Table of Contents

1 Searching overview ........................................................................................................... 6
2 Filters ................................................................................................................................. 8
   2.1 Setting filters .................................................................................................................. 8
   2.1.1 Changing item sets per page ..................................................................................... 9
   2.1.2 Saving filters as a search ............................................................................................ 10
   2.2 Filter types .................................................................................................................... 10
      2.2.1 Boolean filters .......................................................................................................... 10
      2.2.2 Custom only .............................................................................................................. 11
      2.2.3 List filters ................................................................................................................ 12
      2.2.4 MultiList filters ....................................................................................................... 13
      2.2.5 Popup picker filters .................................................................................................. 13
      2.2.6 Textbox filters ........................................................................................................ 14
   2.3 Textbox filter search examples ....................................................................................... 15
      2.3.1 Boolean and other search operators ....................................................................... 15
      2.3.2 Alphabetical filtering ............................................................................................... 15
      2.3.3 Dates and numbers ................................................................................................... 16
3 Keyword search .................................................................................................................. 18
   3.1 Fields ................................................................................................................................ 18
   3.2 Example keyword search strings ................................................................................... 18
   3.3 Using the NOT operator in keyword searches ................................................................. 19
   3.4 Understanding stop words ............................................................................................. 20
      3.4.1 Single letters as stop words ...................................................................................... 20
      3.4.2 Numbers as stop words ............................................................................................ 20
      3.4.3 Punctuation as stop words ....................................................................................... 21
      3.4.4 At sign (@) ................................................................................................................ 21
      3.4.5 Hyphens and dashes .................................................................................................. 21
      3.4.6 Default stop word list ............................................................................................... 21
   3.5 Running a keyword search .............................................................................................. 22
4 Search conditions ............................................................................................................... 23
5.8.10 W/N operator .................................................................................................. 54
5.8.11 Numerical patterns .......................................................................................... 57
5.8.12 Connector words ............................................................................................. 57
5.8.13 Words and phrases .......................................................................................... 58
5.9 dtSearch default alphabet file text ....................................................................... 58

6 Saved search ............................................................................................................ 61
  6.1 Navigating the saved searches browser ............................................................. 62
    6.1.1 Using search list options .............................................................................. 64
  6.2 Controlling the visibility of saved searches ....................................................... 66
  6.3 Organizing saved searches in folders ............................................................... 66
    6.3.1 Adding sub-folders to the root ................................................................. 66
    6.3.2 Managing subfolders ............................................................................... 67
    6.3.3 Adding existing searches to folders ......................................................... 68
  6.4 Creating or editing a saved search ................................................................. 68
    6.4.1 Fields ....................................................................................................... 69
    6.4.2 Using pop-up pickers ............................................................................... 71
    6.4.3 Rerunning out-of-date saved searches .................................................... 72
    6.4.4 Copying a saved search to another workspace ....................................... 73
  6.5 Defining criteria for saved searches .............................................................. 73
    6.5.1 Operators .................................................................................................. 73
    6.5.2 Batch fields as search conditions ............................................................. 76
    6.5.3 Using saved searches as conditions (combined searches) ...................... 77
    6.5.4 Lists as search conditions ...................................................................... 78
  6.6 Saving searches on the Documents tab .......................................................... 79
  6.7 Saved search history ...................................................................................... 80

7 Searching with Regular Expressions (RegEx) ....................................................... 82
  7.1 Use cases for Regular expressions .................................................................. 82
  7.2 Regular expression metacharacters ............................................................... 82
    7.2.1 RegEx quantifiers .................................................................................... 85
    7.2.2 Escaping RegEx Metacharacters ............................................................. 87
  7.3 Using regular expressions with dtSearch ...................................................... 87
7.3.1 RegEx search strings ............................................................................................................ 88
7.3.2 Common dtSearch RegEx examples ........................................................................................ 88

8 Optimized indexing .......................................................................................................................... 91
1 Searching overview

Relativity includes flexible search features designed to facilitate the document review process. These easily accessible features support a range of searching needs from filtering on fields and simple keyword searches to the development of complex queries. The following list summarizes the searching features available in Relativity.

Filters
You can use filters to limit the documents or items that appear in item lists on Relativity tabs and pop-ups. When you enable the filters for an item list, you can set criteria on single or multiple fields so that only matching documents or items appear in the view. Filters query across the searchable set of documents in the active view to return your results. Relativity supports multiple filter types so that you can easily choose the best format for different field types. See the Searching Guide for more information.

Keyword searches
You can run keyword searches from the Documents tab and from Dynamic Object tabs. With these searches, you can leverage the basic functionality for querying the SQL full-text index populated with data from extracted text fields. The keyword search engine supports the use of Boolean operators and wildcards. As the default search engine in Relativity, keyword search automatically populates with extracted text during data import. See the Searching Guide for more information.

Saved searches
These searches provide you with the functionality to define and store queries for repeated use. With flexible settings, you can create a saved search based on any Relativity search engine, assign security permissions to it, and define specific columns to display your search results. Saved searches support the development of complex queries that you build using a form with search condition options. These queries run dynamically to ensure that updated results appear when you access a saved search. See the Searching Guide for more information.

dtSearches
Available on the Documents tab, you can use the advanced searching functionality to run queries with proximity, stemming, and fuzziness operators, as well as with basic features such as Boolean operators and wildcards. Relativity administrators can create a dtSearch index for a specific subset of documents in a workspace, and then assign security to it. They must manually update indexes when the document search sets used to create them are modified. See the Searching Guide for more information.

Analytics
Supporting conceptual searching, Analytics includes documents in a result set when they contain similar ideas or conceptual relationships, rather than matching specific search terms or conditions. You can create searches with Analytics that categorize your documents based on the concepts contained in a sample document set. Instead of categorizing documents, you can also perform clustering, which uses specific algorithms (system-defined rules) to identify conceptually related documents. See the Searching Guide for more information.

Regular expressions
Regular Expressions (RegEx) is a form of advanced searching that looks for specific patterns, as opposed to certain terms and phrases. With RegEx you can use pattern matching to search for particular strings of characters rather than constructing multiple, literal search queries. You can use RegEx with a dtSearch index using dtSearch syntax options to construct complex queries. See the Searching Guide for more information.
**Additional features**
Relativity provides additional features that make searching easily accessible from the Documents tab. With the **search condition** option, you can build queries using the same condition options available for saved searches. You can click **Save as Search** on the Documents tab to create saved searches based on the criteria defined for keyword searches, dtSearches, Analytics, or the search conditions option. See the Searching Guide for more information.

**Note:** If you're interested in learning about the technicalities of how Relativity executes the searches that you run, see "Searching behind the scenes" in the System Guides section of the Relativity Documentation Site.
2 Filters

Filtering provides a fast and easy way to search for items in a list in Relativity. You can use filters to search for values in the fields displayed on the active view and across all records available in the searchable set. Filters are available for item lists displayed on tabs and popup windows.

The field type associated with each column determines the available filter types, such as textboxes, popup pickers, and drop-down lists. You don't need any specific security permissions to use filtering; simply go to the item list or tab where you want to use it.

Using filters

You're a Relativity admin facilitating a review project on a huge set of documents that includes thousands of emails between traders and investors in your client’s products. You need to quickly find all emails sent by trader David Forster that are related to energy trading, because the evidence they might contain may not be admissible in court. To do this, you turn to filters.

You find the view you set up for documents already coded for key issues and you click the filter icon. In the Email From field you enter the last name of the employee in question, "Forster." In the Key Issue field you select the "Energy Trading" choice.

This simple filtering job returns a small set of documents on which you do a quick review and find that they contain references that make them inadmissible. You then flag those documents for removal, thus ensuring that they don't make it into saved searches or production sets.

2.1 Setting filters

The following filter buttons display above item lists in Relativity:
Show/Hide filters - Activates the filters for an item list, displaying a filtering option at the top of each column. Hides filtering options at the top of the list columns if filtering options are available. Your filter settings remain unchanged.

Save as search - Allows you to save the filter conditions as a saved search and provide a meaningful name for the saved search.

