Query Syntax Instructions

Overview

This document contains instructions on how to construct valid queries using AcclaimIP syntax. For a complete list of query field codes, please view the **AcclaimIP Cheat Sheet**.

You can create your own queries through the **Expert Search** form, the **Quick Search** window and the **My Query** form field.

This document explains the syntax you use to interact with the Query Parser in AcclaimIP.

Terms

A query is broken up into *terms*, *operators and fieldcodes*. There are two types of terms:

- Single Terms
- Phrases (strings)

A **Single Term** is a single word such as "*test*" or "*hello*".

A **Phrase** (often called a string) is a group of words surrounded by double quotes, such as "*Hello Dolly.*" Multiple terms can be combined together with Boolean operators to form more complex queries.

Fields

AcclaimIP supports fielded data. When performing a search, you can specify a field, or use the **Default Fieldset** (title, abstract, claims, specification, assignee, inventor, document number and others). A **Cheat Sheet** showing all the fields you can query is found as a help file located at **Start>Help>Cheat Sheet.**

You can search over 50 different data fields by typing the field name followed by a colon ":" and the term you are searching.

For example, to search the patent title field (TTL) for the string "coffee maker" and the abstract field (ABST) for the string "heating element," use the following syntax:

TTL:"coffee maker" AND ABST:"heating element"

or another example:

TTL:"switching transmission modes" AND 802.11n

Since the text, "802.11n" is the **Default Fieldset**, a field indicator is not required.

You can type this example into the **Quick Seach** window, the **Expert Search** form, or the **My Query** form field in your **Search Result Window's Refine Panel**.

Note: The field is only valid for the term that it directly precedes. Therefore, the query:

TTL:switching transmission modes

Will only find "switching" in the title field. It will also find "transmission" and "modes" in the **Default Fieldset**.

Using Parentheses and Double-Quotes

TTL:(switching transmission modes) finds documents with *all three terms* in the patent Title in any order.

TTL:"switching transmission modes" finds patent documents with the three terms in the indicated order.

Wildcard Searches

AcclaimIP supports single and multiple character wildcard searches within single terms (not within phrase queries).

To perform a *single character* wildcard search use the "?" symbol.

To perform a *multiple character* wildcard search use the "*" symbol.

The single character wildcard finds terms that match the term with the character replaced. For example, to search for "text" or "test" in the Specfication you can use the search:

SPEC:te?t

Multiple character wildcard searches finds documents with terms containing zero or more characters in place of the "*" symbol. For example, to search for *test, tests, testing, testable or tester* in the **Default Field-set**, use the search:

*test**

You can also use a wildcard search in the middle of a term.

*te*t*

Note: You cannot use a * or ? symbol as the first character of a search.

Fuzzy Searches

AcclaimIP supports fuzzy searches based on the Levenshtein Distance algorithm. To do a fuzzy search, use the tilde, "~", symbol at the end of a *single word term*. For example, to search for a term similar in spelling to "*roam*" in the Abstract, use the fuzzy search:

ABST:roam~

This search will find terms like *foam* and *roams*.

An optional parameter specifies the required similarity. The value must be between 0 and 1. Values closer to 1 match terms with a higher similarity to the term used in the query. For example:

ABST:roam~0.8

The default value, if a parameter is not specified in a fuzzy search, is 0.5.

Proximity Searches

AcclaimIP supports searching for words that are within a specific distance away from each other. To do a proximity search, use the tilde, "~", symbol at the end of a phrase. For example, to search for "remote" and "control" within 10 words of each other in the Claims of a document, use the search:

ACLM:"remote control"~10

NEAR Operator

NEAR is a type of Boolean operator that finds terms near each other. Unlike basic proximity searching, you can use wildcards, truncated terms and nested Booleans with the NEAR operator.

