2}" matches aa, aaa, aaaa, etc. "a{3,}" matches at least 3 occurrences of the preceding character (aaaaa, or aaaaaaaa, but not aa "a {2,5}" matches between 2 and 5 occurrences of the preceding character (aaa, aaaaaa, but not aa or aaaaaaaaaaaaaaaaaa) "^a(2)\$" matches only aa
	matches the string before the " ", the string after the " ", or both	"rulewizard codewizard" matches rulewizard, codewizard, or both

Additional Information

- The caret symbol (^) indicates the beginning of a string in parentheses. The dollar sign (\$) indicates the end of a string in parentheses. Thus, to get an exact match for a string, use the format ^(STRING)\$. For example ^(soft)\$ would only flag "soft".
- If you want a violation reported if the expression is detected, disable the Regexp window's **Negate** option.
- If you want a violation reported when the expression is not detected, enable the Regexp window's **Negate** option.
- Regular expressions searches are case sensitive by default.
- When using regular expressions, the backslash (\) is an escape character that you can use to match a ".", "*", or another character that has a non-literal meaning.