Clear all - Removes the current filter settings. This option is only available after you set a filter.

The filter type determines the steps required to set the search criteria that it uses.

**Note:** When you create or edit a field, you can select its Filter Type. See Fields in the Admin guide.

To activate the filter options at the top of each column in the item list, click the icon. In the following example, the active filters display on the Documents tab.

You can use filters to narrow your searchable set, allowing you to browse through the key documents in the viewer.

For example, let’s say your workspace is 120,000 records and you’d like to see documents JGRIFFITH_0000544 through JGRIFFITH_0000688. To return those documents, click in the navigation bar, enter JGRIFFITH_0000544 BETWEEN JGRIFFITH_0000688 in the Control Number field and hit Enter. The results become your searching set, allowing you to browse the documents.

Before defining a new search filter though, click to remove the previous settings. For example, say you create a filter on the Privileged field, but now want to search only by the Custodian field. If you don’t clear the filter, Relativity filters your documents by both the Privileged and Custodian fields, and your result set may include fewer documents than you expect.

See Filter types on the next page and Textbox filter search examples on page 15 for more information on using filters.

**2.1.1 Changing item sets per page**

You can use the set selector menu to change the number of items that appear per page. The set selector menu appears at the bottom of the screen. The option you select remains the default setting during your session until you select another option.
2.1.2 Saving filters as a search
To save your filtered item set as a saved search:

1. Click to open the Save Search window.
2. Enter the saved search name.
3. Select the folder in which you want to put the saved search.
4. Click Save.

2.2 Filter types
Relativity includes the following filter types:

2.2.1 Boolean filters
Boolean filters are available only on Yes/No field types. To use a Boolean filter, click the drop-down menu arrow and select Yes, No, or Blank from the list.
In the following example, setting the filter to No displays only documents that don't have native files. If selected, Blank displays only items that don't have the a Yes or No value assigned to the Boolean field.

2.2.2 Custom only

The CustomOnly filter allows you to set filtering criteria using a combination of advanced and Boolean operators.

To use a CustomOnly filter, click the drop-down menu arrow and select (Advanced...) to display a popup with filtering options. The field type determines the operators available in the popup.

To create a search criterion, select an operator from the drop-down box, and enter a value in the textbox. You can connect two search criteria with the AND or OR operators.

Click OK to set the filter. You can modify the filter by selecting (Change Advanced...) in the drop-down menu.

If the CustomOnly filter is not available for a desired field, perform the following steps:

1. Navigate to the Fields tab.
2. Locate the field that you want to enable for the CustomOnly filter and click Edit.
3. Go to the List Properties category of the field layout.
4. Select the drop-down for the Filter Type field, select CustomOnly, and click Save.

The CustomOnly filter type is available for the following field types only:
2.2.3 List filters

List filters are frequently associated with fields used for coding documents. The conditions displayed in the drop-down menu for this filter vary by the type and purpose of the associated field. For example, a field called Responsiveness might have the filter conditions of Responsive, Not Responsive, or Not Sure. Similar to Boolean filters, you only filter on one condition at a time.

In this example, if you select the Non-Responsive condition, then only documents that have been coded with this value will display in the item list.

Notes:
- Fields that contain a large number of items may also cause the List filter to take longer to populate. We recommend you only select the List filter type for Single Choice and Yes/No fields. For fields that may contain a large number of choices, you can designate a Popup filter instead.
- **Fixed Length Text fields** - if the field contains more than 255 items, the List filter type you specify for the field will revert to the Textbox filter type and an error will be written to the Error Log when you visit a Document list view containing the field.
- **Single Choice / Single Object fields** - if the field contains more than 255 items, the List filter type you specify for the field will revert to None (disabled) and an error will be written to the Error Log when you visit a Document list view containing the field.
2.2.4 MultiList filters

Using MultiList filters, you can select multiple conditions from a drop-down menu and connect them with the OR or AND operator. These filters are frequently associated with fields used for coding documents, and their conditions vary by the type and purpose of the field.

To apply a MultiList filter, click the drop-down arrow to display the conditions list. Select the conditions that you want to filter on, and clear those that you don't want to filter on. Select the OR to set the filter to return items with one or more of the specified conditions, or select AND to set the filter to return items with all of the specified conditions.

**Note:** Selecting the (Not Set) condition returns fields that don't have a value assigned for the field.

Click **Set** to display only the items that meet the filter conditions.

![MultiList filter example](image)

Clear the (Show All) checkbox to remove all selected conditions.

2.2.5 Popup picker filters

Using the popup picker filter, you can select values for multiple/single choice fields or multiple/single object fields.

To select items in the popup picker, click the **Select** button. Select one or more items in the Available Items list.
Click **Add** to move the items to the Selected Items list.

**Note:** To remove an item from the Selected Items list, select it and click **Remove**. It should now appear in the Available Items list.

Click **Set** to display only the items that meet the filter conditions. The items now appear in the **Value** box in the search form.

### 2.2.6 Textbox filters

You can use textbox filters to search on specific terms, numbers, and dates. You can enter one or more terms in the textbox and connect multiple terms with any of the following operators:

- **AND**
- **OR**
- **BETWEEN**
- **=** (equal)
- >= (greater than or equal to)
- <= (less than or equal to)

When entering terms and operators in a textbox filter, you must use the proper syntax. See Textbox filter search examples below for more information.

The textbox filter treats each search term as if it were preceded and followed by a wildcard (*) and returns all versions of the term.

**Note:** Don't add an asterisk (*) to the beginning or ending of a search term. The filter won't return any results if you use this operator.

### 2.3 Textbox filter search examples

You can perform text searches using Boolean operators as well as filter alphabetically or numerically using greater than (> or equal to (=) and other operators.

#### 2.3.1 Boolean and other search operators

The following table lists examples of valid search strings using Boolean and other operators.

<table>
<thead>
<tr>
<th>Valid search strings</th>
<th>Returns items where...</th>
</tr>
</thead>
<tbody>
<tr>
<td>cubs OR sox</td>
<td>([FIELD VALUE] like &quot;cubs&quot;) OR ([FIELD VALUE] like &quot;sox&quot;)</td>
</tr>
<tr>
<td>cubs AND sox</td>
<td>([FIELD VALUE] like &quot;cubs&quot;) AND ([FIELD VALUE] like &quot;sox&quot;)</td>
</tr>
<tr>
<td>cubs OR sox AND relativity</td>
<td>([FIELD VALUE] like &quot;cubs&quot;) OR (([FIELD VALUE] like &quot;sox&quot;) AND ([FIELD VALUE] like &quot;relativity&quot;))</td>
</tr>
<tr>
<td>percent sign (%)</td>
<td>Use this operator to check whether the field is set to a value. It behaves like the &quot;Is like&quot; operator in a query.</td>
</tr>
<tr>
<td>underscore (_)</td>
<td>Wildcard for a missing character. Don't use the underscore to check if a field is set to a value; it's slower and more resource-intensive than using the percent sign (%).</td>
</tr>
<tr>
<td>= with term</td>
<td>Returns an exact phrase.</td>
</tr>
<tr>
<td>cubs sox</td>
<td>Returns the exact phrase (that is, the word &quot;cubs&quot; followed by a space and the word &quot;sox&quot;).</td>
</tr>
</tbody>
</table>

The following table shows examples of invalid search strings.

