PatSeer Search Quick Start Guide

Table of Contents

NOTE: This guide assumes you have prior experience of searching in online databases (free or commercial). This guide covers only the search aspects and further details of other features are present in the online help section.

Most Common Syntax Mistakes and their Correction

WRONG USAGE	CORRECT USAGE
Not using parentheses properly across operators E.g. TAC:(mobile OR cellphone AND network)	A query like A OR B AND C is ambiguous. With varying operators each query portion must be enclosed in brackets. E.g. TAC:((mobile OR cellphone) AND network) E.g. TAC:(mobile OR (cellphone AND network))
Using $ or ! for truncation	You should use ***** for truncation and **?** as a single character wildcard.
Using ? to represent 0 or 1 character	**?** in PatSeer stands for exactly one character wildcard and does not match 0 characters. Use ***1** instead to match 0 or 1 characters.
Using wildcard * within double quotes	Wildcard does not work within Double quotes. So **"hybrid vehicle*"** will give zero results. Since the default operator in PatSeer is an exact match, simply search for **(hybrid vehicle*)** to get the desired result set.
Using **w/2**, **w~2, 2d or 2w** for proximity	You should use **w2** for unordered proximity and **wd2** for left->right ordered proximity
Using **ADJ** or **NEAR** operators for proximity	You should use **w1** and **wd1** for unordered and left->right ordered proximity. Here '1'is the number of words to span across. For example **w2**, **w5**, **wd50** etc.
Using **SAME** operator for same paragraph search	You should use **wp** for same paragraph search. If you want to search within same sentence you can also use the **ws** operator. Optionally a number can also be given to restrict the number of words to span across. For example **wp30** will mean within the same paragraph and within 30 words.
Entering a portion of the IPC or CPC class without a wildcard. E.g. IC:A61K	Use the wildcard * at the end of the class if you are giving only a portion of the full class. If you are entering classes within double quotes make sure there are **no** spaces between the main and subclass. E.g. IC:A61K* UC:235* E.g. IC:A61K05* E.g. IC: A61K05/21
Using & in company name. E.g. ASN:(AT&T)	In case you want to use & then the term must be enclosed in Double quotes. So the correct query will be **ASN: "AT&T"** This is true for text field too. So if you are searching for space & aviation in text, then it should be enclosed in double quotes. & is not searchable outside double quotes.
Using semi-colon or comma to separate OR'ed terms E.g. TAC:mobile,cellphone,pda	Please use OR between the terms. E.g. TAC:(mobile or cellphone or pda)
Searching inventor name by first-name surname. E.g. INV:(John Smith)	Most inventor names can be surname first. Also some may have an initial. So you will miss out on "Smith John" or "Smith John R" or "John R Smith". The best way to search is using bidirectional proximity with scope for accommodating 1 more initial/middle name in between. E.g. INV:(John w1 smith)

The basics

- In PatSeer records are not grouped by families by default. So you can search individual records and choose to collapse them by one member per Family or de-duplicate results by Application Number (i.e., by Patents and Applications). The latter de-duplication is particularly useful for US records when you would like to keep only the grant and not its corresponding application in the result. When deduplicating results by family, you can deduplicate by both Simple Family (SFAM) and Extended Family (EFAM).

- When “None” de-dup option is selected, the results that you see are deduped by the publication number (without kind code). So even if a record has been published with 3-4 different kind codes during its life, it will show up as a single result. This is not applicable for Number search however. If you search for a number in Number Search it will show you all the matching kind codes as separate results. When “Show all Kind Codes” option is selected, the results can include multiple kind codes for the same publication number if they have the matched the search query.

- The Preferred record option allows you to decide which record should be chosen while deduplicating results. For example when deduplicating results by Application Number, if you select Latest Publication Date as the preferred record, you will get the Patents in place of applications in your results. When Deduplicating by Family you can also specify a country preference along with kind code. The kind code is optional however. E.g. you can give *US, EP, WO* or with kind code *USB*, USA, EPB*, WO, JPC*.

- Multiple terms and phrases can be combined together with Boolean operators to form a more complex query. If all words are required then **AND** should be inserted between them or if Any Word is required then **OR** should be inserted. To exclude a word use a **NOT** before it.

- The default operator assumed between multiple words is **Exact Match.** If you enter multiple words without any quotes then an exact match is assumed between them by default. (So if you want to combine words using **AND** operator, you will need to specifically mention it in your query. Further you should also specify **AND** between different fields when using Command line syntax.)

  So for example: *TAC:(carbon nanotube) IC:A61K**

  is incorrect and will not work. The correct query is: *TAC:(carbon nanotube)* ***AND*** *IC:A61K**

- Matching a phrase should be done by placing it in double quotes such as *“metal gate electrode”* or you can use proximity operator **wd0** say *(metal wd0 gate wd0 electrode)*. You can also use wildcard operators within a phrase search. For example "optic* fiber".

- You can use underscore to match all three types of combinations between words – single word, hyphenated word or exact match. So jet_engine will match jetengine, jet engine and jet-engine.

- PatSeer Search system has no stopwords. To search for words AND, OR, NOT or syntax related words such as **W**, **WD**, **WS**, **WP** you can include them in double-quotes.
  For example: *T:*"*system and method*"
  INV:"john w"
  Using these words outside double-quote will make the system treat them as operators and not search terms.

- Use parentheses to precisely indicate in case more than one connector is present and avoid any confusion in case of complex queries. In command line search, make sure each field's search terms are enclosed in a bracket. A common mistake is having a query like **A OR B AND C**. This is an ambiguous query and could lead in different results depending upon the sub-queries/field present in A, B, C. The correct way is one of the two: **(A OR B) AND C** ; **A OR (B AND C)**

- All three (left, right and middle) truncation is supported. Use '?' for single character wildcard and '*' for 0 to any number of characters. For performance reasons a right truncation can only be used after a minimum of 2 characters. Also a single query can have a maximum of two words/phrases with left truncation.

- Limited truncation is also supported and for this you need to specify a number after * . For example: *mix*2* will match *mixer* and *mixed* but not mixing. Similarly *colo*1r* will match *color* and *colour* both. Limited truncation is supported for right and middle truncations only.

- The date syntax to use when searching is YYYY-MM-DD. You can also specify only the year and month as YYYY-MM or just the year YYYY.

- Field Searching - You can search within any field by typing the field code followed by a colon ":" and then the term you are looking for. In command-line search if you do not specify any field then it by default searches the full text with Assignee and Inventor. Field codes are given at the end of this document.

  Proximity Search - Both left-to-right ordered and unordered proximity search is supported. The proximity operator can be applied using inline proximity characters **w** and **wd**. You can also search within the same sentence and search within the same paragraph using the operators **ws** and **wp** respectively. Examples shown below:

Proximity Type	Example using Inline proximity characters	Search Explanation
Left-Right Ordered Proximity	TAC:(mobile wd5 network)	Mobile within 0 to 5 words of network in the same order of occurrence
Bidirectional	TAC:(mobile w5 network)	Mobile within 0 to 5 words of network in the

unordered proximity		either order of occurrence
Range bound proximity	TAC:(mobile w2-5 network)	Mobile within 2 to 5 words of network in either order of occurrence
Fixed range proximity	TAC:(mobile w5-5 network)	Mobile and network should be separated by exactly 5 words and can be in either order of occurrence
Exact Match	TAC:(mobile network) TAC:(mobile wd0 network)	Search for exact match of phrase *mobile network*
Same Sentence proximity	TAC:(mobile ws network)	Search for mobile and network within the same sentence. The search will span across the length of the sentence.
	TAC:(mobile ws3 network)	Search within 3 words and within the same sentence. The span across 3 words is unordered.
Same paragraph proximity	TAC:(fiber wp optic*)	Search for fiber and optic* within the same paragraph. The search will span across the length of the paragraph.
	TAC:(fiber wp3 optic*)	Search within 3 words and within the same paragraph. The span across 3 words is unordered.

