Important: The information on this page applies to searches made using locally stored indexes. If you are using Azure Search indexes, refer to the Simple query syntax and Lucene query syntax articles to learn about the supported syntax options.

Users can submit advanced search expressions using the Lucene query parser syntax. You can find detailed information at: http://lucene.apache.org/core/old_versioned_docs/versions/3_0_3/queryparsersyntax.html

To allow the advanced syntax, configure the **Search options** property of the **Smart search dialog with results** or **Smart search results** web part. You can choose one of the following levels of supported syntax:

- **None** – the search does not recognize any Lucene query syntax. The system processes all text entered by users as a part of the search expression.
- **Basic** – the search recognizes all query syntax, *except for field searching*.
- **Full** – the search processes all search query syntax, including field searching.

The system only processes advanced query syntax correctly for search requests that use the **Any word** Search mode. Other search modes may interfere with the syntax. For example, using the OR boolean operator contradicts the *All words* search mode.

Search syntax errors

If you allow advanced search expressions through the **Search options** property, a parsing error occurs if a user enters the query syntax incorrectly.

By default, syntax errors cause the search to show the standard "No results" message, and the system enters the error into the application's Event log. If you wish to view syntax errors in place of the results directly on the live site, enable the **Show parsing errors** property of the given smart search results web part.

Field search

Field searching allows users to define additional conditions in search expressions. All conditions must start with either the + or - symbol. The + symbol indicates that only results which fulfill the field condition should be returned. The - symbol has the opposite meaning, only results that do **not** contain the specified value in the given field are retrieved.

For example:

- +network +NewsReleaseDate:[20080101 TO 20091231]

When searching for this expression using a page index, the smart search returns only news pages containing the word *network*, released in the year 2008 or 2009.

Field search requirements

- Field search only works for fields that have the **Searchable** option enabled in the **Local** search field configuration of the searched object type. If there is no field name specified before a value in the search expression (such as the word *network* in the example above), the system searches the index fields marked as **Content**.
- The system only processes field searches correctly if the search dialog web part has its **Search mode** set to *Any words*.

Unless the web part used to display the search results has the **Block field-only search** property enabled, it is also possible to perform direct field searches without any standard content keywords. This allows users to find records simply by entering an exact field value:

- *DocumentNodeID:(int)17* – returns the page with a nodeID equal to *17*.
- *NewsTitle:"New features"* – returns the news page titled *New features*.

- *SKUDepartmentID:(int)4 OR (int) 6* – returns all products that belong to one of the specified departments (with an ID of either *4* or *6*).

✓ **Tip**: Using field search queries provides a great deal of flexibility, but is not convenient for regular website visitors. If you wish to allow users to limit the scope of searches through conditions on the live site, we recommended creating Search filters.

Searching text fields

When searching in text fields, you do not need to specify the type of the value. However, keep in mind the following when searching for values containing **spaces**:

- Search values containing one or more spaces must be enclosed in quotes, for example: *NewsTitle:"New features"*
- Text fields must have the **Tokenized** option enabled in the **Local** field configuration of the searched object type in order for the search to find matches for values containing spaces.

Searching numeric fields

When performing field searches, the values specified are processed as strings by default. If you are searching in **Integer**, **Floating-point (double)** or **Decimal** type fields, you need to explicitly specify the data type of values in the search syntax:

- NewsID:**(int)**22
- SKUPrice:**(double)**255.0
- DecimalField:**(decimal)**9.99
- DocumentNodeID:[**(int)**1 TO **(int)**100]

Searching date and time fields

Use the following syntax to search in DateTime fields: **<field name>:yyyymmddhhmm**.

For example:

- DocumentCreatedWhen:200812230101
- DocumentCreatedWhen:[200902020101 TO 200906020101]

If you need to specify date and time values through macros, call the **ToSearchDateTime** method to convert the values to the suitable format. For example:

- {% ToSearchDateTime(CurrentDateTime) %}
- {% ToSearchDateTime(CurrentDateTime.AddHours(12)) %}
- {% ToSearchDateTime(CurrentUser.UserCreated, CurrentDateTime) %}

The *ToSearchDateTime* method's second optional parameter sets a default DateTime value returned if the first parameter is null.

Field search with Stop and Simple analyzers

Indexes created by Stop and Simple analyzers cannot be searched using the standard field search format. This is by design, but you can use a workaround with a range query containing identical boundaries:

- newsid:[(int)22 TO (int)22]