<table>
<thead>
<tr>
<th>Invalid search strings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cubs AND</td>
<td>The AND operator requires a right search term.</td>
</tr>
<tr>
<td>cubs OR</td>
<td>The OR operator requires a right search term.</td>
</tr>
<tr>
<td>AND cubs</td>
<td>The AND operator requires a left search term.</td>
</tr>
<tr>
<td>OR cubs</td>
<td>The OR operator requires a left search term.</td>
</tr>
</tbody>
</table>

#### 2.3.2 Alphabetic filtering

The following table lists examples of valid search strings you can use to filter text alphabetically.
Alphabetical filtering

<table>
<thead>
<tr>
<th>Filtering</th>
<th>Returns items where...</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;= cubs</td>
<td>[FIELD VALUE] &gt;= 'cubs'</td>
</tr>
<tr>
<td>&lt;= cubs</td>
<td>[FIELD VALUE] &lt;= 'cubs'</td>
</tr>
<tr>
<td>= cubs</td>
<td>[FIELD VALUE] = 'cubs'</td>
</tr>
<tr>
<td>= cubs AND sox</td>
<td>[FIELD VALUE] = 'cubs AND sox'</td>
</tr>
<tr>
<td>cubs BETWEEN sox*</td>
<td>([FIELD VALUE] &gt;= 'cubs') AND ([FIELD VALUE] &lt;= 'sox')</td>
</tr>
<tr>
<td>relativity and cubs BETWEEN sox</td>
<td>([FIELD VALUE] &gt;= 'relativity and cubs') AND ([FIELD VALUE] &lt;= 'sox')</td>
</tr>
</tbody>
</table>

* If you attempt to use more than one BETWEEN operator in a single filter string (for example, 12/13/2000 BETWEEN 1/0/2008 BETWEEN 5/4/2009), you'll receive an Incorrect Syntax error in the filter box. If you need to search for documents based on multiple BETWEEN operators, you can create a saved search with multiple conditions, each of which uses a date field with a between operator and a date range value. For more information, see the Saved search documentation on our website.

2.3.3 Dates and numbers

The following table lists examples of valid date and number searches, as well as the expected result set.

<table>
<thead>
<tr>
<th>Valid search strings</th>
<th>Returns items where...</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;= 7/24/2008</td>
<td>[FIELD VALUE] &gt;= '7/24/2008'</td>
</tr>
<tr>
<td>&lt;= 7/24/2008</td>
<td>[FIELD VALUE] &lt;= '7/24/2008'</td>
</tr>
<tr>
<td>= 7/24/2008</td>
<td>([FIELD VALUE] = '7/24/2008')</td>
</tr>
<tr>
<td>07/27/2008</td>
<td>([FIELD VALUE] = '07/27/2008')</td>
</tr>
<tr>
<td>&gt;= 100</td>
<td>[FIELD VALUE] &gt;= '100'</td>
</tr>
<tr>
<td>&lt;= 100</td>
<td>[FIELD VALUE] &lt;= '100'</td>
</tr>
<tr>
<td>= 100</td>
<td>[FIELD VALUE] = '100'</td>
</tr>
</tbody>
</table>

The following table includes examples of invalid data and number search strings.
<table>
<thead>
<tr>
<th>Invalid search strings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 7/24/2008</td>
<td>The equal sign must be used with the greater than operator (as in &gt;=).</td>
</tr>
<tr>
<td>&lt; 7/24/2008</td>
<td>The equal sign must be used with the less than operator (as in &lt;=).</td>
</tr>
<tr>
<td>&gt;= 0/24/2008</td>
<td>The search string includes the value 0 for the month.</td>
</tr>
<tr>
<td>= 0/24/2008</td>
<td>The search string includes the value 0 for the month.</td>
</tr>
<tr>
<td>0/24/2008 BETWEEN 8/24/2008</td>
<td>The search string includes the value 0 for the month in the starting date.</td>
</tr>
<tr>
<td>7/24/2008 BETWEEN 0/24/2008</td>
<td>The search string includes the value 0 for the month in the ending date.</td>
</tr>
</tbody>
</table>
3 Keyword search

Keyword search (or SQL index search) is Relativity’s default search engine. You can use a keyword search to query a full text index. The extracted-text fields included in this index vary by workspace.

You can use Boolean operators (AND, OR, NOT) in keyword searches, as well as quotation marks for exact matches, asterisks (*) for wildcards, and other features. However, if you perform a keyword search with multiple terms, documents where those terms exist in separate fields won’t return.

**Note:** If you want to draft queries outside of Relativity, use a plain text editor such as Microsoft Notepad to prevent adding characters or formatting that might return unexpected search results.

While the keyword search offers fewer options than other Relativity searches, it uses an index that’s automatically populated, reducing maintenance and ensuring all required document fields are indexed.

3.1 Fields

A keyword search index is available in the Search Indexes tab by default. Click the **Keyword Search** link.

The keyword search index details page contains the following fields:

- **Name** is the name of the keyword search index. The name is used as the display name for the index in the documents tab search with drop-down menu.

- **Order** is a number that represents the position of the index in the list. The lowest-numbered index will be at the top. The highest-numbered index is at the bottom. Items that share the same value are sorted in alphanumeric order. Index order can be any integer (positive or negative). No decimals are allowed.

- **Active** allows you to determine whether the index should be activated or deactivated. **Yes** means that the index will be activated; **No** means that the index will be deactivated.

3.2 Example keyword search strings

The following table lists search string examples with their expected results.

---

**3 Keyword search**

Keyword search (or SQL index search) is Relativity's default search engine. You can use a keyword search to query a full text index. The extracted-text fields included in this index vary by workspace.

You can use Boolean operators (AND, OR, NOT) in keyword searches, as well as quotation marks for exact matches, asterisks (*) for wildcards, and other features. However, if you perform a keyword search with multiple terms, documents where those terms exist in separate fields won't return.

**Note:** If you want to draft queries outside of Relativity, use a plain text editor such as Microsoft Notepad to prevent adding characters or formatting that might return unexpected search results.

While the keyword search offers fewer options than other Relativity searches, it uses an index that's automatically populated, reducing maintenance and ensuring all required document fields are indexed.

3.1 Fields

A keyword search index is available in the Search Indexes tab by default. Click the **Keyword Search** link.

The keyword search index details page contains the following fields:

- **Name** is the name of the keyword search index. The name is used as the display name for the index in the documents tab search with drop-down menu.

- **Order** is a number that represents the position of the index in the list. The lowest-numbered index will be at the top. The highest-numbered index is at the bottom. Items that share the same value are sorted in alphanumeric order. Index order can be any integer (positive or negative). No decimals are allowed.

- **Active** allows you to determine whether the index should be activated or deactivated. **Yes** means that the index will be activated; **No** means that the index will be deactivated.

3.2 Example keyword search strings

The following table lists search string examples with their expected results.
<table>
<thead>
<tr>
<th>Search string</th>
<th>Returns documents with...</th>
</tr>
</thead>
<tbody>
<tr>
<td>wired</td>
<td>the word wired</td>
</tr>
<tr>
<td>wired magazine</td>
<td>the words wired and magazine</td>
</tr>
<tr>
<td>wired AND magazine</td>
<td>the words wired and magazine</td>
</tr>
<tr>
<td>wired OR magazine</td>
<td>the word wired or the word magazine</td>
</tr>
<tr>
<td>wired, magazine</td>
<td>the word wired or the word magazine</td>
</tr>
<tr>
<td>&quot;wired magazine&quot;</td>
<td>the exact phrase wired magazine</td>
</tr>
<tr>
<td>wired NOT magazine</td>
<td>the word wired and not the word magazine</td>
</tr>
</tbody>
</table>

**Note:** Do not start key word searches with the NOT operator, or use it with the OR operator. For example, these searches are invalid:

- not wired
- wired or not magazine

See NOT Operator Evaluation in Keyword searches.

| wire*               | any words beginning with wire, such as wired, wires, wireless |

**Note:** Key word searches do not support the use of wildcards at the beginning of a word. (Keyword searches are SQL index searches run on the Microsoft SQL Server, which does not support leading wildcards in full text searches.)

<table>
<thead>
<tr>
<th>Document</th>
<th>OCR</th>
<th>Recipient</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS00001</td>
<td>From: Ryan To: Will</td>
<td>Will</td>
<td>Ryan</td>
</tr>
</tbody>
</table>

**Note:** When a search string does not include parentheses, the order of precedence for a keyword search evaluates AND then OR expressions. For example, the search string A AND B OR C is evaluated as (A AND B) OR C.