IMPORTANT: Search within sentence works for all Latin-text authorities. Search within paragraph however works correctly for US, EP, PCT, DE, CA and ES. It may or may not work correctly for countries with OCR'ed text such AU, FR, GB and some others

- Stemming support is integrated into the search form. Stemming cannot be applied to a whole search string and instead has to be supplied on a per-word basis. To stem a word simply add **#** at the end of the word and it will get replaced by its stem and wildcard after that. (So stemming is nothing but a right-truncated wildcard query internally)

- Classification Searching - To search for full class please make sure you enclose it in double quotes. Ex. G06F13/00, "234/45". To search for a portion of the class you should use a wildcard. Ex. IC:G06F* or UC:234*

- In PatSeer characters such as hyphen (**-**) , parentheses ({ [ ) }], apostrophe/quotes (' "), slash (/ or \ ) and comma (,) are indexed as space and consecutive spaces are reduced to a single space

- Searching in Non-English Content - PatSeer has separate fields for each language. The text fields (Title, Abstract, Claims and Description) for English Language are T, A, C, D, TA, TAC, TACD. The same fields for German text are TDE, ADE, CDE, DDE, TADE, TACDE, TACDDE where DE is added to the end of the field. Similarly the following codes can be added to search text in that language - Japanese - JA, French - FR, Korean - KR, Spanish - ES, Chinese (Simplified) - ZH,

Russian - RU, Swedish - SV, Portuguese –PT, Thai - TH and All Others - OH. In case of records that do not have a English Title or Abstract, PatSeer copies an English Title or Abstract (if available) within the simple family of the record i.e., from an equivalent record. The Equivalent record from where the title or abstract is added is shown within square brackets at the end of the text.

- Searching across all Languages - To search across all language simply use the fields T$,A$,C$,D$,TA$,TAC$,TACD$. The keywords given will be searched in text fields across all languages.

- Plural Matching – Simpler forms of plurals are automatically matched. (See section on Plural Matching for details) However, it is advisable to use wildcards yourself to have complete control on your search query.

Special symbols

?	:	matches a single character (You can use it multiple times to denote more number of characters)
*	:	matches zero or more characters
_	:	matches two words together or with a single hyphen /space
*#	:	# is any number. Will match zero to N characters.
term~	:	fuzzy search
wd#	:	left to right ordered proximity search
w#	:	unordered proximity search
ws#	:	unordered same sentence proximity search
wp#	:	unordered same paragraph proximity search
[_ TO _]	:	numeric range
[YYYY-MM-DD TO NOW]	:	Only for Date fields. To search from certain date to present
[BEG TO YYYY-MM-DD]	:	Only for date fields. To search up to a certain date
^	:	Term boosting

Search Examples

led light	: searches for an exact match of the phase “led light”
"led light"	: searches for an exact match of the phrase “led light”
led or "Light Emitting Diode"	: searches for led or “Light Emitting Diode”
led and "Light Emitting Diode"	: both led and “Light Emitting Diode” must be present
led not infrared	: only led must occur without occurrence of infrared
(led wd5 infrared)	: led must occur within 5 words of infrared and before infrared
(led w5 infrared)	: led or infrared must occur within 5 words of each other in any order
(led wd10 tv wd10 remote wd10 infrared)	: All 4 words: led, tv, remote, infrared must occur within 10 words of each other in the same order.
optic*	: matches optic, optics, optical
optic?	: matches optics and not optic or optical
car*t	: matches cart, carrot, carrrot, carkit
colo*1r	: matched color and colour
mix*2	: matches mixer and mixed but not mixing.
G08B1*	: matches G08B1, G08B15, G08B11 and not G08B0
C01G21/06$	: Also matches all the other classes dependent on C01G21/06 such as C01G21/08, C01G21/10, C01G21/12 etc.
brain~	: matches brain, brake, crain, drain (Fuzzy matching)
claim [4 TO 8]	: Matches claim 4, claim 5, claim 6, claim 7, claim 8
skate_board	: Matches skateboard, skate-board and skate board

Searching For Decimal, Percentage and Math Operators

Support for Decimals and Percentages

Searching for percentages and decimals is possible and has been made easy. PatSeer automatically takes care of spaces when searching for decimals or percentages. Further numbers attached to text (or embedded within text) are also automatically searched.

Here are the examples for searching decimals and percentages:

0.1 - matches 0.1, 0.1mm, K0.1N

.01 - matches .01, .01mm

[0.1 TO 0.3] - matches 0.1, 0.111, 0.101,..., 0.2992, 0.3

0.? - matches 0.1,…, 0.9. Will also match 0.5mm or K0.1N

9e.04 - matches 9e.04, 9e .04

"802.11g" - matches 801.11g, 802.11 g

To search for percentage character, please enclose it in double quotes.

5"%" - matches both 5% and 5 %

"5%" - same as above

0.01"%" - matches both 0.01% and 0.01 %

?"%" - matches all single digit percentages

??"%" - matches all double digit percentages

[0 TO 100]"%" - matches 0%, 1%, 2% ….. 99%, 100%

[0.4 TO 0.49]"%" - matches 0.4%, 0.40%, 0.405%, 0.41%, 0.41155%,..., 0.489%, 0.49%

[2.1 TO 2.45]"%" - matches 2.1%, 2.101%, 2.11%,…, 2.44%, 2.45%

IMP NOTE: In Range Query, you must use either integers or decimals on both sides of the TO operator. So *[2.5 TO 3]* may lead to a ambiguity and may not match 3.0

Additional Searchable mathematical operators

In addition to the already supported Greek characters, we have now the following mathematical operators also searchable. Please enclose them in double quotes while searching. Here is the list of the operators whose support has been added:

∑ Sum

∫ Integral

∂ Partial derivative

√ Square root

≠	Not equal to
∞	Infinity
⁄	Division slash
Δ	Delta
∇	Inverted Delta (Nabla/Del)
≥	Greater than or equal to
≤	Less than or equal to
≈	Approximately equal to

Plural Matching & Searching

Simpler forms of plurals are automatically applied to the search terms. This is applicable only for the text search fields such as T,A,C,D and not any other field. This would make words like 'vehicles' also match if only 'vehicle' is used in the search query. It applies to words which have an s-only and ies-only (for words ending in a 'y') plural form.

The following are examples of plurals that will and will not match when you search for the word on the left:

WILL MATCH ITS PLURAL

- paper -> paper or papers
- grid -> grid or grids
- automobiles -> automobile or automobiles
- strategy -> strategy or strategies
- baby -> baby or babies
- galley -> galley or galleys (As 'y' is not preceded by consonant it matches the s-type plural form)

WILL NOT MATCH ITS PLURAL

- process -> process (processes is an es-form of plural and so doesn't get matched with process. wildcard should be used for this if both words are needed)
- gas -> gas (same as above 'gases' will not get matched)
- potato -> potato (same as above 'potatoes' won't get matched)
- child -> child (special types won't be matched and so 'children' won't be matched)
- fungus -> fungus (Latin type plurals will not get matched and so 'fungi' won't be matched)
- phenomenon -> phenomenon ('phenomena' won't be matched)

- life -> life ('lives' won't be matched)
- knife -> knife ('knives' won't be matched)

SEARCHING FOR PLURAL WORD ONLY

To search for the plural form of the word only please enclose the word in double quotes. So "vehicles" will not match vehicle.

