Espacenet – pocket guide

Smart search field identifiers and operators

<table>
<thead>
<tr>
<th>Field identifier</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>in</td>
<td>inventor</td>
<td>in=smith</td>
</tr>
<tr>
<td>pa</td>
<td>applicant</td>
<td>pa=siemens</td>
</tr>
<tr>
<td>ti</td>
<td>title</td>
<td>ti=&quot;mouse trap&quot;</td>
</tr>
<tr>
<td>ab</td>
<td>abstract</td>
<td>ab=&quot;mouse trap&quot;</td>
</tr>
<tr>
<td>pr</td>
<td>priority number</td>
<td>pr=ep20050104792</td>
</tr>
<tr>
<td>pn</td>
<td>publication number</td>
<td>pn=ep1000000</td>
</tr>
<tr>
<td>ap</td>
<td>application number</td>
<td>ap=jp19890234567</td>
</tr>
<tr>
<td>pd</td>
<td>publication date</td>
<td>pd=20080107 or pd=&quot;07/01/2008&quot; or pd=07/01/2008</td>
</tr>
<tr>
<td>ct</td>
<td>citation/cited document</td>
<td>ct=ep1000000</td>
</tr>
<tr>
<td>cpc</td>
<td>Cooperative Patent Classification</td>
<td>cpc=&quot;A61K31/13&quot;</td>
</tr>
<tr>
<td>cpcc</td>
<td>Classification combination</td>
<td>cpcc=&quot;C08F8/30&quot;, cpcc=&quot;C08F297/02&quot;</td>
</tr>
<tr>
<td>ftxt, desc, claims</td>
<td>Full text, description, claims</td>
<td>ftxt=microscope, desc=lens, claims=laser</td>
</tr>
<tr>
<td>Column</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>ia</td>
<td>inventor and applicant</td>
<td>ia=apple or ia=&quot;ries klaus&quot;</td>
</tr>
<tr>
<td>ta</td>
<td>title and abstract</td>
<td>ta=&quot;laser printer&quot;</td>
</tr>
<tr>
<td>txt</td>
<td>title, abstract, inventor and applicant</td>
<td>txt=microscope lens</td>
</tr>
<tr>
<td>num</td>
<td>application, publication and priority number</td>
<td>num=ep1000000</td>
</tr>
<tr>
<td>ipc</td>
<td>all current and previous versions of the IPC</td>
<td>ipc=A63B49/08</td>
</tr>
<tr>
<td>cl</td>
<td>IPC and CPC</td>
<td>cl=C10J3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operator</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Logical operators</strong></td>
<td></td>
</tr>
<tr>
<td>and</td>
<td>pa=bosch and pa=siemens</td>
</tr>
<tr>
<td></td>
<td>will retrieve documents where both Bosch and Siemens are applicants.</td>
</tr>
<tr>
<td></td>
<td>The default operator in <strong>Smart search</strong> is &quot;and&quot;. Left has precedence over right. No operator has precedence by default.</td>
</tr>
<tr>
<td>or</td>
<td>in=smith or in=huber</td>
</tr>
<tr>
<td></td>
<td>will retrieve documents where the inventor's name is Smith or Huber.</td>
</tr>
<tr>
<td>not</td>
<td>txt=laser not semiconductor</td>
</tr>
<tr>
<td></td>
<td>will retrieve documents containing the word <strong>laser</strong>, while excluding documents containing the word <strong>semiconductor</strong>.</td>
</tr>
<tr>
<td>all</td>
<td>ti all &quot;paint brush head&quot;</td>
</tr>
<tr>
<td></td>
<td>will find all terms entered within quotes within the field identifier, although not necessarily in the order in which they appear. This corresponds to ti=(paint and brush and head).</td>
</tr>
<tr>
<td>any</td>
<td>ti any &quot;motor engine&quot;</td>
</tr>
<tr>
<td></td>
<td>will retrieve any of the terms entered within quotes within the field identifier. This corresponds to ti=(motor or engine).</td>
</tr>
<tr>
<td><strong>Proximity</strong></td>
<td>prox/distance&lt;nr mouse prox/distance&lt;3 trap</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>will retrieve documents where the words <strong>mouse</strong> and <strong>trap</strong> are less than three words apart in the TXT identifier and in the order shown.</td>
</tr>
<tr>
<td>prox/unit=sentence</td>
<td>mouse prox/unit=sentence trap</td>
</tr>
<tr>
<td></td>
<td>will retrieve documents where the words <strong>mouse</strong> and <strong>trap</strong> happen to be in the same sentence in the TXT identifier.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Comparison</strong></th>
<th>= pa=siemens</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>will retrieve documents where the applicant's name is Siemens.</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or equal to (only valid for the field identifier pd) pd &gt;=1994</td>
</tr>
<tr>
<td></td>
<td>will retrieve documents having a publication date higher than or equal to 1994.</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal to (only valid for the field identifier pd) pd &lt;=2014</td>
</tr>
<tr>
<td></td>
<td>will retrieve documents having a publication date less than or equal to 2014.</td>
</tr>
<tr>
<td>within</td>
<td>Retrieve documents published within a date range pd within &quot;1994 2014&quot; pd within &quot;1994, 2014&quot;</td>
</tr>
<tr>
<td></td>
<td>will retrieve documents published between 1994 and 2014.</td>
</tr>
</tbody>
</table>