### 3.3 Using the NOT operator in keyword searches

When running a keyword search that is an SQL full text search, carefully format queries that use the NOT operator. For example, you may want to query for email messages that have Ryan as the author, but do not have Will as the recipient. The fields in the following record are included in the index used to demonstrate how this query is run:

A keyword search using the string *Ryan NOT Will* returns the document AS00001 even though you would not expect it in the result set. The following table illustrates the SQL logic used to evaluate the query Ryan NOT Will.
### 3.4 Understanding stop words

When you run a keyword search, stop words are ignored because they do not act as meaningful criteria in a query. They include characters (such as punctuation marks and single letters), numbers, and words (such as *at, a, on and the*).

While keyword searches ignore stop words, their position in a phrase is taken into account when the query is executed. Records that contain phrases with same number of intervening stop words will be returned even when the stop words differ from those in the original phrase.

For example, the same set of documents is returned when you run the queries for the phrase *sun on my head,* or *sun my on head.* The result set will include documents that contain the words *sun* and *head* separated by two intervening stop words. A query for the phrase *sun on head* will return only those documents with one intervening stop word. The following table illustrates how the stop words in these phrases are handled.

<table>
<thead>
<tr>
<th>Searching string</th>
<th>Queries for the phrase...</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun on my head</td>
<td>&quot;sun [stop word] [stop word] head&quot;</td>
</tr>
<tr>
<td>sun my on head</td>
<td>&quot;sun [stop word] [stop word] head&quot;</td>
</tr>
<tr>
<td>sun on head</td>
<td>&quot;sun [stop word] head&quot;</td>
</tr>
</tbody>
</table>

However, if you replace a stop word with a meaningful word (such as *sun [stop word] tree head*), no documents will be returned since they do not contain this phrase. When a stop word is used in a keyword search it can be replaced by any word in the results. Searching *sun on my head* would return a document that has the phrase *sun on street head where street* is not a stop word.

#### 3.4.1 Single letters as stop words

Single uppercase and lowercase letters are default stop words, so you cannot query on them with a keyword search. Each single letter [A-Z and a-z] is considered a stop word.

However, you can query on a capital letter followed by a period, which the SQL search engine assumes is an abbreviation. For keyword searches, this functionality is available only on queries in the English language. It does not apply to lowercase letters followed by a period, which are still considered stop words.

#### 3.4.2 Numbers as stop words

Single digits 0-9 are default stop words, so you cannot query on them with a keyword search. You will not return the expected results if you attempt to query on a single digit. Use the dtSearch feature to query on a specific number or letter.
However, you can use a keyword search to query on whole numbers greater than 9. You can search on more than one digit, such as 09. While these digits may be used to represent a specific numeric value (such as 9), they are not considered single digits, and can be used in a keyword search.

### 3.4.3 Punctuation as stop words

Certain punctuation marks are treated as stop words by default, so you cannot query on them with a keyword search. They include:

- Period (.)
- Dash (–)
- Colon (:
- Semicolon (;)
- Slash (/)

### 3.4.4 At sign (@)

The at sign (@) is ignored in a keyword search, when it is used at the beginning of a query. For example, if you search a domain name, the same number of documents will be returned whether you include or exclude @.

### 3.4.5 Hyphens and dashes

When a search phrase includes a hyphen or dash, the query will return results that include terms containing other punctuation marks. For example, the following results will be returned for a search on the term Pop-up:

- Popup
- Pop.up
- Pop--up

### 3.4.6 Default stop word list

Relativity comes with the following default stop words:

<table>
<thead>
<tr>
<th>Begins with...</th>
<th>Stop words</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>about, after, all, also, another, any, are, as, at</td>
</tr>
<tr>
<td>B</td>
<td>be, because, been, before, being, between, but, both, by</td>
</tr>
<tr>
<td>C</td>
<td>came, can, come, could</td>
</tr>
<tr>
<td>D</td>
<td>did, do, does</td>
</tr>
<tr>
<td>E</td>
<td>each, else</td>
</tr>
<tr>
<td>F</td>
<td>for, from</td>
</tr>
<tr>
<td>G</td>
<td>get, got</td>
</tr>
<tr>
<td>H</td>
<td>has, had, he, have, her, here, him, himself, his, how</td>
</tr>
<tr>
<td>I</td>
<td>if, in, into, is, it, its</td>
</tr>
<tr>
<td>J</td>
<td>just</td>
</tr>
</tbody>
</table>
3.5 Running a keyword search

Use the following steps to run a keyword search.

1. On the Documents tab, click the **Search With** drop-down menu, and select **Keyword Search** from the list.

2. Enter terms for the search in the **Search Terms** box. You can click **Clear** to remove search terms from the box.

3. To display fields for search conditions, click **Located** located next to the **Search With** drop-down menu. See **Search conditions on the next page**.

4. Click **Search**. To stop a long running search, click **Cancel Request**.
4 Search conditions

You can use the search conditions option to build complex queries by selecting fields, operators, and values. While this feature has the same functionality as the search condition section of the saved search form, it's conveniently available from the Documents tab and Relativity Dynamic Object tabs. This option displays up to five rows, with each row representing a separate criterion. Depending on the type of field you select, different operators appear. You can use this option alone or in conjunction with keyword searches, dtSearches, Analytics, or Pivot. When you use search options in conjunction with another search feature, documents must both meet the search criteria and also the conditions specified.

Note: To use the search conditions option, you must have add or edit permissions for Search and access to the Saved Searches Browser assigned to you through the Security page. See Workspace security in the Admin Guide.

To set up search conditions, follow these steps:

1. Navigate to the Documents tab or to a Dynamic Object tab.

2. Click to display fields for a search condition. The search conditions icon's appearance changes when conditions are active.

3. Set the following options to define a search condition:
   - **Left or Opening Parentheses** - Use the first box in the row to select a single, double, or triple parentheses for grouping criteria or controlling precedence in the query.
     
     Note: To enable this menu, you must first select a value for Field.
   - **Field** - Select a field available in your workspace.
   - **Operator** - Select a search operator. The field type determines the available operators. See Operators on page 73 Operators in the Searching guide.
   - **Value** - Enter or select a value. The field type determines the available values.
     
     Note: With date-based fields, you can enter @today as the value instead of choosing a date if “is before” or “is after or on” operator is selected.
- **Right or Closing Parentheses** - Use the last box in the row to select a single, double, or triple parentheses to group criteria.

- **AND or OR Operators** - Use these operators to connect the criterion in each row. To remove your current selections for a condition, click the **Clear** link for the row.

4. (Optional) Click **Add another condition** to display another search condition row. Repeat Step 3 to set the options in the new row. You can set a total of five conditions with the search conditions option. If your query requires more than five conditions, create a saved search. See [Creating or editing a saved search on page 68](#) Creating a saved search in the Searching guide.

5. Click **Search**.

6. (Optional) Perform any of the following tasks to work with your search:

   - **Hide Search Conditions** - Click ![Hide Search Conditions](#). A message appears above the item list indicating that the search conditions are active.

   - **Save Search Settings** - Click ![Save Search Settings](#). See [Saving searches on the Documents tab on page 79](#). See the Searching guide.

   - **Remove Search Settings** - Click the **Clear** button to remove the current conditions and any search type settings.

### 4.1 Canceled queries

You can cancel a long-running search or view by clicking the **Cancel Request** link. This link appears when you perform a keyword search, dtSearch, Analytics search, or use Pivot, as well as when you filter or sort a document or Dynamic Object list. It also appears when you perform other actions on item lists containing documents or Dynamic Objects that initiate a query in the background.

When you click **Cancel Request**, Relativity stops the background query used to populate documents in an item list. If you edit a search and click this link, your changes save, but the item list doesn’t load. For example, your changes save when you perform a mass edit on a list of documents in a search and then click **Cancel Request** when the query is running to redisplay the updated list.

**Note:** Relativity creates an audit record in the History tab for canceled queries. The query description displays the running time of the query and indicates that it was canceled. You must have the appropriate permissions to view this tab.

