

## How can I make a regular expression filter?

Author:  
Firetrust Ltd.

Created On: 05 Oct 2005 01:31 AM

Regular expression filters take a bit of getting used to, but here is a basic outline and some more common examples below that you may like to use yourself.

Character	What it represents	Example
-----------	--------------------	---------

<b>Any</b>	Represents the character typed, with the exception of special characters defined below	A represents A, B represents B etc
------------	--	------------------------------------

<b>.</b>	Any character(except line breaks)	Will match c, 3, [SPACE] etc
----------	-----------------------------------	------------------------------

<b>d</b>	Any digit	Will match 0, 1, 2, 3, 4, 5, 6, 7, 8 or 9
----------	-----------	---

<b>D</b>	Any non-digit	Will match anything except 0, 1, 2, 3, 4, 5, 6, 7, 8 or 9
----------	---------------	---

<b>^</b>	Beginning of line	
----------	-------------------	--

<b>\$</b>	End of line	
-----------	-------------	--

**t** [TAB] thello matches "hello" preceded by a tab

---

**r** Line Break [RETURN]

---

**s** Whitespace Matches any whitespace character: [SPACE], [TAB], Line Break, New Line

---

**S** Non whitespace Matches any non whitespace character

---

**w** Word characters Typically letters, numbers and underscores

---

**W** Non word characters Matches any non word character

---

**character** A character that is normally a special character Special characters are: . # ^ \$ ? + \* | [ ] ( )

---

**[any series of characters]** Any characters inside the brackets [abc] matches a, b or c

---

**[any character - another character]** Any characters within the range of characters [a-c] matches a, b or c

---

**[^ any series of characters]** Any character except ones after the ^ [^c3] matches any character except c or 3

---

---

? 0 or 1 of the previous character `ba?t` matches `bat` or `bt`, but not `boat`

---

---

\* 0 or more of the previous character `ba*t` matches `bt`, `bat`, `baat` etc

---

---

+ 1 or more of the previous character `ba+t` matches `bat`, `baat` etc but not `bt`

---

---

***pattern1/pattern2*** Either of the patterns specified `ba|t` matches `ba` or `t` but not `bat` (it will match `ba` and then `t` for two matches instead of one)

---

---

***(pattern)one of the special characters*** Same as above, but treats the characters in the parenthesis as a group `(ba)*t` match `t`, `bat`, `babat` but not `bt` The online help for the regular expression engine that MailWasher Pro uses can be found [here](#).

---