SEARCHING FOR SINGULAR WORD ONLY

To search for the singular form of the word only, you need to apply a NOT as plural forms are automatically searched. To search for *vehicle* only and not *vehicles* in title/abstract the query will be:

TA:(vehicle NOT "vehicles")

Date Range Searching

In Field Search, there is a *From - To* date Range option for Application and Publication dates. You can enter only a year or year with month and date.

- In the "From" field *2011* converts to *2011-01-01* and *2011-09* converts to *2011-09-01*. So the first day in the year or the first day in the month is automatically assumed.
- In the "To" field *2011* converts to *2011-12-31* and *2011-09* converts to *2011-09-30*. So the last day in the year or the month is automatically assumed

To search for an exact date, enter the same values in both From and To fields.

In Command line search you must specify the exact dates yourself to avoid any ambiguity. If you only enter year or year-month then the following is taken:

APD:[2009 TO 2012] : matches application dates 1st Jan 2009 to **1st Jan** 2012

APD:[2009-06 TO 2009-10] : matches applications dates 1st June 2009 to **1st Oct**2009

Using BEG and NOW in date range

You can use BEG and NOW in a range query as shown in the example below:

APD:[2009-06-01 TO NOW] - matches all dates from June 1, 2009 to current date.

APD:[BEG TO 2014-12-31] – matches all dates up to Dec 31, 2014 (including Dec 31, 2014)

Using the Year fields

Each date field also has a corresponding year field. So Application Date APD has APY, Publication Date PBD has PBY. So you can use these fields if you want to search for a year range. It's simpler and avoids any confusion. So:

APY:[2005 TO 2012] : matches all application dates in years 2005 to 2012 (both years included)

Classification Tree/Sub-tree search

Classification tree search allows you to automatically search for the children (dependent classes) of any full IPC, CPC, US or FI classes.

The search syntax character for this is **$** ("dollar") and you can add that at the end of a "full class". In case of IPC the full class would be the Main Group with /00 (E.g. C01L1/00) or Subgroup (E.g C01L1/195).

For example:

C10G 21/00	**Refining of hydrocarbon oils, in the absence of hydrogen, by extraction with selective solvents** (C
C10G 21/02	• with two or more solvents, which are introduced or withdrawn separately **[2006.01]**
C10G 21/04	• • by introducing simultaneously at least two immiscible solvents counter-current to each other **[2006.01]**
C10G 21/06	• characterised by the solvent used **[2006.01]**
C10G 21/08	• • Inorganic compounds only **[2006.01]**
C10G 21/10	• • • Sulfur dioxide **[2006.01]**
C10G 21/12	• • Organic compounds only **[2006.01]**
C10G 21/14	• • • Hydrocarbons **[2006.01]**
C10G 21/16	• • • Oxygen-containing compounds **[2006.01]**
C10G 21/18	• • • Halogen-containing compounds **[2006.01]**
C10G 21/20	• • • Nitrogen-containing compounds **[2006.01]**
C10G 21/22	• • • Compounds containing sulfur, selenium, or tellurium **[2006.01]**
C10G 21/24	• • • Phosphorus-containing compounds **[2006.01]**
C10G 21/26	• • • Silicon-containing compounds **[2006.01]**
C10G 21/27	• • • Organic compounds not provided for in a single one of groups C10G 21/14-C10G 21/26 **[2006.01]**
C10G 21/28	• Recovery of used solvent **[2006.01]**
C10G 21/30	• Controlling or regulating **[2006.01]**

IC:C01G21/06 only searches for this class.

IC: C01G21/06$ will also search for C01G21/08, C01G21/10, C01G21/12, C01G21/14, C01G21/16, C01G21/18, C01G21/20, C01G21/22, C01G21/24, C01G21/26, C01G21/27 in addition to the C01G21/06.

Similarly if you search for **C01G21/00$** it will also search all the 16 classes under this.

UC:71/31 only searches for this class.

UC:71/31$ will also search for 32 classes that are dependent on this.

UC:71/64.01$ also searches for 12 classes that are dependent on this.

Important Note: The $ is only supported at the end of a full class and should not be used as a replacement for wildcard. **You cannot use C01L$ or C01$ or C01L1**$. You will have to continue to use the wildcard character * for it.

The classification tree search is supported for CPC, IPC, US and FI classes.

Proximity Search

To do a left-right ordered proximity search use **wd** operator between the words. For unordered proximity search use **w** operator between the words. For example:

(antivirus wd5 software) : searches for antivirus within 5 words of software and antivirus must appear before software

(antivirus wd0 software) : searches for antivirus within 0 word of software i.e., antivirus should be right before software. The operator isn't needed as a search for (antivirus software) will also achieve the same result.

(antivirus wd2 software wd2trojan): searches for antivirus, software and Trojan within 2 words of each other.

IMPORTANT:

- **Only for users using the older double-quote syntax for proximity:** The disadvantage of using double-quotes for proximity is that you cannot use wild-card within the double-quote as it will be ignored. So if you are searching for optic* then the query TAC:(fiber optic*) will work and TAC: "fiber optic*" will only search for fiber optic.
- To do an unordered proximity search use **w** or double-quotes and % operator. So*(optics w1 fiber)*will match both *fiber optics* and *optics fiber*
- You can also enter a number range defining limits to number of words to search around. So *(optic w3-10 fiber)* will match only those records in which optic and fiber come within 3 to 10 words of each other.

You can order search within the same sentence and same paragraph using the ws and wp inline proximity operators. See example usage below:

Same Sentence proximity	TAC:(mobile ws network)	Search for mobile and network within the same sentence. The search will span across the length of the sentence.
	TAC:(mobile ws3 network)	Search within 3 words and within the same sentence. The span across 3 words is unordered.
Same paragraph proximity	TAC:(fiber wp optic*)	Search for fiber and optic* within the same paragraph. The search will span across the length of the paragraph.
	TAC:(fiber wp3 optic*)	Search within 3 words and within the same paragraph. The span across 3 words is unordered.

Common mistake: Keep in mind that (mobile **ws** network) is **NOT** the same as (mobile **ws0** network). (mobile ws network) will span across the length of the sentence and (mobile ws0 network) will look for

an exact match within the same sentence. In that sense (mobile ws0 network) and (mobile w0 network) is the same since in PatSeer an exact match query anyhow doesn't span across sentence boundaries.

However (mobile **w** network) is the same as (mobile **w0** network).

Using Boolean with Proximity

Using parentheses within proximity search you can increase the combinations of proximity search to be conducted. For example:

Using Inline Proximity	Explanation
(optical wd1 (fiber or fibre))	Searches for optical within 1 word of fiber or fibre
(mobile* OR cellular* OR handheld) wd5 (display OR LCD OR screen*)	searches for mobile* within 5 words of display/LCD/screen* or cellular* within 5 words of display/LCD/screen* or handheld within 5 words of display/LCD/screen*

You can use multi word phrases inside a Boolean Proximity construct. The inline proximity operators allow you to nest a proximity within an external proximity. For example:

- (turbo w5 (jet w2 engine*))
- T:((semiconductor laser excit*solid) w10 ((laser w1 amplifier) or (laser w1 device)))

You can also use the parentheses syntax but the multi-word phrases must be enclosed within brackets. See example of equivalent syntax below:

Using Inline Proximity	Explanation
((mobile* phone) OR cellphone OR handheld OR (cellular phone) OR (portable wd0 communicat* device)) w5 (display or LCD or screen*)	Since we want *mobile phone*, *cellular phone, portable communication device* **phrases** to be matched with *(display or lcd or screen)* we have included that in a bracket.