The following table explains different scenarios in which you might cancel a running query and whether the query actually cancels.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>You start a query and click <strong>Cancel Request</strong>.</td>
<td>The query is canceled.</td>
</tr>
<tr>
<td>You start a query and close the browser.</td>
<td>The query is canceled.</td>
</tr>
<tr>
<td>You start a query and an administrator resets the IIS on the server.</td>
<td>The query is not canceled.</td>
</tr>
<tr>
<td>You start a query and leave your browser idle for longer than the session timeout specified in Relativity web.config, regardless of whether you</td>
<td>The query is not canceled.</td>
</tr>
</tbody>
</table>
**4.2 Frequently asked searching questions**

This section includes frequently asked questions from Relativity users.

### 4.2.1 Multiple Terms

*Why would I receive an error message saying my query is too complex?*

Relativity can't return precise results when a query includes search conditions that are too complex due to the use of multiple search terms. To avoid this error message, simplify your search criteria. For example, search for 10 instead of 50 terms in your query. You can save and tag your search results from each simpler query. This approach allows you to search on the required terms, while providing better results.

### 4.2.2 Proximity Searches

*Why are terms in my proximity search highlighted even when they don't match my defined W/N criteria?*

Relativity highlights terms that meet the requirements of your proximity search as well as the individual search terms. This behavior doesn't affect the results of your proximity search, which returns the appropriate documents. For example, the results of a proximity search for instances of law within three words of order (that is "law W/3 order") includes highlighted terms of "law" and "order" when they aren't within three words of each other. To accommodate this behavior, focus only on the results that match the proximity search criteria, and disregard the other highlighted terms.

### 4.2.3 Using ampersands

*Can I use an ampersand (&) in my search queries?*

Overall, searching with the & character is unpredictable and gives inconsistent results. Although the & character is in the search index as a space character it's also reserved as an operator in dtSearch and can skew search results. In order to get the most accurate results when searching with the & character use proximity searching instead. Searching with the & character as a dtSearch operator returns search results as expected.

### 4.2.4 Multiple Conditions

*Why do my searches run slowly when I use multiple conditions?*

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>clicked OK or Cancel on the message from the webpage popup.</td>
<td></td>
</tr>
<tr>
<td>You start a query and click Cancel on the message from the webpage popup within the session timeout specified in Relativity web.config.</td>
<td>The query is canceled.</td>
</tr>
<tr>
<td>You start a query and click OK on the message from webpage popup within the session timeout specified in Relativity web.config.</td>
<td>The query continues to run. Relativity returns you back to the waiting screen (see the first row of this table).</td>
</tr>
<tr>
<td>You start a query and paste a different URL into your browser, or you refresh the page.</td>
<td>The query is canceled.</td>
</tr>
</tbody>
</table>
When you add multiple search conditions to a query, Relativity searches on these conditions relative to each other, which slows down the return of your results. For example, you experience slow performance when running a query on all email messages received "after June 1" and "before June 30" of the same year. You can improve performance by using as few conditions as possible, such as excluding the condition "before June 30". Run the query with only the condition "after June 1", and then sort or filter your results to display messages received between the desired dates.

### 4.2.5 Nesting Searches

**Can I nest multiple searches in a saved search?**

For performance reasons, we don’t recommend nesting multiple searches in a saved search. You can select a search as a condition, but using multiple searches as conditions slows down the return of your results. See [Using saved searches as conditions (combined searches) on page 77](#). See the Searching guide for more information.
5 dtSearch

Relativity’s dtSearch engine provides advanced search functionality such as proximity, stemming, and fuzzy searches. It also supports the use of Boolean operators and custom noise word lists as well as the basic searching features available in keyword searches. After building your dtSearch index, the Dictionary search option is also available.

Using dtSearch indexes

There are roughly three million files relevant to a case you’re working on, including emails, email attachments, invoices, and technical manuals related to construction practices and material handling. It’s early in the case, and you need to gain an understanding of the data set. You also need to retrieve certain text related to five substances that you know are prevalent in this data, as an employee from the construction company mentioned these specifically in an email to you. To do this, you need to be able to perform proximity, stemming, and fuzzyproximity, stemming, and fuzzy searches on your data set. So you create a new dtSearch index.

You call the index "Hazardous Materials dtSearch" so that you can easily identify it in the Search drop-down on the Documents list. You might also create an Analytics index for this case with a similar name, so make sure to differentiate them clearly. For the Searchable set field, you select a saved search that you’ve already created called Hazardous Materials searchable set, which contains documents to which you’ve already applied keywords related to the substances mentioned in your client’s email.

Because many of the invoices and emails in your data set contain references to various purchases of building materials made by various departments in the construction company you’re helping to represent, you set the Auto-recognize date, email and credit card numbers field to Yes.
You leave all other fields at their default settings and save the index. You then build and activate the index so that you can select it in the Search drop-down.

When you select the index and search your document set against it, you run a number of proximity searches to see how close terms relating to hazardous substances occur to the names of the building materials that may or may not contain them. The searches you run include the following:

- lead W/10 paint
- lead W/10 plumbing pipes
- lead W/10 connectors
- lead W/10 solder
- asbestos W/10 insulation and
- asbestos W/10 pipe coverings
- asphalt W/10 sealant
- asphalt W/10 adhesives
- radioactive W/10 fluorescent lamps
- radioactive W/10 smoke detectors

As you keep running these proximity searches, you get down to a small group of intriguing emails between a prospective buyer (your client) and a prospective seller, which may prove that
the seller had knowledge of the fact that those building materials were potentially dangerous when they were negotiating a price with your client. This discovery turns out to be crucial to the case.

5.1 Creating a dtSearch index

You can build custom dtSearch indexes for a subset of documents or for certain document fields in a workspace. You must have the appropriate permissions to complete this task. See Workspace security in the Admin Guide.

Before you begin, you need to create a saved search that includes the fields that you want to include in the index. You can then name the index based on the document search set used to create it.

Note: Within a field, dtSearch truncates any string longer than 32 characters that doesn't contain a space character. It indexes only the first 32 characters of the string. See Using dtSearch syntax options on page 38.

To create a new dtSearch index:

1. Navigate to the Search Indexes tab and click New dtSearch Index. The dtSearch index form appears with required fields in orange.

2. Complete the fields on the dtSearch index form. See Fields on the next page.

3. Click Save to display the index details page. The index details page now displays three additional read-only fields and the dtSearch index console. See Fields on the next page and dtSearch console on page 31.

4. Click Build Index: Full. A dialog window asks you to verify that you want to run a full build. You can also select Activate this index upon completion. Indexes must be active in order to search them.

Note: Click OK to build your index. If a dtSearch manager or worker agent encounters a network-related error during the build process, it will execute up to three retry attempts at 30 second intervals.

5. If you didn't select Activate this index upon completion in the dialog window, click Activate Index on the console. The index won't activate if there are errors. Activating an index makes it available in the Search menu.

6. (Optional) Click Refresh Page at any point in the build to see the index's current build status. If errors occur during the build, the Retry Errors button enables on the console under the Errors and Status heading. Click this button to attempt to resolve any errors.

Once the index is built, the console enables additional options. See dtSearch console on page 31.

5.1.1 Accent-insensitive indexes

Because Relativity uses only accent-insensitive indexes, some characters are translated to the base character, which causes those characters and any terms containing those characters to be deduplicated in a Search Terms Report.
Note: dtSearch uses .ABC files, but only for characters in the range from 33 to 127. All other characters are handled according to the definitions in the Unicode character tables.

Example: accented characters like á or ñ are converted to the unaccented versions, a or n.

Example: If you searching for the term fröhlich, searching that term as fröhlich or frohlich would both return the hit. However, highlighting in the Viewer may not display both variations.

5.2 Fields

The dtSearch index page includes the following fields:

- **Name** - the dtSearch index name. This name appears within the "search with" menu in the Documents tab.
- **Order** - the integer value (positive or negative) representing the position of the index in the search indexes list. Indexes sort from lowest (top) to highest (bottom) order number. Those with the same order number sort alphanumerically.
- **Searchable set** - the set of documents to be indexed. You can select all documents in the workspace or choose from any saved search in the workspace. If you select:
  - `<all documents in workspace>` - Relativity indexes the fields with Include in Text Index set to yes.
A saved search - Relativity indexes the documents returned by the search as well as the returned documents' fields. It may use a dtSearch or Relativity Analytics index. Make sure the index is active.

