

Using Regular Expressions in Policy Patrol

Policy Patrol 3.0 has been enhanced with a powerful string-matching tool to allow you to easily find and filter (spam) keywords. Spam often contains word derivatives and special characters to try to fool spam filters. For instance instead of using the word Viagra in a solicitation the offending email may use word variations such as V.I.A.G.R.A, VIAGRA!!, \$VIAGRA\$, VIAGRAV and V1AGRA to fool spam filtering software. With Policy Patrol you can now easily create one single regular expression to catch all these variations.

Regular expressions have been used in Unix systems and scripting languages like Perl and Python for many years. They can be simple or very complex. There are many resources including entire books written on using regular expressions (see more information). This document describes the basic Regular Expression syntax and how to use Regular Expressions in Policy Patrol.

Regular Expression Syntax

The following is a list of some of the most common regular expression syntax. You should be careful to not create overly general expressions. Although `.*hot.*` would match many spam phrase like 'hot offer' or 'hot deal on prescription drugs', it would also match 'hotel' and 'photography', which are not necessarily spam words.

Character	Function	Description/Example
.	<i>Any single character</i>	The dot (.) Matches any single character. <i>Examples:</i> 'me.' would match men, met, etc (but not me as it requires an additional character)
*	<i>Zero or more of the previous character</i>	The asterisk (*) matches zero or more instances of the previous character in order. <i>Examples:</i> 'mo*' would match zero or more of the character 'o', e.g. mo, moo, and mooo. 'fa.*' would match zero or more of the special character (.), i.e. any single character. So it would match words such as fa, fan, and fantastic.

Character	Function	Description/Example
+	<i>One or more of the previous character</i>	The plus sign (+) matches 1 or more of the previous character <i>Examples:</i> 'mo+' would match moo or mooo (but <i>not</i> mo as it requires 1)
?	<i>Zero or one of the previous character</i>	The question mark (?) matches 1 or 0 of the previous character <i>Examples:</i> 'mi?' would match mi or mii (but <i>not</i> miii as it matches only 0 or 1).
[]	<i>Any character from the set</i>	The square brackets ([]) match any character from a predefined group. <i>Examples:</i> '[aeiou]' would match any vowel.
[^]	<i>Any character not from the set</i>	The circumflex characters is a not operator, inside square brackets ([^]) it would match any character not from a predefined group. <i>Examples:</i> '[^aeiou]' would match any character not a vowel (any consonant).
	<i>Or</i>	The separator can for instance be used with the group characters (and). <i>Examples:</i> '(a @)' would match a and @. '(box filter tv)' would match box, filter and tv.
\s	<i>White space</i>	Backslash s matches any white spaces including a tab. <i>Examples:</i> 'reduce\sdebt' would match reduce debt, but not reduce\$debt, or reducedebt.
{ num }	The preceding element <i>num</i> times	A number within curly braces matches the preceding element that number of times. <i>Examples:</i> '[aeiou]{3}' would match any word with three vowels in a row.

Character	Function	Description/Example
{ <i>min, max</i> }	The preceding element between <i>min</i> and <i>max</i> times	Two numbers (the second must be >= the first), separated by a comma, within curly braces matches the preceding element between <i>min</i> and <i>max</i> times. <i>Examples:</i> '[aeiouy]{2,5}' would match any word with between two and five vowels in a row. '[aeiouy]{2,}' would match any word with more than two vowels in a row. '[aeiouy]{,3}' would match any word with fewer than 3 vowels in a row.
\b	The <i>start</i> of a word	To match whole words only, use the character sequence "\b". <i>Examples:</i> '\bhot' would match the word hot and hotel, but not photo. '\bhot\b' would match only hot. <i>Note:</i> Remember that the Policy Patrol filter options 'Whole word(s) are matched' and 'Whole or part of word(s) are matched' are not operative with regular expressions. Use the \b sequence at the beginning and end of your regular expression if you do not want to match word subsets.

Regular Expression examples

Below are a few regular expression examples.