In the example in table above, since there is no operator between mobile* and phone, an exact match is assumed by default. This is equivalent to entering (mobile* wd0 phone).

Hit Cutoff Query

Say you wanted to search for only those records in which a word or a phrase appears more than X number of time in the field you have specified. The hit cutoff query is exactly for this type of need. The

query uses a **|#** syntax that should be added at the end of a word or a phrase that you want to specify a cutoff for. Here # is the count for cutoff. For example:

Using Hit Cutoff Query	Explanation
TAC:(mobile AND network|3)	Search for mobile and network but where network appears 3 or more times within TAC
D: "mobile phone"|5	Search for records that have the phrase mobile phone appearing 5 or more time in the description
TACD: "thermal spray"|5 AND coat*|5	The phrase “thermal spray” should appear 5 or more times and coat/coats/coating/coated should appear 5 or more times. (the sum of all variations is counted here)

NOTE: Hit Count Cutoff query cannot be used along with a proximity or exact match syntax. So TAC:(optic*|2 disk) will fail. TAC:(optic*|2 AND disk) will however work.

Minimum Must Match Query

This is query is used when you want to match any X number of words from a list of words. For instance this type of query can be used if you have 8 words out of which you need atleast 3 words in any record to be relevant for inspection.

Use ***any#*** to enforce a minimum number of words to match within an OR’ed list where # is the number of minimum words that should be present.

TAC:any2(compute or terminal or pc or laptop)* : This will match any 2 words from the list of words in OR query.

It’s important that all words within the bracket are combined with an OR only. AND/NOT will not work inside the list of words. However you can combine the whole query any#() with AND or NOT.

For example: *TAC:((any2(compute* or terminal or pc or laptop) AND network*) NOT protocol)*

Wildcards and operators are not to be used inside ANY#, NLPL, NLPM, NLPH query

Stemming

To stem a word simply add # at the end of the word and the word will get automatically replaced with its stem. This technique can be used in all search forms except for command line search. The purpose of such real-time stemming integration is to make the search transparent so that you are sure what is being searched and what isn’t.

encoding# will get replaced by encod*

formulation# will get replaced by formul*

Language specific stemming rules are applied and stemming is support when searching in 6 languages - English, German, French, Spanish, Russian and Swedish. The stemming operator is the same and the system detects the language based on the field name the search is applied to. So if you are searching in TA (English Title and Abstract) then the system will apply the English Stemmer and if you are searching in TADE (German Title and Abstract) then the system will apply the German Stemmer.

Term Boosting

You can boost a term or a phrase to influence the relevance order of results.

GPS^4 OR triangulation : matches either words but GPS documents come earlier in results.

"skate board"^10 : the phrase is boosted by order of 10

By default, the boost factor is 1. Although the boost factor must be positive, it can be less than 1 (e.g. 0.2).

Search Scripting

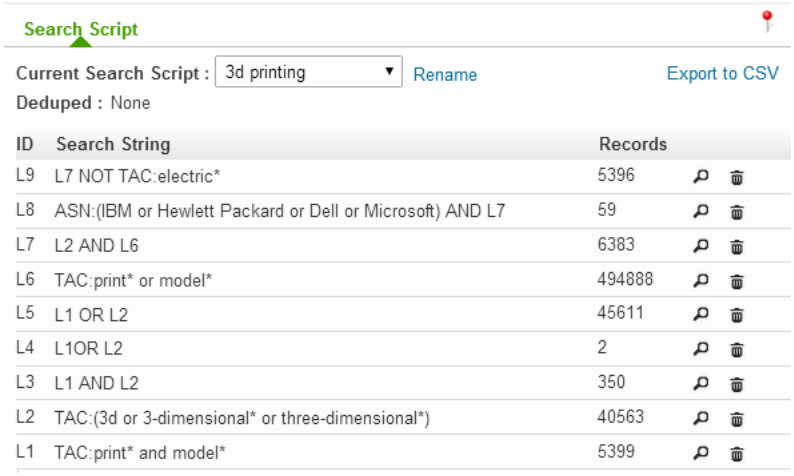
Search scripting helps you combine search queries with identifiers assigned to each search query. So you can prepare your final search strategy in steps rather than having to formulate the large search in a single go. Each search added to a script is given an identifier and search identifiers can be combined using AND, OR and NOT in Command Line Search.

The Search Scripts that you create can be seen in the “Script” tab that is present on the right in any of the search form page.

The overall process of search scripting works in the following manner:

1) Enter your Search on either Quick/Simple/Field or Command line Search form
2) Select the appropriate deduplicate option for your search. (Deduplicate once set for a search script remains the same for all search strings added to the script)
3) Click on “Add to Script” link next to the Search button
4) You will see the search is added to a search script that is shown in a separate window on the right. You can pin the “Script” window so that it remains on top always.
5) The First search is assigned an identifier “L1”
6) Further searches that you conduct will be given successive identifiers L2, L3 and so on
7) In command line search form you can now easily use these identifiers to build further queries. For example you can enter queries like (L1 AND L2) NOT L3.

8) Once your final query is run, click on the search icon next to the query in the script in order to jump to the results.



A search script that you start preparing is by default stored under the name “Untitled”. Even if you log out and come back later you will be able to see it under “Untitled”. This is similar to how opening new word document gives it a default name. You can Rename and save the script into an appropriate name.

All saved search scripts can be seen under the “Saved Searches” tab.

IMPORTANT NOTES:

- You can remove an entry from a search script. Such an action will also remove any other entries that dependent on it. For instance if you remove L2 and let’s say that L6 = L2 AND L4, then L6 is also going to be deleted.
- A new Search identifier will always be one more than the highest search identifier. Deleted search identifiers are not reused in the same script.
- For the counts to be accurate, the Dedup option is fixed with the first search query that is added to the script. All further search queries use the same dedup option. You cannot change the dedup for an existing script and you will have to create a new script in that case.

Hybrid Searching – Family Deduplication Modes

In PatSeer, you can de-duplicate records to one member per family (INPADOC or Simple) or deduplicate patents and applications. The latter can be useful if you want to have the patent selected in cases where both the patent and its application are present in the result set.

When deduplicating by family, you can give a country code preference and optionally select either Prefer or Restrict options. So when giving a country code preference you have a total of three modes of dedup. These are explained in the table below:

Deduplication - Country Preference Modes	Dedup Code	Search Action	Displayed Record
Country preference only	SFAM(C), EFAM(C)	The search runs across all families and country preference is applied only over those members that have matched the search query	From one of the matching hits in the family
Country preference with Prefer option checked	SFAM(P), EFAM(P)	The search runs across all families and country preference is applied on all members of any family in which one or more records have matched the search query	From the family having one or more matching hits in order of the country preference (Not necessarily the matching hit)
Country preference with Restrict option checked	SFAM(R), EFAM(R)	The search runs across only those families that have one or more members from the given set of countries only	From the matching families in order of the country preference

