

Regular expressions are a way of defining patterns. The values of Index Fields will be evaluated against the defined regular expression. Where a valid match is found, it will be used to populate the Index Field.

A full discussion of regular expression definition is beyond the scope of this document – an online search will provide further details if required.

Some basics of regular expressions:

Character class	Description	Pattern	Matches
[<i>character_group</i>]	Matches any single character in <i>character_group</i> . By default, the match is case-sensitive.	[ae]	"a" in "gray" "a", "e" in "lane"
[^ <i>character_group</i>]	Negation: Matches any single character that is not in <i>character_group</i> . By default, characters in <i>character_group</i> are case-sensitive.	[^aei]	"r", "g", "n" in "reign"
[<i>first - last</i>]	Character range: Matches any single character in the range from <i>first</i> to <i>last</i> .	[A-Z]	"A", "B" in "AB123"
.	Wildcard: Matches any single character except \n. To match a literal period character (. or \u002E), you must precede it with the escape character (\).	a.e	"ave" in "nave" "ate" in "water"
\w	Matches any word character.	\w	"I", "D", "A", "1", "3" in "ID A1.3"
\W	Matches any non-word character.	\W	" ", "." in "ID A1.3"
\s	Matches any white-space character.	\w\s	"D " in "ID A1.3"
\d	Matches any decimal digit.	\d	"4" in "4 = IV"
\b	Matches a word boundary	\b[A-Z]{3}\b	Matches 'CAT' in 'A BLACK CAT RUNS' but not in 'SCATTERED'

Quantifier	Description	Pattern	Matches
*	Matches the previous element zero or more times.	\d*\.\d	".0", "19.9", "219.9"
+	Matches the previous element one or more times.	"be+"	"bee" in "been", "be" in "bent"
?	Matches the previous element zero or one time.	"rai?n"	"ran", "rain"
{ <i>n</i> }	Matches the previous element exactly <i>n</i> times.	",\d{3}"	",043" in "1,043.6", ",876", ",543", and ",210" in "9,876,543,210"
{ <i>n</i> , }	Matches the previous element at least <i>n</i> times.	"\d{2,}"	"166", "29", "1930"
{ <i>n</i> , <i>m</i> }	Matches the previous element at least <i>n</i> times, but no more than <i>m</i> times.	"\d{3,5}"	"166", "17668" "19302" in "193024"

Regular expressions will match when a field's value contains the pattern. So for example looking for 'BC123' will match in 'ABC1234'. To avoid this, it is possible to use \b at the beginning or end of a regular expression.

\b[A-Z]{1}[0-9]{2}

A12	TRUE
A123	TRUE
1234	FALSE

\b[A-Z]{1}[0-9]{2}\b

A12	TRUE
A123	FALSE
AB1234	FALSE