Regular Expression	Matches
v.i.a.g.r.a	This would match v*i*a*g*r*a and v/i/a/g/r/a, but not viagra or vi@gra.
buy.*now	This would match buy now, buy.now, buy-now, buy//now and buynow.
d.?e.?b.?t	This would match debt, d*e*b*t and d-e-b-t but not d//e//b//t.
\\$\\$+	This would match \$\$\$\$\$, \$\$ and \$\$!! but not \$.

'...V..[A @]..G..R..[A @]...'	This would match a word with any first three letters, followed by a V, followed by any two characters, then an A or an @, etc, such as: V.I.A.G.R.A, VIAGRA!!, \$VIAGRA\$, ,V1AGRA, VI@GR@.
'...V..A..G..R...A...'	This would match V.I.A.G.R.A, VIAGRA!!, \$VIAGRA\$, VIAGRAV and V1AGRA, but not VI@GR@.
'\[^\aeiouy]{6,}'	This would match words with more than 6 consonants in a sequence.
'\[^\t\n\r\f]{35,}'	This would match words with more than 35 characters.
cable.?????(box filter tv).?	This would match: cabletv, cable*tv, cablebox, cable-box, freecable tv and cable*****box*.

Using Regular Expressions in Policy Patrol

Regular Expressions can be entered in Policy Patrol Word/phrase filters. To create a Word/phrase filter follow the next steps:

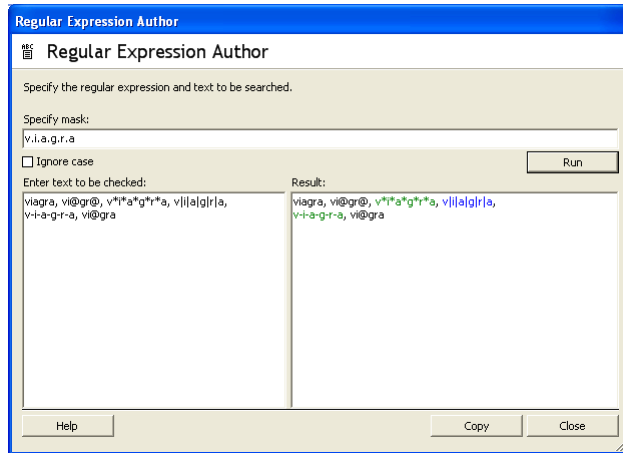
1. In the Policy Patrol Administration console go to **Filters**, select the appropriate folder and click **New....**
2. When asked which type of filter you wish to create, select **Word/Phrase Filter**. Click **Next**.
3. Enter the regular expression in the Word/phrase column and tick the box regular expression. Note that the options **Whole word(s) are matched** and **Whole or part of word(s) are matched** do not apply to regular expressions since this can be indicated in the regular expression itself (see '\b' in the Regular Expression Syntax). The case sensitivity and multiple count options are valid for regular expressions.

4. Enter a name for the filter and a description. When you are done, click **Finish** to create the filter.

Regular Expression Author

Policy Patrol includes a Regular Expression Author to help you create and test your regular expressions. Follow the next steps to use the Regular Expression Author:

1. In your Word/phrase filter, click on the **Regular Expression Author** icon in the toolbar.



2. In **Specify mask**, enter your regular expression, for instance `v.i.a.g.r.a`. If you wish to ignore case, select the option **Ignore case**.
3. In the left dialog, enter the sample text to be checked for the regular expression.
4. Click on **Run**. The words that match the regular expression will be colored green and blue alternately. For instance, in the example above, you can see that the regular expression `v.i.a.g.r.a` matches `v*i*a*g*r*a`, but not `viagra` or `vi@gra`.
5. If the result is not as you had intended, alter the regular expression and press **Run** again. If your regular expression produced the intended results, press **Copy** and **Close**. Now paste the regular expression into the word/phrase filter and tick the box **Regular expression**.

Sample Regular Expression filters

Policy Patrol includes several sample filters that include regular expressions, including the 'Spam words' and 'Offensive content' word/phrase filters, found in Filters > Sample filters > Policy Patrol Spam Filter.

Sales: sales@reearthsoftware.co.uk
Support: support@reearthsoftware.co.uk

Policy Patrol® is a registered trademark of Red Earth Software®. Copyright © 2001- 2004 by Red Earth Software.