**Note:** When creating a dtSearch index, it's best practice to only index the fields you want to search. Move all other fields to the Fields (Required) left column. Typically, you only index the extracted text field if you're searching the body of emails.

- **Index share** - populated by default by a system admin.

- **Auto recognize date, email, and credit card numbers** - a yes/no field. See *Auto-recognition on page 40* for details.

- **Send Email Notification upon Completion or Failure to** - allows you to send email notifications when your index population fails or completes. Enter the email address(es) of the recipient(s). Separate multiple entries with a semicolon.

- **Sub-index size** - determines the size of each sub-index created when you generate a dtSearch index. The minimum value is 1000.

  **Note:** To set a new default for this field, an administrator can edit the dtSearchDefaultSubIndexSize configuration value. See the Configuration table guide.

- **Sub-index fragmentation threshold** - determines the fragmentation level at which the system automatically compresses a dtSearch sub-index during an incremental build. An incremental build automatically compresses any sub-index equal to or greater than the fragmentation threshold. The Sub-index fragmentation threshold value must be equal to or greater than one.

  **Note:** The dtSearchDefaultSubIndexFragmentationThreshold configuration table value determines the default Sub-index fragmentation threshold. It is set to 9 by default.

- **Noise Words** - allows you to edit the list of words that are ignored during indexing.

- **Alphabet** - allows you to edit the index’s alphabet file. See *Making a character searchable on page 50*.

Index status fields:

- **Active** - indicates whether the index is active (Yes) or inactive (No).

  **Note:** File type fields, linked fields, and HTML enabled fields may have text associated with them that is not visible in your document views. This includes the system FileIcon field, which is populated with the original file name upon import. See the System Fields section in the Admin guide.

### 5.3 dtSearch console

The dtSearch index console includes the following options:
- **Build Index: Full** - creates a full build of the index. During the build, the button toggles to Cancel Build.

  **Note:** Canceling the build aborts the indexing thread, leaving the index in an unstable state. Relativity deletes these indexes from the population table and gives them an inactive status. You can't search against an index with an inactive status until you run a full build. Canceling also deletes the index files in the index share.

- **Build Index: Incremental** - allows you to update an index after adding or removing documents. During an incremental build the existing index remains available for searching, but changes to the index are not reflected in search results until the incremental build is complete. Canceling an incremental build returns the index to its previous state.

  **Note:** The incremental build process copies each sub-index that requires modification, updates the copy, then replaces existing sub-indexes with the updated copies. Duplicate sub-indexes are removed when the Case manager agent runs. The system automatically compresses a sub-index during an incremental build only if the sub-index fragmentation level is equal to or above the Sub-index fragmentation threshold value. The incremental build process doesn't update the index for documents currently in the index with modified text.

- **Compress Index** - compresses the dtSearch index returning all sub-indexes with a fragmentation level greater than zero to a fragmentation level of zero. You can search against the original (uncompressed) dtSearch index while compression is in progress. Once compression is complete, the system automatically replaces the old sub-indexes with the defragmented sub-indexes. Duplicate sub-indexes are removed when the Case manager agent runs.
**Note:** The Compress Index button only runs compression against sub-indexes that have a fragmentation level greater than zero. Canceling compression returns the index to its original fragmented state before compression began.

- **Deactivate Index** - deactivates the index and removes it from the "search with" menu in the Documents tab (but not from the database).

- **Swap Index** - allows you to swap your index with a replacement index in order to use its resources while your index builds or is inactive or disabled for any reason. This enables you to keep searching while your primary index experiences downtime. You can only select indexes in the Replacement Index with an Active status. This index you swap to doesn’t automatically run an incremental update. Selecting the index from the drop-down and clicking OK completes the index swap. You can't reverse the swap results in the current dialog box. You must close this swap and run again to swap back or swap another time. This functionality is useful in very limited cases for example, if you are doing a full rebuild on a very large index. Since dtSearch incremental builds are online, someone can search documents already indexed.

**Note:** The Swap Index function updates anything in the Views table, which affects batches, saved searches, nested searches, etc.

- **Retry Errors** - enables only if errors occur, this button allows you to resolve errors.

- **Show Document Errors** - enables only if document errors occur. This button creates an exportable list of document-level errors.

- **Show Detailed Status** - shows you statistical data for the index, including:
  - Doc Count - the total number of documents in the index
  - Index Size - the size of the index in bytes
  - Created Date - the date you created the index
  - Updated Date - the date you updated the index
  - Last Build Duration - how long the last build took to complete in hours, minutes, and seconds

- **Refresh Page** - allows you to see the index's current build status.

### 5.4 dtSearch index page

After you create and build a dtSearch index, the dtSearch page contains several sections where you can view details about your index.

#### 5.4.1 Index Status

The Index Status section of the layout is where you can view what state your dtSearch index is in. The name of the Index Status section is populated with the name of your dtSearch index. When you’re building an index, this section changes to a progress bar where you can track your index’s progress in real-time. When the index is no longer in progress, this section changes to a static field that displays the below fields.
- **Status** - the status of the index. For example, "Active - Indexed" or "Inactive - Indexed".
- **Document Breakdown** - the number of indexed documents.

### 5.4.2 dtSearch Index Information

The dtSearch Index Information section provides general details about the settings applied to your dtSearch index. This section contains the following information:

- **Name** - the name of your index.
- **Order** - the integer value (positive or negative) representing the position of the index in the search indexes list. Indexes sort from lowest (top) to highest (bottom) order number. Those with the same order number sort alphanumerically.
- **Searchable set** - the set of documents to be indexed. You can select all documents in the workspace or choose from any saved search in the workspace.
- **Index share** - populated by default by a system admin.
- **Auto-recognize date, email, and credit card numbers** - a yes/no field.
- **Email notification recipients** - the emails that receive an email notification when your index population fails or completes.

### 5.4.3 Advanced Settings

The Advanced Settings section provides sub-index details about your dtSearch index. This section contains the following information:

- **Sub-index size** - determines the size of each sub-index created when you generate a dtSearch index. The minimum value is 1000.
- **Sub-index fragmentation threshold** - the fragmentation level at which the system automatically compresses a dtSearch sub-index during an incremental build.
- **Sub-indexes scheduled for compression** - the number of sub-indexes at or above the sub-index fragmentation threshold. If one or more sub-indexes is equal to or greater than the sub-index fragmentation level, the system automatically compresses those sub-indexes during the next incremental build.

### 5.4.4 Temporary Index Details

The Temporary Index Details section only appears during an incremental build. This table displays sub-indexes that were copied from your original index and are in the process of modification during the incremental build. Once the sub-indexes in this table are updated, they replace the original sub-indexes from which they were copied. This section contains the following information:

- **Population Table** - the name of the table that a sub-index is populating.
- **Build Status** - the state that the sub-index is currently in.
- **Worker Agent** - the name of the agent that's handling the sub-index.
- **Worker Agent Status** - the state that the worker agent is currently in.
- **Index File Share** - the location where your sub-index is stored.
- **Document count** - the number of documents assigned to the sub-index.
- **Error(s)** - any errors encountered by the sub-index.
- **Fragmentation Level** - the fragmentation level of the sub-index. Any index at or above the Sub-index fragmentation threshold appears in red.

### 5.4.5 Current Index Details

The Current Index Details section displays the sub-indexes that make up your dtSearch index. This section contains the following information:

- **Population Table** - the name of the table that a sub-index is populating.
- **Build Status** - the state that the sub-index is currently in.
- **Worker Agent** - the name of the agent that's handling the sub-index.
- **Worker Agent Status** - the state that the worker agent is currently in.
- **Index File Share** - the location that your sub-index is stored.
- **Document count** - the number of documents assigned to the sub-index.
- **Error(s)** - any errors encountered by the sub-index.
- **Fragmentation Level** - the fragmentation level of the sub-index. Any index at or above the Sub-index fragmentation threshold appears in red.

### 5.5 Temporary storage

If you specify a temporary storage location, dtSearch builds the index in this directory and then copies the index over to the final index share when the build completes. Using a temporary storage location could potentially speed up the build time and reduce network contention. See the Admin guide for more information on Servers.