NEAR is properly typed in all upper case letters followed by a number which represents the padding. Padding is the maximum number of terms that can be found between the two terms. By default, the padding is set to zero (0) if no explicit padding number is given. Some examples:

light NEAR1 diode --> Finds "light emitting diode" for example, with the term "emitting" representing the 1 term of padding between the two expressed terms.

light NEAR diode --> Requires that both terms be next to each other to return a patent. Therefore it would return "light diode" and "diode light".

NEAR can be used in complex nested queries. For example:

mouth NEAR5 (protector OR guard OR shield)

Near can be used with wildcards in complex nested Boolean queries. For example:

("mobile device" OR "cell phone") NEAR10 (signal OR emit*)

ADJ Operator

The Adjacency or ADJ operator works similarly to the NEAR operator but the terms must be in the order you specify. For example:

Light ADJ5 diode --> Finds patents with 5 or fewer terms between "light" and "diode" but requires that "light" is to the left of "diode."

Examples of NEAR/ADJ queries.

Query Syntax

foo NEAR3 bar

foo* NEAR bar

(foo OR bar) NEAR10 something

(foo AND bar) NEAR something

(foo bar) NEAR something #note, same as above (default OP is AND)

"foo bar" NEAR something

(foo ADJ3 bar) ADJ (x NEAR y)

TTL:(foo NEAR bar)

TTL:(foo NEAR bar OR something) #note, this == TTL:(foo NEAR bar) OR TTL:something

a OR b NEAR c OR d #note, this == a OR (b NEAR c) OR d

(a OR b) NEAR (c OR d) #note, possibly the intended interpretation of the above

“near” #note, searches on the literal word near

TTL:”near” #note, searches on the literal word near

Queries that give errors	**Error message**
NEAR	Unexpected NEAR by itself
foo NEAR	Unexpected NEAR without right side clause
near foo	Unexpected NEAR without left side clause
TTL:foo NEAR SPEC:bar	Fields on both sides of NEAR must match
TTL:foo NEAR (SPEC:bar OR ABST:bar)	Fields on both sides of NEAR must match
TTL:foo NEAR (SPEC:bar AND ABST:bar)	Fields on both sides of NEAR must match
foo NEAR (a OR b AND c)	Cannot mix AND/OR with NEAR
foo NEAR (a AND b OR c)	Cannot mix AND/OR with NEAR
foo NEAR (NOT b)	Cannot use NOT with NEAR
foo NEAR (a NOT b)	Cannot use NOT with NEAR

Range Searches

Range Queries allow you to match documents whose field values are between the lower and upper bound specified by the Range Query**.** Range Queries can be inclusive or exclusive of the upper and lower bounds. Sorting is done lexicographically. For example:

ISD:[20020101 TO 20030101]

Finds documents whose issue date (ISD) has values between 20020101 and 20030101, inclusive of the two end dates.

Special Terms Used in Range Queries

The term **NOW** can be used in place of *today's date*. If you use **NOW** as an end date in your saved queries, then the date will always move forward to the present date.

ISD:[01/01/2000 to NOW] — shows all patents issued from 01/01/2000 to today.

ISD:[NOW-5YEARS to NOW] — shows all patents issued in the last five years.

Note: "NOW-5YEARS" is the first parameter in this query and the "-" does not mean "to" in the range query; it means minus.

EXPIRATION:[TO NOW]* — shows all expired patents in a set. * means today's date and "NOW-1DAY" means yesterday. Remember too, range queries using square brackets are inclusive of the two ends of the range.

*EXPIRATION:[NOW TO *]* — displays all active patents where the predicted expiration date on the patent is from NOW (today), to anytime in the future, *.

Note that **Range Queries** are not reserved for date fields. You can also use range queries with non-date fields:

*ANA_INREF_CT:[10 to *]* — finds all documents with 10 or more forward citations.

Inclusive range queries are denoted by square brackets. *Exclusive* range queries are denoted by curly brackets.

ISD:{2000 to 2012} — finds patents whose issue date is from 2001 to 2011 (exclusive of the end dates because of the curly brackets).

Boosting a Term

AcclaimIP provides a **Relevance** ranking for matching documents based on the terms it found. To boost a term use the caret, "^", symbol with a boost factor (a number) following the term you search. The higher the boost factor, the more relevant the term will be in the search results.

