

Search Syntax

Knowing how to create refined, efficient searches is important for both users and administrators. Leveraging the Ultraseek search syntax to its full potential yields efficient, specific results. Users are helped along with this through rotating tips, the help interface, and the Advanced Search form.

This appendix describes the Ultraseek syntax and contains the following sections:

- [Search Strategies](#)
- [Plus and Minus Operators](#)
- [Vertical Bar and “Search these results”](#)
- [Double-Vertical bars](#)
- [Wildcard Search](#)
- [Field Searches](#)

Search Strategies

It is important to identify phrases and terms when constructing a query. In each case, Ultraseek is instructed to match words in consecutive order.

Spell Checking

Query, or spelling suggestions are not made for every query. Ultraseek insists on a reasonable level of confidence before making the suggestion. This confidence is based on the typographic difference between the query and the suggestion as well as the frequency of the suggested query terms relative to the frequency of the original query terms.

Suggestion, not correction: Ultraseek does not correct the spelling of queries. Instead Ultraseek proposes alternate queries based on re-spelling the terms of a query. For instance, in the following English language queries, each query term is correctly spelled:

```
silicone valley  
silicon implant
```

However, the following suggestions can be made:

```
silicone valley  
Did you mean: silicon valley?  
silicon implant  
Did you mean: silicone implant?
```

After the user has submitted a query, if Ultraseek has an alternate query to suggest, it will be presented under the query box, and above the search results. For instance, if the original query is:

```
britneey speers rock music
```

The following suggestion appears:

```
Did you mean: britney spears rock music
```

The phrase “britney spears” is hyperlinked back to the query page, with the adjusted spelling. Words with adjusted spellings from the original query are presented with the HTML `<I>` emphasis.

Spell Checking can be disabled by commenting out the following line of Python in `query.html` located in the Ultraseek `/docs` directory:

```
if enable_spell and st == 1: exec self.file("spell.html");
```

The commented version looks like this:

```
# if enable_spell and st == 1: exec self.file("spell.html");
```

After you turn off spell checking, you need to reload the query.html for the changes to take effect. To reload the file, click the **reload** button next to the Document directory field under the **Server | Parameters | Main** page.

Phrases and Terms

Consider the following query:

```
white house cat
```

This query will produce better results if phrases are identified. Surround phrases with quotation marks to identify them:

```
"white house" cat
```

```
white "house cat"
```

In the first example "white house" cat, Ultraseek will recognize the two phrases. In the first example, the search results will contain the phrase "white house" as a primary phrase and the results will be ranked higher. The search will also return "cat" as a secondary phrase, or word, and will rank the documents containing that word/phrase with lower relevancy.

Note If there is a document that contains both the phrase "white house" and the term "cat", then that document will be ranked by the phrase "white house". Therefore, it is possible to have a document ranked very high which contains the term "cat", even though this is the secondary search term.

In the second example, white "house cat", Ultraseek will again recognize both phrases, but with differing results. In the second result set, the color white will be ranked higher than "house cat" because it appears first in the search terms. The results set will have all documents with the term "white" first, "house cat" second, and white "house cat" third.

Commas and Separating Phrases

Ultraseek allows you to search phrases without quotation marks. By doing this, Ultraseek interprets the entire statement as one phrase. If you want to search multiple phrases and enter the example below, you will get search results which contain all of the terms because Ultraseek interprets “white house cat” as a single phrase.

```
white house cat
```

Note Long phrase searches may return documents that contain a portion of the phrase.

However, if you want to search primarily for “white house”, and secondly for “cat”, then you would separate the terms by using a comma as below:

```
white house, cat
```

Your search results would first contain documents with “white house” in them, and then documents with “cat”.

Alternatively, if you want to search primarily for the term white, and secondly for “house cat”, the search entry would look like this:

```
white, house cat
```

Your search results would first contain documents with the term white in them, and then the documents with house cat. If you had intended to search for documents containing “white house cat” as a single phrase, then you would enter the terms without a comma, and in parenthesis:

```
“white house cat”
```

Capitalization

If a word is submitted entirely in lower case, case is ignored in the resulting hits. However, if any capitalization is used, case is exactly matched.

For example, a query for next will return documents containing the words “next”, “Next”, and “NeXT”. However, a query for NeXT will only return documents containing the word with exactly matching case: “NeXT”.

Plus and Minus Operators

The plus and minus operators are useful tools that improve the control you have over search results by allowing you to exclude or include terms.

Plus

If a term must appear in each search result, put a plus in front of it:

```
product price +module
```

In the above example, the words `product` and `price` might be included in the search results. However, the word `module` *must* occur in each hit. The plus sign can be used with other search syntax tools:

```
+site:www.verity.com +"new and improved"
```

In the above example, the plus sign is used in conjunction with the site field search. The plus sign is also used with quotation marks to require a phrase.

Note Using the “+” excessively is not recommended. Ultraseek will scan each document for the entire term `product price module` when it determines relevance. Therefore, if you need to see the term `module` in each of the results, enter the terms without a comma, and Ultraseek will interpret them as a phrase and rank all documents with all three terms higher in relevancy than those without.

Minus

Put a minus sign in front of terms that are not desired in search results:

```
bats -baseball
```

Search results will prefer documents that contain information about *bats (flying rodents)*, over *bats (baseball bats)*. However, the minus operator does not necessarily exclude terms altogether. Documents that contain terms marked with a minus operator will be excluded from the search results.