### 5.6 Running a dtSearch

To run a dtSearch on in the Documents tab in Relativity:

1. Select a dtSearch search index from the "search with" menu. The available dtSearches vary by workspace.

2. (Optional) Click **Dictionary** to determine the total occurrences of a term in the dtSearch index. See [Running a Dictionary search on page 37](#).
3. Enter terms for the search in the Search Terms box. You can click **Clear** to remove the current search terms from the box.

**Note:** Your total number of search terms cannot exceed 65,000 characters. Additionally, words longer than 32 characters are truncated during indexing.

The following table shows search string examples and their expected results. For more information, see **Using dtSearch syntax options on page 38**.

<table>
<thead>
<tr>
<th>Search String</th>
<th>Returns Documents With...</th>
</tr>
</thead>
<tbody>
<tr>
<td>apple pear</td>
<td>The exact phrase <strong>apple pear</strong></td>
</tr>
<tr>
<td>&quot;apple pear&quot;</td>
<td>The exact phrase <strong>apple pear</strong></td>
</tr>
<tr>
<td>apple AND pear</td>
<td>The word <strong>apple</strong> and the word <strong>pear</strong></td>
</tr>
<tr>
<td>(apple and pear) AndAny (grape or banana)</td>
<td>Any document that contains <strong>apple</strong> and <strong>pear</strong>, with <strong>grape</strong> and <strong>banana</strong> also being counted as hits. <strong>Grape</strong> and <strong>banana</strong> aren't evaluated as conditions; they're simply added to the search results as optional terms to be highlighted if they appear in the document.</td>
</tr>
<tr>
<td>apple OR pear</td>
<td>Either <strong>apple</strong> or <strong>pear</strong></td>
</tr>
<tr>
<td>apple W/5 pear</td>
<td><strong>Apple</strong> appears within 5 words of <strong>pear</strong></td>
</tr>
</tbody>
</table>
| apple PRE/5 pear | **Apple** appears within 5 words before **pear**  
**Note:** Relativity does not use the POST operator. However, you can mimic this functionality by reversing the order of the terms, and using the PRE operator. |
| apple NOT W/5 pear | **Apple** occurs not within 5 words of **pear** |
| apple AND NOT pear | **Apple** appears but **pear** does not |
| apple W/5 xfirstword | **Apple** appears in the first 6 words of the document |
| apple w/5 xlastword | **Apple** appears in the last 6 words of the document |

4. (Optional) To display fields for search conditions, click the search conditions icon next to the "search with" menu. See **Search conditions on page 23**.

5. Click **Search**. To stop a long running search, click **Cancel Request**.

**Note:** The Sort by Rank option doesn’t have any functionality in dtSearch. All documents returned have a rank of 100%. To learn more about the Rank column, see Running a concept search in the viewer in the Analytics guide.
5.7 Running a Dictionary search

When you run a dtSearch, you can use Dictionary Search to query the index for a term and find the total occurrences and number of documents in which it occurs. The Dictionary Search dialog displays the results, including the following columns:

- **Keyword** - the word contained in the index. If you use fuzziness or stemming in your search, this column displays any variations of the term.
- **Total Words** - the number of times the word appears in the dtSearch index.
- **Total Documents** - the number of documents in the index that contain the word.

A dictionary search returns only the first 2,000 items in the result set. If your search returns more than that, a message displays to indicate that only 2,000 items were returned.

To run a dictionary search:

1. In the "search with" menu, select a dtSearch index. See Running a dtSearch on page 35.

2. Click Dictionary to determine the total occurrences of a term in the dtSearch index.

3. Enter a search term in the textbox. You can enter variations of a search term as well as the wildcard (*), stemming (~), or fuzzy searching (%) operators.

4. (Optional) Set one of the following search operators:
   - **Fuzziness Level** - Select a value from one to ten to set the degree of variation in the terms returned. A larger number returns terms with more variation. This option is independent of the fuzziness (%) character that you can enter in the textbox on the Dictionary Search dialog. See Fuzzy searching on page 47.
   - **Enable Stemming** - Select to return grammatical variations of a word. For example, a search on apply will return applying, applies and applied. This option is independent of the stemming (~) character that you can enter in the textbox on the Dictionary Search dialog. See Stemming on page 51.
5. Click **Search** to display a list of keywords and the associated totals.

![Dictionary Search](image)

6. Perform any of the following tasks with your search results:
   - **Copy to Clipboard** - Select the checkboxes for the terms that you want to copy, and click **Copy to Clipboard**. On the Documents tab, right-click in the **Search Terms** box, and click **Paste**. The terms will added to the textbox separated by the OR operator.
   - **Sort** - Click a column header to sort in descending or ascending order.
   - **Reset Columns Sizes** - Displays columns with their default widths.
   - **Export to Excel** - Click to download an Excel spreadsheet containing the keywords and totals from the dictionary search.
   - **Show Filters/Clear All** - See [Filters on page 8](#).

### 5.8 Using dtSearch syntax options

dtSearch includes special characters and other operators that you can use to define search criteria. The following table summarizes the syntax options available for queries run against a dtSearch index.

<table>
<thead>
<tr>
<th>Special characters or operators</th>
<th>Search functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>date(), mail(), creditcard()</td>
<td>Auto-recognition on page 40</td>
</tr>
</tbody>
</table>
5.8.1 Exact phrase

Searching for words right next to each other with no operator between them constitutes an exact phrase in dtSearch. For example, if you search for *apple pear*, Relativity returns documents that contain the exact phrase *apple pear*. There is no rule that requires quotation marks around a phrase of any number of words. However, you must use quotation marks for exact phrases that are Boolean expressions when a phrase contains a connector. For example:

*clear and present danger*

"*clear and present danger*"

In the first case, dtSearch searches for documents that contain the word *clear and present danger*. In the second case, dtSearch searches for the phrase *clear and present danger*.

The following graphic depicts what documents are returned when you execute an exact phrase search:
5.8.2 Auto-recognition

Auto-recognition provides you with the ability to search for various date formats, e-mail addresses, and credit card numbers. However, it can dramatically affect indexing and searching performance. You must activate auto-recognition before you can use it in your workspace. Contact your administrator for more information.

5.8.2.1 Date recognition

Date recognition searches for strings that appear to be dates. It uses English-language months, including common abbreviations, and numerical formats. For example, these date formats are recognized:

- January 15, 2006
- 15 Jan 06
- 2006/01/15
- 1/15/06
- 1-15-06
- The fifteenth of January, two thousand six

Note the following date and date range search strings:

- To search for a date, enter a date expression between the parentheses in the string date(); for example, date(jan 10 2006)
- To search for range of dates, enter a date range between the parentheses in the string date(); for example, date(jan 10 2006 to jan 20 2006)
To search for a range of dates near the word *apple*, enter *date(jan 10 2006 to jan 20 2006) w/10 apple*

Unterminated date ranges aren't supported. To search for any date after or before a particular date, enter a bounded range with a maximal or minimal value for the bounds. The maximum value for a year is 2900, and the minimum value is 1000. For example, *date(jan 10 2006 to jan 1 2900)*

dtSearch recognizes numeric strings as dates, as long as it can be interpreted as a valid date. This includes formats common in the US and UK, including:

- MM/DD/YY or MM-DD-YY
- MM/DD/YYYY or MM-DD-YYYY
- DD/MM/YY or DD-MM-YY
- DD/MM/YYYY or DD-MM-YYYY

In the case of ambiguous dates, such as 01/05/10, dtSearch defaults to MM/DD/YY.

If the date contains words the dtSearch converts the words to a numeric value to help interpret the date. For example, 30 must be a day and not a month, and 2015 must be a year (not a day or month).

### 5.8.2.2 Email address recognition

Email address recognition searches for text with the syntax of a valid email address, such as *sales@example.com*. With this feature, you can search for a specific email address regardless of the alphabet settings for "@", ".", or other punctuation in the email address. You can also use the word listing functions in dtSearch to enumerate all email addresses in a document collection. You must include either the * or ? wildcard expression to enumerate all email addresses in a document collection.

