MediaPortal Wiki uses MindTouch, which incorporates <u>Lucene.Net 1.9</u>, a powerful and flexible search engine.

All you need to do is enter your search terms in the search box, however, you can search using many advanced search techniques.

# Pages Searched

The search function respects user permissions; this means you will **not** see search results for pages/files that you can't view.

You may also use the drop down box below the search box to select if you wish to search:

- Main Pages (default)
- All Pages
- Sub pages of the current page
- Only your user pages (sub pages of your User Page)

# Search Terms

A search query is broken up into terms and operators. There are two types of terms: Single Terms and Phrases.

- A Single Term is a single word such as "test" or "hello".
- A Phrase is a group of words surrounded by double quotes such as "hello dolly".

Multiple terms can be combined together with Boolean operators to form a more complex query (see below).

# Search by Field

When performing a search you can either specify a field, or use the default field of 'content'. You can search any field by typing the field name followed by a colon ":" and then the term you are looking for.

MindTouch defines the following fields:

- **author**: name of last person to modify the page or file
- comments: contents of comments on a page (not available for files)
- content: contents of the page or file
- **description**: contents of the file description (not available for pages)
- **date.edited**: date when page or file was last modified (format is "YYYYMMDDhhmmss"; e.g. "20070801171530" corresponds to 5:15:30pm on August 1st, 2007)
- **path**: the absolute path to the page (not available for files)
- size: the size in bytes of a page or file
- tag: list of all tags on a page (not available for files)
- **title**: the title of a page or the filename of a file
- **type**: the type of the indexed resource ("wiki" for pages, "document" for documents, and "image" for images)
- **uri**: the complete uri for the indexed page or file (deprecated, hence no example. use **path**: instead.)
- **viewcount**: The amount of times a page has been viewed.
- wordcount: The number of words in a page.
- **language**: the language of the page ("fr","en",... use "neutral" for pages without language specified (more info <u>here</u>))

Note: This list is not exhaustive, other fields may be available.

#### **Field Examples**

If you want to find the document entitled "The Right Way" which contains the text "don't go this way", you can enter:

title:"The Right Way" AND content:go

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

title:Do it right

Will only find "Do" in the title field. It will find "it" and "right" in the default field (in this case the text field).

To find all pages that have been viewed more than 20 times, use:

```
type:wiki AND viewcount > 20
```

To find items in a particular location in your wiki, use path. To find files stored in a path called Network, use:

path:Network

But that only includes items stored directly in that location, not anything within any subtrees. To include everything within that path, use:

path:Network OR path:Network/\*

## Search Term Modifiers

Search or query terms can be 'modified' to provide a wide range of searching options.

#### Wildcard Searches

Single and multiple character wildcard searches within single terms (not within phrase queries) are supported:

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

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

te?t

Multiple character wildcard searches looks for 0 or more characters. For example, to search for test, tests or tester, you can use the search:

Search Term Modifiers

test\*

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

te\*t

Note: You cannot use a \* or ? symbol as the first character of a search. (It would require scanning the list of tokens in the index in its entirety to look for those that match the pattern, which could be very expensive.) Lucene 2.1 introduced support for this, turned off by default, but MindTouch does not expose the support at this time.

## **Fuzzy Searches**

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" use the fuzzy search:

roam~

This search will find terms like foam and roams.

An additional (optional) parameter can specify the required similarity. The value is between 0 and 1, with a value closer to 1 only terms with a higher similarity will be matched. For example:

roam~0.8

The default that is used if the parameter is not given is 0.5.

## **Proximity Searches**

You may finding words are a within a specific distance away. To do a proximity search use the tilde, "~", symbol at the end of a Phrase. For example to search for a "codec" and "quicktime" within 10 words of each other in a document use the search:

"codec quicktime"~10

#### **Range Searches**

Range Queries allow you to match documents whose field(s) values are between the lower and upper bound specified by the Range Search. Range Searches can be inclusive or exclusive of the upper and lower bounds. Sorting is done lexicographically.

```
mod_date:[20020101 TO 20030101]
```

This will find documents whose mod\_date fields have values between 20020101 and 20030101, inclusive. Note that Range Searches are **not** reserved for date fields. You could also use range queries with non-date fields:

Search Term Modifiers

#### title:{Home TO Music}

This will find all documents whose titles are between Home and Music, but not including Home and Music.

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

#### Note: the 'TO' operator must be uppecase.

#### Boosting a Search Term

You can control the relevance level of matching documents based on the terms found. To boost a term use the caret, "^", symbol with a boost factor (a number) at the end of the term you are searching. The higher the boost factor, the more relevant the term will be.

Boosting allows you to control the relevance of a document by boosting its term. For example, if you are searching for

video layout

and you want the term "video" to be more relevant boost it using the ^ symbol along with the boost factor next to the term. You would type:

video^4 layout

This will make documents with the term "video" appear more relevant. You can also boost Phrase Terms as in the example:

"video layout"^4 "compatible skin"

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 logic operators, such as AND, "+", OR, NOT and "-" as Boolean operators

#### Note: Boolean operators must be ALL CAPS

#### OR

The **OR** operator is the default conjunction operator. This means that if there is no Boolean operator between two terms, the OR operator is used. The OR operator links two terms and finds a matching document if **either** of the terms exist in a document. This is equivalent to a union using sets. The symbol || can be used in place of the word OR.

To search for documents that contain either "video layout" or just "layout" use the query:

```
"video layout" layout
```

#### **Boolean Operators**

or

```
"video layout" OR layout
```

### AND

The **AND** operator matches documents where **both** terms exist anywhere in the text of a single document. This is equivalent to an intersection using sets. The symbol **&&** can be used in place of the word AND.

To search for documents that contain "video layout" and "filmstrip" use the query:

```
"video layout" AND "filmstrip"
```

#### +

The "+" or required operator requires that the term after the "+" symbol exist somewhere in a field of a single document.

To search for documents that must contain "video" and may contain "layout" use the query:

+video layout

## NOT

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 "video layout" but not "filmstrip" use the query:

```
"video layout" NOT "filmstrip"
```

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

```
NOT "video layout"
```

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

To search for documents that contain "jakarta apache" but not "Apache Lucene" use the query:

"jakarta apache" -"Apache Lucene"

## Grouping

Use parentheses to group clauses to form sub queries. This can be very useful if you want to control the boolean logic for a query.

To search for either "jakarta" or "apache" and "website" use the query:

(jakarta OR apache) AND website

This eliminates any confusion and makes sure that 'website' **must** exist and either term 'jakarta' or 'apache' **may** exist.

# Field Grouping

Use parentheses to group multiple clauses to a single field.

To search for a title that contains both the word "return" and the phrase "pink panther" use the query:

```
title:(+return +"pink panther")
```

## **Escaping Special Characters**

You can escape special characters that are part of the query syntax. The current list of special characters are

• +-&& ||!(){}[]^"~\*?:\

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

(1+1))