Vertical Bar and “Search these results”

The vertical bar and “Search these results” are useful tools that allow users to create specific, refined queries. The vertical bar will tell Ultraseek to include content on both sides of the vertical bar, where the string to the right of the vertical bar is a subset of the string to the left of the vertical bar. The vertical bar behavior mimics “Search these results” behavior. For example, suppose a user were to type the following query:

```
restaurant bistro
```

Resulting hits will contain the words “restaurant” and/or “bistro”. The user then clicks “Search these results” and enters the subsequent query:

```
pizza spaghetti
```

Results are now narrowed to those hits which also contain the words “pizza” and/or “spaghetti”. This process is called *query refinement*.

This same behavior can be duplicated in a single query by using the vertical bar, as shown in the following example:

```
restaurant bistro | pizza spaghetti
```

Resulting hits will contain the words “restaurant” and/or “bistro”, and also contain the words “pizza” and/or “spaghetti”.

Note that the vertical bar can be used repeatedly in the same query. For example, the above query can be refined even further:

```
restaurant bistro | pizza spaghetti | San Francisco "bay area"
```

The Difference Between Vertical Bar and Plus

There is often confusion between the vertical bar operator and plus operator. In many cases, both operators produce the same effect. For example, this query

```
+restaurant +pizza
```

and this query

```
restaurant | pizza
```

will return the same results. In each case, both the word “restaurant” and the word “pizza” must appear in the search results. However, by adding several more query terms to the mix, the difference becomes more apparent.

Examples

```
+restaurant +bistro +pizza +spaghetti  
restaurant bistro | pizza spaghetti
```

In the first example, the search results must contain **all** of the terms listed. In the second example, the search results must contain **at least one term to the left and one term to the right** of the vertical bar.

Double-Vertical bars

In search syntax, the double-vertical bar has exactly the same function as the vertical bar, but with one difference: relevance ranking. In searches that contain a double-vertical bars, only the content to the right of the double-vertical bars is used in relevance ranking. Content to the left of the double-vertical bars has no impact on relevance scores.

For example, consider the following query:

```
restaurant bistro || pizza spaghetti
```

The resulting hits will include content from both sides of the double-vertical bars, but only the words “pizza” and “spaghetti” will impact relevance scores. Hits containing these words will rank higher in the search results.

There is little use for the double-vertical bars in user queries, however administrators may want to know this syntax in order to better understand the `qp` form variable. When an administrator sets a value for the `qp` form variable, that value is prefixed to the user’s query with a double-vertical bar. For example, suppose the administrator has set a `qp` value of “language modules” in a hidden form element:

```
<input type="hidden" name="qp" value="language modules">
```

Now suppose a user types in the following query:

```
parts accessories
```

Ultraseek interprets the combination of `qp` and the user query with the following syntax:

```
language modules || parts accessories
```

Resulting hits will contain the word “language” and/or “modules”, and will contain the word “parts” and/or “accessories”. Relevance scores will be ranked based on the words parts and accessories.

Wildcard Search

Wildcard search expands the results of a search query to terms that match a wildcard operator in a search term. Ultraseek expands the wildcard term to matching terms in the searched collections, and appends the term matches to the query string. For example, a search for the term “stan*” expands to include the following results: “stand”, “standalone”, and “stanford”.

Ultraseek supports wildcard expansion in individual terms, but not in phrases. For example, wildcards used within quotes (“Stanford undergrad*”) are ignored, and not expanded. Searching for Stanford Undergrad* will search for “Stanford” “Undergrad”, but not the phrase “Stanford undergrad”.

To increase the relevance of results, only terms that appear in three documents will appear as a wildcard expansion. In addition, wildcard searches are case sensitive: a search for “stan*” produces different results than a search for “Stan*”. Ultraseek expands each wildcard search to a maximum of 40 terms in the search results. This number may be less depending upon the terms and their frequency within the collections.

If the wildcard search expands to too many terms to be useful, you may have difficulty locating the term that you are searching for. In this case, try a more specific search term. For example, you may have better results searching for “install*” instead of “inst*”.

Supported Operators

The following wildcard operators are supported:

Character	Description
*	Matches any number of characters in the search term, including the term itself. For example, a search for undergrad* returns results for “undergrad”, “undergrads”, and “undergraduate”.
?	Matches a single character. For example, a search for “st?n” returns results for “stan” and “stun”. Note: If you use a question mark (?) in a query, such as in ‘What’s up Doc?’, Ultraseek will expand the results for the word “Doc”, which may not return the intended search results. In this situation, enclose the search phrase in quotes.

Wildcard Position

The wildcard character can be at any position in the query term (leading, middle, or trailing). For example, “*stan”, “stan*”, “*stan*”, “st?n”.

Note Using wildcard operators at the beginning of a search term requires a full search of the term database in the collections you are searching, which will reduce search performance.

Rare Term Exclusion

The rare term exclusion threshold is configurable. In `patches.py`, you add the following information:

```
import wildcard
wildcard.MIN_DOC_FREQ = 0 # Does not check for existence of term
wildcard.MIN_DOC_FREQ = 1 # Checks that the term exists in at least
one doc
wildcard.MIN_DOC_FREQ = 3 # default value
<etc.>
```

Note that “deprecated” terms can persist in the term database for a time after they have been deprecated. For example, say you had a document containing “xyzzzy”, but the document has been modified so it no longer contains “xyzzzy”. The term “xyzzzy” will persist in the terms database. If you set `MIN_DOC_FREQ` to 0, it will return xyzzzy as an expansion of `xy*`. If you set `MIN_DOC_FREQ` to 1, it will check to see if xyzzzy still (appears to) exist before expanding `xy*` to include xyzzzy.

Using Modifiers in Wildcard Searches

Ultraseek supports the use of exclude modifiers. For example, modifier `-exclud*` expands to `modifier -exclude -excluded -excluding`. However, required modifiers (+) are not supported.