- mail(sales@example.com) - returns the exact email address: sales@example.com
- mail(sa*@example.com) - returns variations of the email address: sal@example.com; sales.sa@example.com

### 5.8.2.3 Credit card number recognition

Credit card number recognition searches for any sequence of numbers that matches the syntax for a valid credit card number issued by a major company, such as Visa, MasterCard, and so on. A credit card number is recognized regardless of the pattern of spaces or punctuation embedded in the number:

- 1234-5678-1234-5678
- 1234567812345678
- 1234 5678 1234 5678

Credit card issuers use numerical tests to exclude sequences of numbers that aren't valid credit card numbers. Since these tests don't detect all invalid numbers, the feature for credit card number recognition may find additional invalid numbers.

To search for a credit card number, enter a credit card number between the parentheses in *creditcard()* as exemplified in *creditcard(1234)*.
5.8.3 Boolean operators
The dtSearch engine supports Boolean operators, including AND, OR, and NOT. You can use these operators to connect multiple phrases or terms in a single search expression.

Note: When using Boolean operators in a proximity search, noise words are included. Although the noise words are not searchable, they are still counted in the proximity search.

5.8.3.1 AND operator
When you use the AND operator to connect expressions, only documents that contain all the expressions in the search string will be returned in the result set. The following search strings illustrate how to use this operator:

- *apple pie AND poached pear* retrieves any documents that contain both phrases.
- *(apple or banana) AND (pear w/5 grape)* retrieves any documents that contain *apple* or *banana* AND contain *pear* within five words of *grape*.

The following graphic depicts what documents are returned when you use the AND operator in a dtSearch string:

5.8.3.2 AndAny operator
You can combine a search for required search terms with other terms that are optional. The words before the AndAny connector are required, and the words after the AndAny connector are optional. For example, *(apple and pear) AndAny (grape or banana)* would find any document that contains *apple* and *pear*, with *grape* and *banana* also being counted as hits only if *apple* and *pear* are also present in the document.

A document is only returned if it contains at least the required search terms. If the document contains any or all of the terms that come after the AndAny connector but none of the terms that come before the
AndAny connector, it will not be returned. In other words, the AndAny operator doesn’t change the set of docs responsive for any optional terms coming to the right of the AndAny operator. The AndAny terms are not evaluated as conditions, but are simply added to the search results.

The following example further explains the AndAny operator:

You have three documents, each containing the term(s) specified below:

- Document 1: Apple
- Document 2: Apple, Grape, Pear
- Document 3: Grape, Pear

Note the following behavior:

- When you search for the term apple, documents 1 and 2 are returned.
- When you search for the string apple AND pear, only document 2 is returned.
- When you search for the string apple AndAny pear, documents 1 and 2 are returned.

The following graphic depicts what documents are returned when you use the AndAny operator in a dtSearch string:

---

5.8.3.3 OR operator

When you use the OR operator to connect expressions in a search string, documents that contain one or more of these expressions will be returned in the result set. For example, the search string apple pie or poached pear will return documents that contain apple pie, poached pear, or both phrases.

The following graphic depicts what documents are returned when you use the OR operator in a dtSearch string:
5.8.3.4 NOT operator

In a dtSearch, you can use the NOT operator at the beginning of a search expression to negate its meaning and exclude documents from a result set. For example, the search expression *applesauce and NOT pear* returns documents that contain the word *applesauce*, but not those documents that contain both the words *applesauce* and *pear*.

- **NOT operator as a standalone** - You can use the NOT operator by itself at the beginning of a search expression. For example, the search expression *NOT pear* returns all the documents that do not contain the word *pear*. The search expression *NOT (apple w/5 pear)* returns all the documents that do not contain the word *apple* within five words of *pear*.

- **NOT operator as a connector** - When the NOT operator appears in the middle of a search expression, it must be used in conjunction with either AND or OR. For example, the search expression *apple OR NOT pear* returns all the documents that contain the word *apple* and those that do not contain the word *pear*.

**Note:** You can also use NOT in a proximity search as illustrated by the NOT W/N (NOT Within N words) operator.

- **AND NOT operator** - You can use the AND NOT operator to develop queries for documents that include the first expression but not the second expression. For example, you may want to query for email messages that have Ryan as the author, but do not have Will as the recipient. The following record illustrates these conditions:

<table>
<thead>
<tr>
<th>Document</th>
<th>OCR</th>
<th>Recipient</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS00001</td>
<td>From: Ryan To: Will</td>
<td>Will</td>
<td>Ryan</td>
</tr>
</tbody>
</table>

You can perform a dtSearch using the search string "Ryan AND NOT Will" and return results that don't include document AS00001.

The dtSearch engine combines into a single pool the text for all fields identified for inclusion in an index. A search string using the AND NOT operator will query the index that includes the combine text from all indexed fields, rather than querying the content of individual fields. This behavior ensures consistent result sets when querying with the AND NOT operator.
Note: A keyword search is an SQL full text search, which queries individual fields. Keyword search won’t return the same results as dtSearch when the NOT operator is used to query across multiple fields. See NOT operator on the previous page.

The following graphic depicts what documents are returned when you use the AND NOT operator in a dtSearch string:

5.8.3.5 Operator precedence
The precedence, or order of evaluation, determines how a group of expressions is evaluated in a query. You can use parentheses to group expressions and control which ones are evaluated first. For example, the following search strings illustrate how to group expressions:

- (apple and pear) or grape
- apple and (pear or grape)

When a search string doesn’t include parentheses, a dtSearch evaluates OR expressions and then AND expressions. For example, the search string A AND B OR C is evaluated as follows for a dtSearch: A AND (B OR C).

Note: The order of precedence for a keyword search evaluates AND expressions and then OR expressions: (A AND B) OR C.

When two expressions are connected by W/N, at least one of them must be a single word or phrase, or a group of words and phrases connected by OR:

- (apple and banana) W/10 (pear or grape)
- (apple and banana) W/10 orange tree
5.8.4 Built-in search words

dtSearch includes the following built-in search words:

You can use these terms to limit a search to the beginning or end of a file. For example, *apple W/10 xlastword* searches for *apple* within 11 words of the end of a document.

- **xfirstword** - Marks the beginning of a file.

The following graphic depicts what documents are returned when you use the W/# xfirstword operator in a dtSearch string:

- **xlastword** - Marks the end of a file.

The following graphic depicts what documents are returned when you use the W/# xlastword operator in a dtSearch string:
5.8.5 Fuzzy searching

Using the dtSearch engine, you can perform fuzzy searches, which return documents containing spelling variations of a specified term. You may want to use fuzzy searching when querying documents that contain misspelled terms, typographical errors, or have been scanned with Optical Character Recognition (OCR).

The percent sign (%) is the character used for fuzzy searches. The number of (%) used indicates how many characters in the search term will be ignored by the dtSearch engine when it runs the query. The position of the % indicates the number of characters from the beginning of the term that must match exactly with words in the result set. The following search strings illustrate how this character is used:

- **app%ly** indicates that a matching word must begin with **app** and differ from **apply** by only one character.
- **a%%ply** indicates that a matching word must begin with **a** and differ from **apply** by only two characters.

The following graphic depicts what documents are returned when you use fuzzy searching in dtSearch:

For a description of using fuzzy searching and stemming together, see [Using fuzzy searching and stemming together](#).

5.8.5.1 Using the fuzziness operator and fuzziness level option

In Relativity, you have can use the fuzziness character (%) or the Fuzziness Level menu to perform fuzzy searches. The availability of these search options depends on the location where you are running a dtSearch:

- **Documents tab** - When you select a dtSearch in the Search With option, you can use the fuzziness character (%). See [Running a dtSearch on page 35](#).
- **Dictionary Search** - When you click the Dictionary link, you can use the fuzziness character (%) and the **Fuzziness Level** menu on the Dictionary Search dialog. See [Running a Dictionary search on page 37](#).

  In the **Fuzziness Level** menu, you can select a value from 1 to 10, which applies to all terms in the textbox. