Advanced searching

The instructions on this page describe how to define and execute a search in the Project Navigator using the advanced search. You can also define and execute a search using the Basic searching.

On this page

Related topics

- Basic searching
- Using Profields filters
- Export search results to Microsoft Excel

What is an Advanced Search?

An advanced search in the Project Navigator allows you to use structured queries to search for projects. Your search results will be displayed in the Project Navigator, where you can export them to MS Excel. You can also save your advanced searches as Profields filters if you wish.

When you perform an advanced search, you are using our Profields Query Language (PQL).

A simple query in PQL consists of a *field*, followed by an *operator*, followed by one or more *values*. For example, the following simple query will find all projects whose Technology (id=23) is "Java":

```
profield[23] = "Java"
```

(This example uses the Profield's project field "Technology" which id is "23", the EQUALS *operator*, and the *value* `"Java"`.)

Be aware that it is not possible to compare two fields.

How to Perform an Advanced Search

1. Choose **Projects**> **Project Navigator**. The project navigator will be displayed.
    - If there are existing search criteria, click the **New filter** button to reset the search criteria.
    - Click **Advanced** link to switch to advanced searching.
2. Type your query using the fields, operators and field values.
3. Click the **Search** button to run your query.

Switching between 'Advanced' and 'Basic' Search

In general, a query created using 'Basic Searching' will be able to be translated to 'Advanced Searching' (i.e. PQL), and back again.

However, a query created using 'Advanced Searching' may not be able to be translated to 'Basic Searching', particularly if:

- the query contains an OR operator (note you can have an IN operator and it will be translated, e.g. `project in (A, B)`)
- the query contains two or more times the same field.
- the query is too complex to translate to 'Basic Search'.

Setting Precedence of Operators

You can use parentheses in complex PQL statements to enforce the precedence of operators.

For example, if you want to find all development projects which Country (id=27) is China or Continent (id=590) is Asia:

```
category = development AND ( profield[27] ~ China OR profield[590] ~ Asia
)
```

Note that if you do not use parentheses, the statement will be evaluated left-to-right.

You can also use parentheses to group clauses.

Keywords Reference

A keyword in PQL is a word or phrase that does (or is) any of the following:

- joins two or more clauses together to form a complex PQL query
- alters the logic of one or more clauses
- alters the logic of operators
- has an explicit definition in a PQL query
- performs a specific function that alters the results of a PQL query.

List of Keywords

- AND
- OR

AND

Used to combine multiple clauses, allowing you to refine your search.

Note that you can use parentheses to control the order in which clauses are executed.

Example

Find all projects which category are marketing and IT Systems:

```
category = marketing and "IT Systems"
```

OR

Used to combine multiple clauses, allowing you to expand your search.

Note that you can use parentheses to control the order in which clauses are executed.

(Note: also see IN, which can be a more convenient way to search for multiple values of a field.)

Examples

Find all projects which category are marketing or IT Systems::

```
category = marketing or "IT Systems"
```

Fields Reference

A field in PQL is a word that represents a Profields field (a field that has already been defined in Profields) and also especified project fields. In a clause, a field is followed by an operator, which in turn is followed by one or more values. The operator compares the value of the field with one or more values on the right, such that only true results are retrieved by the clause.

Note: these fields does not support auto-complete.

List of Fields:

- Profields fields
- Key
- Category

Profields Fields

A Profields Field is a field that has already been defined in Profields. You can search projects by a Profields field value and can be six different types and each of them have their supported operators.

Syntax

You can search by any project custom field by indicating the word 'profield' and then the 'field id' into square brackets (i.e.: for the text field with id 25 you can write: "profield[25] ~ HOLA"). This feature is only available for non-calculated fields so "Accumulative field" and "Script field" are not supported.

```
profield[fieldid]
```

Profields Field Types

- Text
- User
- List
- Datetime
- Original Estimate
- Float

Text

Supported Operators

=	!=	~	!~	>	>=	<	<=	IN	NOT IN
✗	✗	✓	✓	✗	✗	✗	✗	✗	✗

User

Supported Operators

=	!=	~	!~	>	>=	<	<=	IN	NOT IN
✓	✓	✗	✗	✗	✗	✗	✗	✓	✓

List

Supported Operators

=	!=	~	!~	>	>=	<	<=	IN	NOT IN
✗	✗	✓	✓	✗	✗	✗	✗	✗	✗

Datetime

Supported Operators

=	!=	~	!~	>	>=	<	<=	IN	NOT IN
✓	✓	✗	✗	✓	✓	✓	✓	✓	✓

Original Estimate

Supported Operators

=	!=	~	!~	>	>=	<	<=	IN	NOT IN
✓	✓	✗	✗	✓	✓	✓	✓	✓	✓

Float

Supported Operators

=	!=	~	!~	>	>=	<	<=	IN	NOT IN
✓	✓	✗	✗	✓	✓	✓	✓	✓	✓

Key

This is a field that allows you to search project by its Key.