Field Codes and Examples

TEXT FIELDS		
T	Title	T:((led OR "light emitting diode") AND display)
A	Abstract	A:((led OR "light emitting diode") AND display)
C	Claims	C:((led OR "light emitting diode") AND display)
INC	Independent Claims	INC:((led OR "light emitting diode") AND display)
D	Description	D:((led OR "light emitting diode") AND display)
SOI	Summary of Invention (Within Description)	SOI:((led OR "light emitting diode") AND display)
ADV	Advantage of Invention (Within Description)	ADV:(improve* w5 (speed or velocity))
DRW	Description of Drawing (Within Description)	DRW:((arm w1 rest) AND chair)
DPA	Prior/Background Art Section within Description	DPA:((led OR "light emitting diode") AND display)
DEX	Examples Section at end of Description	DEX:((led OR "light emitting diode") AND display)
TA	Title and Abstract	TA:((led OR "light emitting diode") AND display)
TAC	Title, Abstract and Claims	TAC:((led OR "light emitting diode") AND display)
TAI	Title, Abstract and Independent Claims	TAI:((led OR "light emitting diode") AND display)
TACD	(Full Text) Title, Abstract, Claims and Description	TACD:((led OR "light emitting diode") AND display)
TAS	Title, Abstract, Summary of Invention	TAS:((led OR "light emitting diode") AND display)
TAW	Title, Abstract, Drawings section	TAW:((led OR "light emitting diode") AND display)
TAIS	Title, Abstract, Indep. Claims, Summary of Invention	TAIS:((led OR "light emitting diode") AND display)
TACS	Title, Abstract, Claims, Summary of Invention	TACS:((led OR "light emitting diode") AND display)
TAIW	Title, Abstract, Indep. Claims, Drawings section	TAIW:((led OR "light emitting diode") AND display)
TACW	Title, Abstract, Claims, Drawings section	TACW:((led OR "light emitting diode") AND display)
TAIA	Title, Abstract, Indep. Claims, Advantages	TAIA:((led OR "light emitting diode") AND display)
TACA	Title, Abstract, Claims, Advantages	TACA:((led OR "light emitting diode") AND display)
TAIAS	Title, Abstract, Indep. Claims, Advantages, Summary	TAIAS:((led OR "light emitting diode") AND display)
TACAS	Title, Abstract, Claims, Advantages, Summary	TACAS:((led OR "light emitting diode") AND display)
TAX	Title, Abstract, Examples Section	TAX:((led OR "light emitting diode") AND display)
TASX	Title, Abstract, Summary, Examples Section	TASX:((led OR "light emitting diode") AND display)
TAIX	Title, Abstract, Indep. Claims, Examples Section	TAIX:((led OR "light emitting diode") AND display)
TACX	Title, Abstract, Claims, Examples Section	TACX:((led OR "light emitting diode") AND display)
TAISX	Title, Abstract, Indep. Claims, Summary, Examples	TAISX:((led OR "light emitting diode") AND display)
TACSX	Title, Abstract, Claims, Summary, Examples	TACSX:((led OR "light emitting diode") AND display)
FCL	First Claim	FCL:(rechargeable lithium cell and phthalocyanine*)
FCW	First Claims Word Count	FCW:[10 TO 100]
NINC	Number of Independent Claims	NINC:3 AND TAC: sucrose

DATES AND YEARS		
APD	Application Date	APD:2007-06-21; APD:2009-12; APD:[2001-01-01 TO 2009-12-31]
APY	Application Year	APY:2001; APY:[2001 TO 2004]
EED	Estimated Expiry Date	EED:2012-01-23; EED:2012-12; EED:[2012-01-01 TO 2015-12-31]
EEY	Estimated Expiry Year	EEY:2015; EEY:[2013 TO 2014]
EPBD	Earliest Publication Date (for Extended Family)	EPBD:2001-01-23; EPBD:2001-01; EPBD:[2010-01-01 TO 2010-12-31]
EPBY	Earliest Publication Year (for Extended Family)	EPBY:2001; EPBY:[2001 TO 2002]
EAPY	Earliest Application Year (for Extended Family)	EAPY:2001; EABY:[2001 TO 2002]
EPRD	Earliest Priority Date (for Extended Family)	EPRD:2004-11-21; EPRD:2004-11; EPRD:[2001-01-01 TO 2005-12-31]
EPRY	Earliest Priority Year	EPRY:2004; EPRY:[1992 TO 2000]
PBD	Publication Date	PBD:2011-11-20; PBD:2011-11; PBD:[2003-01-01 TO 2007-12-31]
PBY	Publication Year	PBY:2011; PBY:[2010 TO 2011]
PRD	Priority Date (at Record level)	PRD:2001-03-23; PRD:2001-03; PRD:[2001-01-01 TO NOW]
PRD	Priority Date (at Record level)	PRD:2001-03-23; PRD:2001-03; PRD:[2001-01-01 TO NOW]
PRY	Priority Year (at Record level)	PRY:2011; PRY:[2010 TO 2011]
SPBD	Earliest Publication Date (for Simple Family)	SPBD:2015-08-17; SPBD:2015-08; SPBD:[2015-01-01 TO 2015-12-31]; SPBD:[2015-01-01 TO NOW]
SPBY	Earliest Publication Year (for Simple Family)	SPBY:2001; SPBY:[2001 TO 2002]
SABY	Earliest Application Year (for Simple Family)	SABY:2001; SABY:[2001 TO 2002]
SPRD	Earliest Priority Date (for Simple Family)	SPRD:2015-08-17; SPRD:2011-08; SPRD:[2015-01-01 TO 2015-12-31]; SPRD:[2015-01-01 TO NOW]
SPRY	Earliest Priority Year (for Simple Family)	SPRY:2004; SPRY:[1992 TO 2000]
CLASSIFICATIONS		
AC	All Classes (IPC,CPC,US,FI,FTERM,EC)	AC:(B23B29* OR “713/201”)
CPC	Cooperative Patent Classification (CPC)	CPC:B23B29/24$; CPC:A01N47*
CPCG	CPC Main Group	CPCG:B23B29*
CPSC	CPC Sub Class (First 4 letters)	CPSC:C07C
FI	Japanese File Index Classification	FI:C08L*; FI:"C08L25/04"
FTERM	Japanese FTERM	FTERM:4J001*; FTERM:"3C058/AA09"
FOC	Field of Search	FOS:"714015"
IC	International Classification (All versions)	IC:G06F13*; IC:G06F13/00
ICGR	International Patent Classification Group	ICGR:B66D3
ICO	International Classification (Version 1 to 7)	AC:(B23B29* OR “713/201”)
ICR	International Classification Revised (Version 8 - 9)	ICR:G06F13*; ICR:"G06F13/00"
ICSC	International Patent Classification (Sub Class)	ICSC:A61K
JFF	Japanese F-I Facets	JFF:LDR

LOC	Locarno Classification	LOC: 1216
UC	US Classification	UC:"713/201"; UC:713*
NCPC	Number of Cooperative Patent Classification	NCPC:[10 TO 15]
NCPSC	Number of Cooperative Patent Classification (Sub Class)	NCPSC:6
NFTERM	Number of Japanese FTERM	NFTERM:50
NICR	Number of International Patent Classification Full	NICR:3
NUC	Number of US Full Classification	NUC:3 AND PBY:2014
NICSC	Number of International Patent Classification Main	NICSC:3
PARTIES (ASSIGNEES, INVENTORS, ATTORNEYS)		
AASN	All Assignee Fields (ASNN, CASN, RASN, ASNO, ASNNL)	AASN:"General Motors"
ASN	Assignee	ASN:"General Motors"
ASNN	Normalized Assignee	ASNN:LG ELECTRONICS CO LTD
ASNNL	Assignee Non Latin	Search within all Non Latin Assignee Names
ASNO	Assignee Original	ASNO:LG ELECTRONICS INC
ASNC	Assignee Country	ASNC:(US OR CA)
ASNA	Assignee Address	ASNA:NY or “New York”
ASNST	Assignee State	ASNST:(CA or California)
ASNCT	Assignee City	ASNCT:Boston
ASNPIN	Assignee City Pincode	ASNPIN:90210
ATN	Attorney, Agent or Firm	ATN:"Richardson"
ATNA	Attorney Address	ATNA:NY or “New York”
ATNPIN	Attorney Pincode	ATNPIN:90210
CAAN	Current Assignee + Original and Normalised Assignees	CAAN:IBM
CASN	Current Assignee	CASN:IBM
PASN	Parent Corporate Owner	PASN:Alphabet
EXMR	Examiner	EXMR:"John"
INV	Inventor	INV:”MICHAEL CLARK”
INVC	Inventor Country	INVC:AU
INVA	Inventor Address	INVA:NY or “New York”
INVST	Inventor State	INVST:WY or Wyoming
INVCT	Inventor City	INVCT:Boston
INVPIN	Inventor City Pincode	INVPIN:90210
RAAN	US Reassignment Assignee + Original and Normalised Assignees	RAAN:IBM
RASN	All Assignees in US Reassignment History	RASN:("International Business Machines" or IBM)
ASNT	Assignee Type	ASNT:Govt
NASNN	Number of normalized Assignee	NASNN:[3 TO 5]
NCASN	Number of Current Assignee	NCASN:[3 TO 5]

