Filter Expressions

Contents

Filter Expressions

Filter data in reports, logs, and lists of objects using simple regular expressions.

Regular expressions used to filter information in PHEMI Central must be compatible with both the Java and the Python programming languages.

The simple regular expressions described in this section are supported by the system. More complicated regular expressions may succeed, but are not guaranteed to succeed.

Character Classes

Generally, the allowed character classes include lowercase letters, uppercase letters, numbers (decimal digits), and the underscore character ("_"). Taken together, these characters comprise the "word" character class.

To match a specific string of letters, numbers, punctuation, or a mix, simply enter the string. For example entering the string `2014` matches any entry containing the string "2014". A blank space is entered simply as a blank space.

To make more flexible expressions, use *metacharacters* and *escape sequences*.

Metacharacters

Metacharacters are characters that are not matched themselves, but indicate how a filter expression should be interpreted.

Character(s)	Description
[ ]	Square brackets. Matches any single character within the brackets. For example, `[abc]` matches any of the characters "a", "b", or "c". Backslash escapes are not allowed within square brackets. To filter on a sequence of more than one character inside brackets, delimit the expressions using a single quote; for example, ['11"12"13'] A closing square bracket may be included in the bracket expression if it is the first character of the set (or the first character after the carat), as in `[]abc]`.
-	Hyphen. When within square brackets, specifies a range of values. For example, `[a-z]` matches any lowercase letter from "a" through "z". The expression `[a-d]` is equivalent to `[abcd]`. The hyphen is treated as a literal character if it is the first or last character inside square brackets, as in `[abc-]` or if it is the first character after the carat inside square brackets.
[^]	Matches the negation or complement of the characters within the brackets. For example, `[^abc]` matches any character that is neither an "a" nor a "b" nor a "c", while `[^a-z]` matches any character that is not a lowercase alphabetic character. When outside square brackets, the carat is matched as an ordinary character.
^	Matches the starting position of the string or line.
$	Matches the ending position of a string or line, or the position immediately preceding the newline character.
\	The escape character. When followed by an ordinary character, the escape character signals a special sequence; for example, `\n` indicates a newline character. When followed by a metacharacter, the metacharacter loses its special meaning and is matched in the same way as an ordinary character.

Character(s)	Description
.	Dot, or period. Matches any single character, except newline (\n). For example, `.at` matches "bat", "cat", "hat", "mat", and so on. Within a bracket expression, the dot matches a literal dot. Therefore, while `.at` matches "bat", "cat", and so on, `[.at]` matches "." or "a" or "t".

Escape Sequences

There are certain characters that take on a new meaning when they are preceded by the escape character, the backslash.

Escape Sequence	Description
\n	Newline.
\r	Carriage return.
\t	Horizontal tab.
\x0B	Vertical tab.
\f	Form feed.
\d	A digit. Equivalent to [0-9].
\D	A non-digit. Equivalent to [^0-9].
\s	A whitespace character. Equivalent to [\t\n\r\x0B\f].
\S	A non-whitespace character. Equivalent to [^\s].
\w	A member of the word character class. Equivalent to [a-zA-Z_0-9].
\W	A non-word character. Equivalent to [^\w].