

UNITED STATES  
PATENT AND TRADEMARK OFFICE



# Federal trademark searching: Field tag searching with regular expressions

Jason Lott

Managing Attorney, Trademarks Customer Outreach

Images used in this presentation are for educational purposes only.



UNITED STATES  
PATENT AND TRADEMARK OFFICE ®

# Legal disclaimer

- This program is not legal advice.
  - It is for informational purposes only.
  - It does not include all aspects of searching.
  - It only covers general concepts and the mechanics of searching the federal register.

# Discussion topics

- Search system fundamentals
- Search system options
  - Field tags with regular expressions
- Searching takeaways

Discussion topic

# Search system fundamentals



# Search system fundamentals

- Two ways to use the system
  - Dropdown searches
  - Field tag searches
- Two ways to adjust your options
  - Basic mode
  - Expert mode

Discussion topic

# **Search system options: Field tags with regular expressions**



# Fields tags with regular expressions

- What are regular expressions?
  - Programming language
    - Greater flexibility in constructing searches
    - Greater control in constructing searches





# Fields tags with regular expressions

- Pattern matching searching
  - Efficiently execute searches that include phonetic equivalents and alternate spellings.
    - cat → **cat** or **kat** or **qat** or **ccat** or **ckat** or **cqat**



# Fields tags with regular expressions

- Tips for regular expression searching
  - Search terms go between two forward slashes.
  - Pattern match by placing letters inside brackets.
  - Request a single character by inserting a period.
  - Request infinite characters by inserting a period and asterisk (“dot star”).

**CM:/.\*trad.m[aeiouy]rk.\*/**

Discussion topic

# **Field tags with regular expressions: Searching syntax**



# Searching syntax

- Signal the system for advanced searches.
  - Surround search term with forward slashes.
  - Type search term in lowercase letters.
    - CM:/trademark/



# Searching syntax

- Signal the system to pattern match.
  - Surround letters with square brackets.
  - Use quantifiers to search for multiple instances of a character.

Common vowel searches	
Search a or e or i or o or u or y	[aeiouy]
Search 0-1 vowels	[aeiouy]{0,1}
Search 1-2 vowels	[aeiouy]{1,2}

# Searching syntax

Common phonetic equivalent combinations		
Sound	Example	Pattern match term
Hard K	Cat; kite; fox	[ckqx] or [ckqx]{1,2}
Soft S	Soup; scene; zen	[scz] or [scz]{1,2}
F/PH	Fun; phonics	[fph] or [fph]{1,2}
J/G	Genius, jeans	d{0,1}[gj]
TION	Action, fashion	[scth]{0,3}[aeiou]{1,2}n
Long I/E	Sigh, lye	[iey]{1,2}

Discussion topic

# **Field tags with regular expressions: Truncation and wildcards**

# Truncation and wildcards

- Wildcards
  - Different functionality when using regular expressions
    - Compare:
      - Field tag searching: ? (returns any one character)
      - Field tag searching with regular expressions: ? (returns preceding character repeated 0-1 times)



# Truncation and wildcards

- Wildcards
  - Different functionality when using regular expressions
    - Compare:
      - Field tag searching: \* (returns infinite characters)
      - Field tag searching with regular expressions: \* (returns preceding character repeated 0+ times)



# Truncation and wildcards

Wildcard	Explanation
.	Match any character (excluding space) exactly one instance
.{0,n}	Match any character (excluding space) 0 to n instances
.*	Match any character (excluding space) 0 or more instances

Discussion topic

# **Field tags with regular expressions: Basic operators and groupings**



# Basic operators and groupings

- Operators
  - Connect two or more search statements.
  - Must be in UPPERCASE letters.
    - AND
    - OR
    - AND NOT



# Basic operators and groupings

- Groupings

- In regular expressions, a space is **not** treated as an OR.
  - CM:/.\*hip hop.\* / will not find any marks.
  - CM:(/.\*hip.\* / .\*hop.\* /) will find HIP marks or HOP marks.
- Search statements inside of parentheses are performed first.

Discussion topic

# **Search system options: Regular expression hypothetical**



# Regular expression hypothetical

You want to register:

**GOOBER**

as a brand name for

**surfboards**



Discussion topic

# **Field tags with regular expressions: Additional examples and demos**



# Additional examples and demos

Wildcard	Explanation	Examples
*	Repeats the preceding character 0+ times	<i>/cat*/</i> matches <i>ca, cat, catt, cattt</i> , but not <i>cats</i>
?	Repeats the preceding character 0-1 times	<i>/fis?t/</i> matches <i>fit</i> and <i>fist</i>
+	Repeats the preceding character 1+ times	<i>/com+/</i> matches <i>com, comm, and commm</i>
( )	Forms a group treated as a single character	<i>/abc(def)?/</i> matches <i>abc</i> and <i>abcdef</i> , but not <i>abcd</i>

# Additional examples and demos

Operator	Explanation	Examples
" "	Matches an exact term	ON:"nike inc" returns all marks owned by Nike Inc.  Does not return owners with just <i>nike</i> in the name
[ TO ]	Designates a range	RD:[2020-01-01 TO 2020-12-31] returns all marks with a registration date in 2020
~	Returns words in a certain proximity	CM:"zebra star"~3 returns <i>zebra star</i> , <i>star zebra</i> , and <i>zebra on a star</i>  Does not return <i>zebra on a rocket to the stars</i>

# Additional examples and demos

Operator	Explanation	Examples
	<p>Acts as the OR operator</p> <p>Matches a character or string on the left side or the right side of the bar</p>	<p><code>/abc xyz/</code> matches <i>abc</i> or <i>xyz</i></p>
[^ ]	<p>Acts as the AND NOT operator by negating the characters or string in the brackets</p> <p>Matches any character that's not listed inside the brackets</p>	<p><code>/[^abc]/</code> matches characters except <i>a</i>, <i>b</i>, or <i>c</i></p> <p><code>/[^a-c]/</code> matches characters except <i>a</i>, <i>b</i>, or <i>c</i></p>

# Additional examples and demos

Operator	Explanation	Examples
\	<p>Allows you to search a reserved character such as:</p> <p>. ? + *   { } [ ] ( ) " \ / # @ &amp; &lt; &gt; ~ - : ;</p> <p>Matches any character to the right of the backslash</p>	<p>/disney\+/ matches <i>disney+</i></p>
[]	<p>Allows you to search a reserved character such as:</p> <p>. ? + *   { } [ ] ( ) " \ / # @ &amp; &lt; &gt; ~ - : ;</p> <p>Matches any character inside the brackets</p>	<p>/disney[+]/ matches <i>disney+</i></p>

Discussion topic

# Searching takeaways

# Trademark search takeaways

- Regular expression searching
  - Provides an efficient way to search for alternate spellings and phonetic equivalents.
  - Use forward slashes to trigger a regex search.
  - Use uppercase for field tags and operators.
  - Use lowercase for search terms.
  - Don't use quotation marks inside a regex search.
  - Don't use AND, OR, and AND NOT inside a regex search.

**Questions?**



DISCLAIMER: References to particular trademarks, service marks, certification marks, products, services, companies, or organizations are for illustrative and educational purposes only and do not constitute or imply endorsement by the U.S. government, the U.S. Department of Commerce, the U.S. Patent and Trademark Office, or any other federal agency.