Table of Contents

Query Search

Query searches help you search the contents of forms attached to your topics, as well as the values of other meta-data attached to the topic. Using query searches you can search:

1. The fields of forms
2. Parent relationships
3. File attachment information (but **not** the attached files themselves)

Query searches are defined using a simple SQL-like query language. The language consists of *field specifiers* and *constants* joined with *operators*.

Field specifiers

You use field specifiers to say what value from the topic you are interested in.

All meta-data in a topic is referenced according to a simple plan.

- `name` - name of the topic
- `web` - name of the web the topic is within
- `text` - the body text of the topic
- `META:FILEATTACHMENT`
  - `arraysize`
  - *for each attachment*
    - `name`
    - `attr`
    - `path`
    - `size`
    - `user`
    - `rev`
    - `date`
    - `comment`
- `META:TOPICPARENT`
  - `name`
- `META:TOPICINFO`
  - `author`
  - `date`
  - `format`
  - `rev` - topic revision (`12`) to match `%REVINFO{'$rev'}%` and FormattedSearch `$rev`
  - `version` - internal Store topic version (`1.12` for rcs based Stores)
- `META:TOPICMOVED`
  - `by`
  - `date`
  - `from`
  - `to`
- `META:FORM` - the main form of the topic
  - `name` (known as the *formname*)
- `META:FIELD` - the fields in the form.
  - `arraysize`
  - *for each field in the form*
    - `name`
    - `title`
    - `value`
- `META:PREFERENCE`
  - `arraysize`
  - *for each preference in the topic*
    - `name`
    - `value`

See TWikiMetaData for details of what all these entries mean.

Most things at the top level of the plan - `META:TOPICPARENT`, `META:TOPICINFO` etc - are *structures* which are indexed by *keys*. For example, `META:TOPICINFO` has 4 entries, which are indexed by the keys `author`, `date`, `format` and `version`. `META:FILEATTACHMENT`, `META:FIELD` and `META:PREFERENCE` are all *arrays*, which means they can have any number of records under them. Arrays are indexed by *numbers* - for example, the first entry in the `META:FIELD` array is entry 0. The field specifier `arraysize` returns the number of items in an array, for example, `attachments.arraysize` returns the number of attachments.

It's a bit clumsy having to type `META:FILEATTACHMENT` every time you want to refer to the array of attachments in a topic, so there are some predefined aliases that make it a bit less typing:

- `attachments` means the same as `META:FILEATTACHMENT`
- `info` means the same as `META:TOPICINFO`
- `parent` means the same as `META:TOPICPARENT`
- `moved` means the same as `META:TOPICMOVED`
- `form` means the same as `META:FORM`
- `fields` means the same as `META:FIELD`, You can also use the name of the form (the value of `form.name` e.g. `PersonForm`)
- `preferences` means the same as `META:PREFERENCE`

This plan is referenced using a simple syntax:

Syntax	Means	Examples
`X`	refers to the field named `X`.	`info`, `META:TOPICMOVED`, `attachments`, `name`.
`X.Y`	refers to the entry with the key `Y` in the structure named `X`	`info.date`, `moved.by`, `META:TOPICPARENT.name`, `attachments.arraysize`
`X[query]`	refers to all the elements of the array `X` that match *query*. If *query* is of the form `name='Y'` then you can use the same `X.Y` syntax as is used for accessing structures.	`attachments[size>1024]`, `DocumentForm[name!='Summary' AND value~'top secret'].value`, `DocumentForm.Summary`
`X[N]`	where `X` is an array and `N` is an integer number >= 0, gets the Nth element of the array `X`	`attachments[3]`
`X/Y`	accesses `Y` from the topic specified by the *value* of `X`. `X` must evaluate to a topic name	`parent.name/(form.name='ExampleForm')` will evaluate to true if (1) the topic has a parent, (2) the parent topic has the main form type `ExampleForm`.

Note: at some point TWiki may support multiple forms in the same topic. For this reason you are recommended **not** to use the `fields` shortcut when accessing form fields, but always use the name of the form instead.

There is a shortcut for accessing form fields. If you use the name of a field (for example, `LastName`) in the query without a . before it, that is taken to mean "the value of the field named this". This works if and only if the field name isn't the same as of the top level entry names or their aliases described above. For example, the following expressions will all evaluate to the same thing:

- `PersonForm[name='Lastname'].value`
- `Lastname`
- `PersonForm.Lastname`

If `X` would conflict with the name of an entry or alias (e.g. it's `moved` or maybe `parent`), you can prepend the name of the form followed by a dot, as shown in the last example.

Constants

You use constants for the values that you compare with fields. Constants are either strings, or numbers. Strings are always delimited by single-quotes (you can escape a quote using backslash). Numbers can be any integer or floating point number. Constants can be supplied by TWikiVariables, such as `'%TOPIC%'` representing the current topic name.

Operators

Field specifiers and constants are combined using *operators* to create queries.