Boosting allows you to control the relevance of a term by boosting its importance in the document. For example, if you are searching for *coffee filter,* and you want the term "coffee" to be more relevant, boost it using the ^ symbol along with the boost factor next to the term. You could type:

coffee^4 filter — boosts documents with the term "coffee" and they appear higher on the search result list. You can also boost **Phrase Terms** as in the example:

"coffee filter"^4 "coffee maker"

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

Boolean Operators

Boolean operators allow terms to be combined through logical operators. AcclaimIP supports **AND**, "**+**", **OR**, **NOT** and "**-**" as Boolean operators (Note: Boolean operators must be ALL CAPS).

The Boolean AND Operator

The **AND** operator is the default conjunction operator. This means that if there is no Boolean operator between two terms, the **AND** operator is used. The **AND** operator links two terms and finds a matching document if both of the terms exist in a document. This is equivalent to an *intersection* using sets.

To search for documents that contain both "coffee filter" AND "heater" use the query:

"coffee filter" AND heater

The symbol && (double ampersand) can be used in place of the word **AND**.

The Boolean OR Operator

The **OR** operator matches documents where either term exists anywhere in the text of a document. This is equivalent to an *union* using sets.

To search for documents that contain "coffee filter" OR "coffee maker" use the query:

"coffee filter" OR "coffee maker"

A good search technique is to use the Boolean OR operator between synonyms (with or without boosting) when searching for a concept. For example:

(video OR monitor OR display OR screen)

The + Operator

The "+" or *Required Operator* requires that the term after the "+" symbol exists somewhere in a field of a document.

To search for documents that must contain "coffee" and may contain "filter" use the query:

+coffee filter

Note: If you sort by Relevance, the query above will favor (rank higher) documents containing both terms.

The NOT Operator

The **NOT** operator excludes documents that contain the term after NOT. This is equivalent to a *difference* using sets. The symbol ! can be used in place of the word NOT.

To search for documents that contain "coffee maker" but not "coffee filter" use the query:

"coffee maker" NOT "coffee filter"

Note: The **NOT** operator cannot be used with just one term. For example, the following search will return no results:

NOT "coffee maker"

The Hyphen (-)

The "-" or prohibit operator excludes documents that contain the term after the "-" symbol.

To search for documents that contain "coffee maker" but not "coffee filter" you can use the query:

"coffee maker" -"coffee filter"

Grouping

AcclaimIP supports using parentheses to group clauses to form sub-queries. It is useful if you want to control the Boolean logic for a query.

To search for either "coffee" or "tea" and "filter" use the query:

(coffee OR tea) AND filter

The parentheses eliminate any confusion and makes sure that "filter" must exist, and *either* one of the terms "coffee" or "tea," must also exist.

Note: You should be aware that the optional terms "coffee" and "tea" will affect the Relevance sort, so documents with a high density of all three terms will show up at the top of the list, and documents with a single occurrence of "filter" and no occurences of "coffee" or "tea" will appear at the bottom of the list.

Field Grouping

AcclaimIP supports using parentheses to group multiple clauses in a single field.

To search for a title that contains both the word "coffee" and the phrase "flavoring method" use the query:

TTL:(coffee +"flavoring method") — this is the same as typing:

(TTL:coffee AND TTL:"flavoring method") — but is a little shorter.

Searching Empty or Not Empty Fields

Empty or null fields can be searched by using the **FIELD** search parameter:

FIELD:isnotemptyICN AND NOT ICN:US — finds all patents with non-US inventors.

FIELD:isemptyANRE_CUR — finds all patents with no current assignee (unassigned patents).

Escaping Special Characters

AcclaimIP supports escaping special characters that are part of the query syntax. The current list of special characters are:

+ - && || ! ( ) { } [ ] ^ " ~ * ? : \

To escape these characters, use the \ before the character. For example to search for (1+1):2 use the query:

\(1\+1\)\:2