NINV	Number of Inventors	NINV:2; NINV:[2 TO 10]
CITATIONS		
ALLCT	All Citations (BCT or FCT)	ALLCT:WO2014150626A1
BCT	Backward Citations	BCT:US6000000
FCT	Forward Citations	FCT:US6000000
FCTPA	Forward Citations by Applicant	FCTPA:US6000000
FCTPX	Forward Citations by Examiner	FCTPX:US6000000
FCTCX	where # can be X,Y,A,D,I,L,O,P,T,E Search in Forward Citations of a particular category #	FCTCX:US6000000
FCTREJ	Forward Citations Rejected by Examiner	FCTREJ:US6000000
REF	References (Non Patent backward citations)	REF:"Physics Today"
NBCT	Number of Backward Citations	NBCT:2; NBCT:[5 TO 100]
NREF	Number of Non-patent References	NREF:[3 TO 100]
COUNTRIES, NUMBERS AND FAMILIES		
APN	Application No	APN:EP20070824897
APNO	Application Number	APNO:553/MUMNP/2009
DS	Designated States	DS:(BE OR AT OR CH)
SFAM	Simple Family	SFAM: EP0261739; SFAM: EP0261739B1
FAMC	SFAM All countries	FAMC:(DE OR FR)
EFAM	Extended Family	EFAM: EP0261739; EFAM: EP0261739B1
EFID	PatSeer Extended Family ID	EFID:4423585
FAMID	PatSeer Simple Family ID	FAMID:27177089
KC	Kind Code	KC:(B1 OR B2)
PN	Patent No	PN: EP0261739
PNC	Patent No with Kind Code	PNC: EP0261739; PNC: EP0261739B1
PNKC	Patent No with Kind Code	PNKC:(EP2469552B1 OR US8261433B1)
PRN	Priority No	PRN:DE200610054043
PBC	Publication Country	PBC:CA
PRC	Priority Country	PRC:(WO OR EP)
PTYP	Record Type (Patent, Application, Utility Model)	PTYP:Patent; PTYP: Application
RPNC	Related Publications of the Input Numbers	US7375059 OR US2010251777 OR CN100361579C
LEGAL STATUS AND OTHERS		
LSC	Current Legal Status	TAC:(graphene) AND LSC:(Active-Granted / Applied)
LSE	Legal Status Events	LSE:"DE 2004" ; LSE:((US OR EP)) wd5 ([20040101 TO 20101231])
REGST	Register Status (US Only)	REGST:"Patented Case"
LDSA	Active in Designated States	LDSA:DE
LSSE	Legal Status - Other Significant Events	LSSE:opposition
LST	Legal Status Text	LST:hybrid vehicle
LLSE	Latest Legal Status Event (INPADOC)	LLSE:"DE 2004" ; LLSE:((US OR EP)) wd5 ([20040101 TO 20101231])

LSED	Latest Legal Status Event Date (INPADOC)	LSED:[20170101 TO 20170801]
LSN	Numeric Legal Status 0-Unknown, 1- Dead, 2-Active Pending, 3-Active Granted	LSN:(2 OR 3)
REGD	Register Status Date (US Only)	REGD:[2017-06-01 TO NOW]
REG	Region	REG:Asia
DOM	Technology Domain	DOM: "Mechanical Engineering"
SDOM	Technology Sub-Domain	SDOM: "Mechanical Tools"
SIM	Similar Patents to a given patent or list of patents	SIM:(US5884323A OR US6000000)

SEARCHING IN OTHER LANGUAGES	
T$, A$, C$, D$, TA$, TAC$, TACD$	Search in All Languages
T, A, C, D, TA, TAC, TACD	Search in English
TDE, ADE, CDE, DDE, TADE, TACDE, TACDDE	Search in German
TFR, AFR, CFR, DFR, TAFR, TACFR, TACDFR	Search in French
TKR, AKR, CKR, DKR, TAKR, TACKR, TACDKR	Search in Korean
TJA, AJA, CJA, DJA, TAJA, TACJA, TACDJA	Search in Japanese
TES, AES, CES, DES, TAES, TACES, TACDES	Search in Spanish
TZH, AZH, CZH, DZH, TAZH, TACZH, TACDZH	Search in Chinese
TRU, ARU, CRU, DRU, TARU, TACRU, TACDRU	Search in Russian
TSV, ASV, CSV, DSV, TASV, TACSV, TACDSV	Search in Swedish
TPT, APT, CPT, DPT, TAPT, TACPT, TACDPT	Search in Portuguese
TTH, ATH, CTH, DTH, TATH, TACTH, TACDTH	Search in Thai
TOH, AOH, COH, DOH, TAOH, TACOH, TACDOH	Search in Other Languages

STRING SEARCH FIELDS

The following field codes are for complete string only searches where the input must be an exact match of the searched text. The Search is **case sensitive** and will also not match portions on names. Wildcards aren't allowed and only complete names will be matched. Search for text/names with multiple words should be enclosed in double-quotes

PTYP_S	Record Type (String Format)	PTYP_S:"Patent" PTYP_S:"Design Application"
ASN_S	Full Assignee name (in Uppercase)	ASN_S:"GOOGLE INC"
INV_S	Full Inventor name (in Uppercase)	INV_S:"JOHN SMITH"

SIMILAR RECORDS BY CLASSIFICATION / ASSIGNEE / INVENTOR SEARCH

The following field codes take a record no. (With or without kind code) and searches for similar records by the field code criterion given. Usually the field code is **ANY**XX or **ALL**XX where XX is the particular field being matched.

ANYIC	Any IC of input patent number	ANYIC: WO2014028879A1
ALLIC	All IC of input patent number	ALLIC: WO2014028879
ALLICGR	All IC Group of input patent number	ALLICGR: WO2014028879A1

ALLICSC	All IC Sub-Class of input patent number	ALLICSC: WO2014028879A1
ANYUC	Any UC of input patent number	ANYUC: US6667125B2
ALLUC	All UC of input patent number	ALLUC: US6667125B2
ALLUCMN	All UC Main Class of input patent number	ALLUCMN: US6667125B2
ANYCPC	Any CPC Class of input patent number	ANYCPC:EP2711422
ALLCPC	All CPC Class of input patent number	ALLCPC:EP2711422
ALLCPCG	All CPC Group of input patent number	ALLCPCG: WO2014028879
ALLCPSC	All CPC Sub-Class of input patent number	ALLCPSC: WO2014028879A1
ANYASN	Any Assignee of input patent number	ANYASN: CN203179714U
ALLASN	All Assignee of input patent number	ALLASN: CN203179714
ANYINV	Any Inventor of input patent number	ANYINV: EP2711422
ALLINV	All Inventor of input patent number	ALLINV: EP2711422A1

Query Modifiers

Query modifiers are special functions that take a search query as an input and modify the results of that query.

The list of Query Modifiers are:

EFAMOF()	This query modifier will give all the extended family members of the results of input query in the result set	*E.g. EFAMOF(C:((led OR "light emitting diode") AND display))*
SFAMOF()	This query modifier will give all the simple family members of the results of input query in the result set	*E.g. SFAMOF(C:((led OR "light emitting diode") AND display))*
BCTOF()	This query modified will give all the backward citations of the results of the input query. Only the first 1000 results of the input query will be considered.	*E.g. BCTOF(TAC:((led OR "light emitting diode") AND display))*
FCTOF()	This query modified will give all the forward citations of the results of the input query. Only the first 1000 results of the input query will be considered.	*E.g. FCTOF(TAC:((led OR "light emitting diode") AND display))*
NLPH()	This query modifier runs a Natural Language Search in High Sensitivity Mode over the inputted text. The field code specified in the query will be used as the portion to match the inputted text.	*E.g. NLPH(TAC:(This statement is only an example of the free text you can enter to match across the patents.))*
NLPM()	This query modifier runs a Natural Language Search in Medium Sensitivity Mode over the inputted text. The field code specified in the query will be used as the portion to match the inputted text.	*E.g. NLPM(TACD:(This statement is only an example of the free text you can enter to match across the patents.))*
NLPL()	This query modifier runs a Natural Language Search in Low Sensitivity Mode over the inputted text. The field code specified in the query will be used as the portion to match the inputted text.	*E.g. NLPL(TA:(This statement is only an example of the free text you can enter to match across the patents.))*

Chemical Name Searching

PatSeer has these chemical names indexed in a fashion that makes the search process simpler for you. You only need to keep a one key rule in mind to form your queries for chemical name searches:

- In PatSeer characters such as hyphen (-) , parentheses ({ [ ) }], apostrophe/quotes (‘ ”), slash (/ or \ ) and comma (,) are indexed as space and consecutive spaces are reduced to a single space

So all you have to do is put down the name or a portion of it within an exact match query and remove all above characters.

Let’s take a couple of examples of to understand this.

Chemical name to search	**Search syntax for exact match**	**Search syntax to match variations**
(2,3-trans)-tetrahydro-2-phenyl-5- hydroxy-3-hydroxymethylfuran	TAC:(2 3 trans tetrahydro 2 phenyl 5 hydroxy 3 hydroxymethylfuran)	TAC:(2 3 trans tetrahydro* 2 phenyl 5 hydroxy 3 hydroxymethyl*)
6(r)-[2-[8(s)(2,2-dimethylbutyryloxy)-2(s)	TAC: (6 r 2 8 s 2 2 dimethylbutyryloxy 2 s)	TAC: (6 r 2 8 s 2 2 dimethylbut* 2 s)
3,4,6-trideoxy-3-(dimethylamino-ß-d-xylo-hexopyranosyl	TAC: (3 4 6 trideoxy 3 dimethylamino ß d xylohexopyranosyl)	TAC: (3 4 6 trideoxy 3 dimethylamino* ß d xylohexopyr*)

In the search syntax we have assumed you are searching within Title, Abstract and Claims. In order to match all name variations from a portion simply use a wildcard to truncate it as shown above in the last column. For example **TAC:(6 R 2 8 S 2 2 dimethylbut* 2 S)** will also match 6(R)-(2-(8(S)(2,2-DIMETHYL**BUTRYLOXY**)-2(S) in addition to 6(R)-[2-[8(S)(2,2-dimethyl**butyryloxy**)-2(S).

Legal Status Searching

If you are looking to search Legal Status codes and descriptions then **LSE - Legal Status Events** is field to use. LSE allows you to search by PRS code and also restrict it by country. The following examples explain how to search within the LSE field:

LSE:DE	matches all legal events in DE. EP and PCT records that face a legal event in their national phase DE records are also matched.
LSE:(DE 2004)	matches all legal events in DE in the year 2004
LSE:(DE wd5 R123)	matches DE and PRS Code R123
LSE:((DE 2004) wd5 R123)	matches records that have PRS code R123 in DE in year 2004
LSE:Revoked	matches all records that have "Revoked" mentioned anywhere in their legal status IMPORTANT: PRS Code descriptions are not always present in legal status definitions. So it is always advisable to search by PRS Code.
LSE:(DE wd10 Revoked)	matches all records that have Revoked mentioned in their legal events for DE
LSE:((DE 2004) wd10 Revoked)	Same as above but only for events in year 2004
LSE:20150115	matches any LSE update on 15th Jan 2015
LSE:[20151001 TO 20151130]	matches any LSE update from 1st Oct 2015 till 30th Nov 2015
LSE:[201510 TO 201510] Alternatively, you can use LSE:[20151001 TO 20151031]	matches any LSE update in month of October
LSE:[201510 TO 201511]	matches any LSE update in month of October and November 2015
LSE: [2014 TO 2015]	matches any LSE update in 2014 or 2015 (No change if only searching by year)
LSE:[20151015 TO 20151016]	matches any LSE update on 15th or 16th of October 2015

NOTE: *In Field Search form, next to the LSE field you will also see an LSE Query Builder icon, which makes it easy for you to construct advanced LSE queries. It is recommended that you use the LSE Query builder as much as possible.*

In addition to LSE, PatSeer also includes **two calculated fields**- LSC and LSSE

LSC or Current Legal Status is the current status of the record as calculated by PatSeer when going through its details. Every time new legal status data is received the LSC gets updated for all applicable records. For each record LSC can **only one of the** following values:

Name	Valid Values	Description of Value
Current Legal Status - LSC	Inactive – Withdrawn / Surrendered	If a record has been voluntarily given up
	Active – Granted	A granted patent or an application that has been granted
	Active - Applied	A pending application
	Inactive - Opposition / Revoked	If a record has been revoked or has become inactive due to opposition
	Inactive - Rejected / Refused / Suspended	If an application has been rejected prior to grant.
	SPC Active - Granted / Applied	If an SPC is Active or has been applied for the record
	SPC Inactive - Expired / Rejected	If the SPC has expired of the SPC application was rejected
	Inactive – Expired	If a granted patent has expired
	Inactive – Non Payment	If a granted patent has expired due to non-payment of maintenance fee

LSSE or Legal Status Other Significant Events captures other events in the life of the record that do not necessarily affect the status of the record but may be important to a searcher. This field is also updated regularly for all records where legal status updates are received. For each record this field can have **one or more** of the following values:

Name	Valid Values	Description of Value
Legal Status Other Significant Events - LSSE	Opposition	If the record has faced an opposition.
	Scope Limited	If a record has been limited in scope for example if claims have been removed or revoked.
	Inactive in Designated State	Not Active in one or more Designated States - If a EP, PCT record has faced cancellation/expiry/ withdrawal/abandonment in one or more designated states.
	Assignment	If the record has been reassigned to another entity.
	Collateral	If the record has been given as security to a bank.
	Licensing	If the record has been licensed to another entity
	Agent Change	If the Agent for the record has changed

	Litigation	If the Record has faced a Litigation in court
	Reexamination	If the patent has faced a reexamination request after grant
	Term Extension	If the term of the patent has been extended
	Reinstatement	If the Record was reinstated after expiry of a payment window

In Field Search if you press a spacebar in the LSC or LSSE fields, then you will see a drop down with a list of the valid values for you to select from.

Syntax Change Revision History

Query Changes effective August 7, 2017

- **New Classification Tree Search:** Use the $ operator at the end of a full class to search for all the lower classes in the hierarchy of that class

Query Changes effective August 18, 2015

- **New Syntax operator - Underscore**: Use underscore to match all three types of combinations between words – single word, hyphenated word or exact match. So jet_engine will match jetengine, jet engine and jet-engine.
- **New Query Modifiers for Natural Language Searching**: NLPH(), NLPM() and NLPL() - They allow you to combine natural language searches with Boolean search.

Query Changes effective April 15, 2015

The key changes are:

- **New FCTOF() and BCTOF() query modifiers –** As the name suggests these query modifiers allows you to get all the forward citations or backward citations respectively for your input query. Note that due to performance reasons, only the first 1000 results from the input query will be considered.

Syntax Changes effective June 10, 2014

The key changes are:

- **Kind Code support in Deduplication preference**: Get greater control in selection of which family member to display in your result set. When Deduplicating records by simple/extended families you can now specify the kind code. So the dedup. preference can be given like: USB*, USA, EPB*, WO, JPC . It's not mandatory to give kind code however. You can still give a preference like: EP,WO,US
- **New Minimum Must Match Query** - Use any# to enforce a minimum number of words to match within an OR'ed list. E.g TAC:any2(compute* or terminal or pc or laptop) - This will match any 2 words from the list of words in OR query. Similarly you can use any5, any10 etc.
- **New EFAMOF() and SFAMOF() query modifiers** - Get all family members for resulting records of your query in one go ! These Query modifiers will take another query as in input and will return all the family members of the results of that query. E.g. EFAMOF(TAC:antivirus) – This

query will give all extended family members of the results of TAC:antivirus in the search results page. Any complex query can be passed as an argument to these query modifiers.

Syntax Changes effective May 12, 2014

The key changes are:

- Left Truncation support has been added and it now possible to have queries such as **capsule**
- Automatic matching of basic s-type and ies-type plurals. (So if you search for *vehicle* it will also match *vehicles*. Or if you search for *strategy* it will also match *strategies*.)
- List of values in Legal Status Current (LSC) field have been changed
- Field Codes Table with Examples updated with latest set of all field codes

Syntax Changes effective April 4, 2014

The key changes are:

- It is **no longer necessary** to enclose Full Classes in double-quotes for accurate classification searching. So the following are now valid examples of searching in classifications:

  CPC:B23B29/24 CPC:A01N47* CPC:B23B29/2*

  IC: G06F13/00 IC:G06F13/* IC:G06F*

  UC:713/201 UC:713/* UC:713* UC:713/201.*

  Same is true for searching in FI, FTERM AND FOS (Field of Search)

Syntax Changes effective August 10, 2013

The key changes are:

- Wildcards are no longer supported within double quotes. A double quote query has been upgraded to do an exact phrase match and so syntax terms such as AND,OR,NOT,W,WD,WS,WP can also be searched within double quotes
- You are advised to use the inline proximity operators (w,wd,ws,wp) for your proximity searches as it supports wildcards and is more powerful than using double quotes with ~ or %. As **using double-quotes for proximity** doesn't bring any inherent advantages over the inline proximity construct, its use has been deprecated and may be completely remove in the future.
  Note: This doesn't affect using double quotes for phrase matching and that will always work.
- Support for searching within Sentence (using the ws# operator) and Searching within same paragraph (using the wp# operator) has been added
- A new type of search - Hit Cutoff Query using the | operator after a term or phrase has been introduced

Syntax Changes effective May 23, 2013

PatSeer upgrades effective today (May 23, 2013) include changes and enhancements that you should be aware about. This short guide will help you quickly understand these.

The key motivation behind these changes and enhancements is to help you form more powerful search queries (such as nested phrase and proximity queries) and also to leverage searching styles that you may already be used to.

To sum up in this update, there is **one** change and **one** major enhancement to proximity syntax.

Details of the Change

We have changed the default operator between keywords to be an exact match instead of the AND operator. The examples below will help understand the effects of this change:

Sample Search Query	Older Interpretation	New Interpretation
T:(jet engine)	T:(jet AND engine)	T: "jet engine" Basically it is interpreted as an exact match. Note that this will also match occurrences of jet-engine too.
T:”jet engine”	T:”jet engine”	Remains the same. There is no change for phrases within an exact match.
T:turbo jet or jet engine or rocket engine	T:((turbo AND jet) or (jet AND engine) or (rocket AND engine))	T:”turbo jet” or “jet engine” or “rocket engine”
T:jet AND engine	T:(jet AND engine)	Remains the same.

So if you want to combine words using **AND** operator, you will now need to specifically mention it in your query. Further you should also specify **AND** between different fields when using Command line syntax.

So for example:

*TAC:(carbon nanotube) IC:A61K**

is incorrect and will not work now. The correct query is:

TAC:(carbon nanotube) ***AND*** *IC:A61K**

Details of the Enhancement

We have introduced **inline proximity operators** to the search syntax in addition to the existing proximity syntax. These new inline operators are **w** for bidirectional and **wd** for left-right ordered proximity.

Valid usage is **w** (same as w0 meaning within 0 words), **w5** (within zero to five words), **w2-4** (within two to four words), **w4-4** (within exactly 4 words). Similarly **wd, wd5, wd2-4, wd4-4** can be used for left-to-right ordered proximity.

Note: There is no change with the existing proximity syntax and both the syntax can be used. So all your older queries will still work and you can continue to use the older syntax too.

So if you want to search for optic* within 2 words of fiber then you can give it as:

TAC:(optic w2 fiber)*	This will be look for optic* and fiber appears within two words and appearing in any order
TAC:(optic wd2 fiber)*	This will be look for optic* and fiber appears within two words and appearing in same order

A key benefit of the new syntax is that you can use phrases within proximity searches too by chaining the proximity operators. (Please make use of parentheses here to avoid ambiguous queries)

For example:

TAC:((optic w2 (fiber or fibre)) wd5 (transmission amplifier))*

This query will search for optic* within 2 words of fiber or fibre and occurring within 5 words of the phrase transmission amplifier. Also note that since there is no operator between transmission and amplifier and exact match is assumed as discussed above.

So the combination of the changes and enhancements allow you to use multiple phrases within a proximity construct.

TAC:((thermal or flame or (wire arc) or warm or cold or (high velocity oxy fuel) or HVOF or plasma or detonat) wd2 (spray* or coat*))*

In the query above *wire arc* and *high velocity oxy fuel* is treated as exact match. The rest are searched within 2 words of spraying or coating. So the query will match phrases like:

- Plasma spraying/coating

- Detonation spraying/coating
- Wire arc spraying/coating
- Flame spraying/coating
- High velocity oxy-fuel spraying/coating (HVOF)
- Warm spraying/coating
- Cold spraying/coating

Finally, note that you can even use a combination of the old and new syntax. For example:

"(((mobile wd1 phone) OR cellphone OR (cell wd1 phone) OR handheld OR (hand held) OR (cellular wd1 phone) OR (portable communicat* device))* ***AND*** *(display or LCD or screen*))"%5*

In the above example multiple proximity operators have been used within a double quote proximity construct. The above query could also have been written as:

((mobile wd1 phone) OR cellphone OR (cell wd1 phone) OR handheld OR (hand held) OR (cellular wd1 phone) OR (portable communicat* device))* ***w5*** *(display or LCD or screen*)*