Operator	Meaning
=	Left-hand side (LHS) exactly matches the value on the Right-hand side (RHS). Numbers and strings can be compared.
!=	Inverse of =.
~	wildcard match ('*' will match any number of characters, '?' will match any single character e.g. "PersonForm.Surname ~ '*Smit?'") Note: Surname ~ 'Smith' is the same as Surname = 'Smith'
<	LHS is less that RHS. If both sides are numbers, the order is numeric. Otherwise it is alphabetic (applies to all comparison operators)
>	>
>=	≥
<=	≤
lc(x)	Converts x to lower case, Use for caseless comparisons.
uc(x)	Converts x to UPPER CASE. Use for caseless comparisons.
d2n(x)	Converts a date (expressed in one of the formats that TWiki can parse) to a number of seconds since 1st Jan 1970. This is the format dates are stored in inside TWiki, and you have to convert a string date using d2n before you can compare it with - for example - the date an attachment was uploaded. Times without a timezone are assumed to be in server local time. If you have date fields in your forms, note that they are **not** stored in TWiki's internal format, but are stored as text strings. You should still use d2n to convert them to numbers for comparisons, though.
NOT	Invert the result of the subquery
AND	Combine two subqueries
OR	Combine two subqueries
()	Bracketed subquery

💡 The same operators are supported for %IF statements.

Putting it all together

When a query is applied to a topic, the goal is to reduce to a TRUE or FALSE value that indicates whether the topic matches that query or not. If the query returns TRUE, then the topic is included in the search results.

A query matches if the query returns one or more values when it is applied to the topic. So if I have a very simple query, such as `"attachments"`, then this will return TRUE for all topics that have one or more attachments. If I write `"attachments[size>1024 AND name ~ '*.gif']"` then it will return TRUE for all topics that have at least one attachment larger than 1024 bytes with a name ending in `.gif`.

Gotcha

- Remember that in the query language, TWiki topic names are *constants*. You cannot write `Main.JohnSmith/UserForm.FirstName` because `Main.JohnSmith` will be interpreted as a form field name. If you want to refer to topics you **must** enclose the topic name in single quotes i.e.

```
'Main.JohnSmith'/UserForm.FirstName
```

Examples

Query examples

- `attachments[name='purdey.gif']` - true if there is an attachment call `purdey.gif` on the topic
- `attachments[name~'*.gif'].arraysize > 3` - true if there are at least 4 `.gif` files attached to the topic
- `(fields[name='Firstname'].value='Emma' OR fields[name='Firstname'].value='John')` `AND fields[name='Lastname'].value='Peel'` - true for 'Emma Peel' and 'John Peel' but **not** 'Robert Peel' or 'Emma Thompson'
- `(Firstname='Emma' OR Firstname='John') AND Lastname='Peel'` - shortcut form of the previous query
- `HistoryForm[name='Age'].value>2` - true if the topic has a `HistoryForm`, and the form has a field called `Age` with a value > 2
- `HistoryForm.Age > 2` - shortcut for the previous query
- `preferences[name='FaveColour' AND value='Tangerine']` - true if the topic has the given preference setting and value
- `Person/(ClothesForm[name='Headgear'].value ~ '*Bowler*' AND` `attachments[name~'*hat.gif' AND date < d2n('2007-01-01')])` - true if the form attached to the topic has a field called `Person` that has a value that is the name of a topic, and that topic contains the form `ClothesForm`, with a field called `Headgear`, and the value of that field contains the string `'Bowler'`, and the topic also has at least one attachment that has a name matching `*hat.gif` and a date before 1st Jan 2007. (Phew!)

Search examples

Find all topics that are children of this topic in the current web

```
%SEARCH{"parent.name = '%TOPIC%'" web="%WEB%" type="query"}%
```

Find all topics that have an attachment called 'grunge.gif'

```
%SEARCH{"attachments[name='grunge.gif']" type="query"}%
```

Find all topics that have form `ColourForm` where the form field 'Shades' is 'green' or 'yellow' but not 'brown'

```
%SEARCH{"(lc(Shades)='green' OR lc(Shades)='yellow') AND NOT(lc(Shades) ~ 'brown')" type="query"}
```

Find all topics that have PNG attachments that have been added since 26th March 2007

```
%SEARCH{"attachments[name ~ '*.png' AND date >= d2n('2007-03-26')]"}%
```

Find all topics that have a field 'Threat' set to 'Amber' and 'cold virus' somewhere in the topic text.

```
%SEARCH{"Threat='Amber' AND text ~ '*cold virus*'"}%
```

Related Topics: IfStatements, SearchHelp, VarIF, VarGET, VarSET, VarSEARCH, FormattedSearch, TWiki:TWiki.QuerySearchPatternCookbook

---

This topic: TWiki > QuerySearch
Topic revision: r7 - 2011-04-04 - TWikiContributor

Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback
Note: Please contribute updates to this topic on TWiki.org at TWiki:TWiki.QuerySearch