Note that the query **pd >=1994 and pd <=2014** will also retrieve documents published between 1994 and 2014.
Truncations

Truncation symbols (wildcards) available in **Smart search** and **Advanced search**:

<table>
<thead>
<tr>
<th>Wildcard</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>stands for a string of characters of any length</td>
<td>car* will retrieve car, cars, card, cart, care, carbon, etc.</td>
</tr>
<tr>
<td>?</td>
<td>stands for zero or one character</td>
<td>car? will retrieve car, card, cart, care, etc. but <strong>not</strong> cards, carbon</td>
</tr>
<tr>
<td>#</td>
<td>stands for exactly one character</td>
<td>car# will retrieve card, cart, care, cars, etc. but <strong>not</strong> car, cards</td>
</tr>
</tbody>
</table>

**Restrictions**

- Left truncation (?car) is **not** supported.
- If two alphanumeric characters precede ? or # (co? or pa#), then a maximum of three truncation symbols is allowed (ca??? will retrieve call, cart, card, care, cable, etc.).
- If three or more alphanumeric characters precede a ? or # symbol, then a maximum of seven truncation symbols is allowed.
- There must be at least three alphanumeric characters preceding a * symbol.
- Truncation is not allowed in application numbers.
- Wildcards should not be used in the **IPC** and **CPC** fields as the data is auto-posted, meaning that each symbol is indexed at different levels. Example: B (section level), B65 (class level), B65D (subclass level), B65D81 (group level).

**Searching with dates**

**Date formats**

You can search for a specific **publication date** in both the **Advanced search** mask and the **Smart search** mask using any of the following formats:
### Date ranges

The following formats are admissible for **publication date** range searches in the **Advanced search** mask.

<table>
<thead>
<tr>
<th>Format</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>yyyy</td>
<td>2014</td>
</tr>
<tr>
<td>yyyy-mm</td>
<td>201403</td>
</tr>
<tr>
<td>yyyy-mm</td>
<td>2014-03</td>
</tr>
<tr>
<td>mm/yyyy</td>
<td>03/2014</td>
</tr>
<tr>
<td>mm.yyyy</td>
<td>03.2014</td>
</tr>
<tr>
<td>yyyyymmdd</td>
<td>20140305</td>
</tr>
<tr>
<td>yyyy-mm-dd</td>
<td>2014-03-05</td>
</tr>
<tr>
<td>dd/mm/yyyy</td>
<td>05/03/2014</td>
</tr>
<tr>
<td>dd.mm.yyyy</td>
<td>05.03.2014</td>
</tr>
</tbody>
</table>

The following formats are admissible for **publication date** range searches in the **Smart search** mask.

<table>
<thead>
<tr>
<th>Format</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Date1&gt;:&lt;Date2&gt;</td>
<td>1994:2014</td>
</tr>
<tr>
<td>&lt;Date1&gt;,&lt;Date2&gt;</td>
<td>199401,201412</td>
</tr>
<tr>
<td>&quot;&lt;Date1&gt; &lt;Date2&gt;&quot;</td>
<td>&quot;19940101 20141231&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Format</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>pd=&quot;&lt;Date1&gt;:&lt;Date2&gt;&quot;</td>
<td>1994:2014</td>
</tr>
<tr>
<td>pd=&lt;Date1&gt;,&lt;Date2&gt;</td>
<td>199401,201412</td>
</tr>
<tr>
<td>pd=&quot;&lt;Date1&gt; &lt;Date2&gt;&quot;</td>
<td>&quot;19940101 20141231&quot;</td>
</tr>
</tbody>
</table>
Nested queries

Parentheses can be used to specify the order in which the search terms and operators should be interpreted.

Information within parentheses is read first, and then information outside parentheses is read next.

If there are nested parentheses, the search engine processes the innermost parenthetical expression first, then the next, and so on, until the entire query has been interpreted.

Examples:

- (mouse or rat) and trap
- ((mouse or rat) and trap) or mousetrap

Search limitations

- Maximum of 10 search terms per field.
- Maximum of 20 terms in total and 19 operators per mask.
- When combining search fields in the Advanced search mask, the default operator is "and". It cannot be changed.
- Default operators within a search field in the Advanced search mask are the following:

<table>
<thead>
<tr>
<th>Search field</th>
<th>Searched items</th>
<th>Default operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title, Abstract, Full text</td>
<td>Keywords</td>
<td>and</td>
</tr>
<tr>
<td>Publication number</td>
<td>Document numbers</td>
<td>or</td>
</tr>
<tr>
<td>Application number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Priority number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Publication date</td>
<td>Date</td>
<td>or</td>
</tr>
<tr>
<td>Applicant</td>
<td>Names</td>
<td>and</td>
</tr>
<tr>
<td>Inventor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPC</td>
<td>Classification codes</td>
<td>and</td>
</tr>
<tr>
<td>IPC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If other operators are required these should be entered manually.

- Truncation: see above
- Apostrophes, hyphens and diacritical characters are not recognised.
- Names are not searchable for documents published in other alphabets (Cyrillic, Greek, Japanese, etc.).