Syntax

```
key = projectKEY
```

Supported Operators

=	!=	~	!~	>	>=	<	<=	IN	NOT IN
✓	✓	✗	✗	✗	✗	✗	✗	✓	✓

Category

This is a field that allows you to search project by Category ID and Category NAME

Syntax

```
category = categoryID
```

```
category = categoryNAME
```

Supported Operators

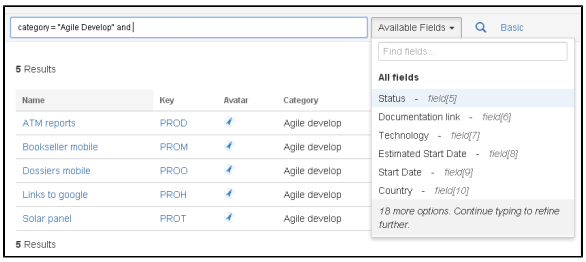
=	!=	~	!~	>	>=	<	<=	IN	NOT IN
✓	✓	✗	✗	✗	✗	✗	✗	✓	✓

Available Fields button

The 'Available Fields' button in Advanced Searching enables you to use Profields fields in a clause without knowing their 'id'.

To use Available Fields button in advanced searching:

1. Click the Available Fields button where you want to use a Profields fields in a clause. All the Profields fields available appears in the dropdown list showing their names and field id.



2. Choose the required field from the list (i.e. Status). The selected field id appears in the clause (fiedl[5]):

3. Now you can type the clause as you required and click the search button  to display the search results.

Operators Reference

An operator in PQL is one or more symbols or words which compares the value of a field on its left with one or more values on its right, such that only true results are retrieved by the clause.

List of Operators:

- EQUALS: =
- NOT EQUALS: !=
- GREATER THAN: >
- GREATER THAN EQUALS:>=
- LESS THAN: <
- LESS THAN EQUALS: <=
- IN
- NOT IN
- CONTAINS: ~

EQUALS: =

The "=" operator is used to search for projects where the value of the specified field exactly matches the specified value. (Note: cannot be used with text fields; see the CONTAINS operator instead.)

To find projects where the value of a specified field exactly matches *multiple* values, use multiple "=" statements with the AND operator.

Examples

Find all projects which "Category" is equals to "marketing":

```
category = marketing
```

Find all projects which "Category" is equals to "IT Systems":

```
category = "IT Systems"
```

NOT EQUALS: !=

The "!=" operator is used to search for projects where the value of the specified field does not match the specified value.

Examples

Find all projects which "Category" is not equals to "marketing":

```
category != marketing
```

Find all projects which "Category" is not equals to "IT Systems":

```
category != "IT Systems"
```

GREATER THAN: >

The ">" operator is used to search for projects where the value of the specified field is greater than the specified value.

To see a field's supported operators, check the individual field reference.

Examples

Find all projects which Budget (id=345) greater than 100k:

```
profield[345] > 100000
```

Find all overdue projects which End date (id=298) greater than 2014/06/30:

```
profield[298] > 2014/06/30
```

Find all projects which Effort Estimated (id=37) greater than six weeks:

```
profield[37] > 6w
```

GREATER THAN EQUALS: >=

The ">=" operator is used to search for projects where the value of the specified field is greater than or equal to the specified value.

To see a field's supported operators, check the individual field reference.

Examples

Find all projects which Budget (id=345) greater than or equals to 100k:

```
profield[345] >= 100000
```

Find all overdue projects which End date (id=298) greater than or equals to 2014/06/30:

```
profield[298] >= 2014/06/30
```

Find all projects which Effort Estimated (id=37) greater than or equals to six weeks:

```
profield[37] >= 6w
```

LESS THAN: <

The "<" operator is used to search for projects where the value of the specified field is less than the specified value.

To see a field's supported operators, check the individual field reference.

Examples

Find all projects which Budget (id=345) less than 100k:

```
profield[345] < 100000
```

Find all overdue projects which End date (id=298) less than 2014/06/30:

```
profield[298] < 2014/06/30
```

Find all projects which Effort Estimated (id=37) less than six weeks:

```
profield[37] < 6w
```

LESS THAN EQUALS: <=

The "<=" operator is used to search for projects where the value of the specified field is less than or equal to than the specified value.

To see a field's supported operators, check the individual field reference.

Examples

Find all projects which Budget (id=345) less than or equals to 100k:

```
profield[345] <= 100000
```

Find all overdue projects which End date (id=298) less than or equals to 2014/06/30:

```
profield[298] <= 2014/06/30
```

Find all projects which Effort Estimated (id=37) less than or equals to six weeks:

```
profield[37] <= 6w
```

IN

The "IN" operator is used to search for projects where the value of the specified field is one of multiple specified values. The values are specified as a comma-delimited list, surrounded by parentheses.

Example

Find all projects which categories are marketing or IT Systems:

```
category in (marketing, "IT Systems")
```

NOT IN

The "NOT IN" operator is used to search for projects where the value of the specified field is not one of multiple specified values. The values are specified as a comma-delimited list, surrounded by parentheses.

Example

Find all projects which categories are not marketing nor IT Systems:

```
category not in (marketing, "IT Systems")
```

CONTAINS: ~

The "~" operator is used to search for projects where the value of the specified field matches the specified value.

The "%" sign is used to define wildcards (missing letters) both before and after the pattern.

Examples

Find all projects where the Customer (id=944) is "DEISER":

```
profield[944] ~ "DEISER"
```

Find all projects where the Customer (id=944) starts with "DEISER":

```
profield[944] ~ "DEISER%"
```

Find all projects where the Customer (id=944) ends with "DEISER":

```
profield[944] ~ "%DEISER"
```

Find all projects where the Customer (id=944) contains "DEISER":

```
profield[944] ~ "%DEISER%"
```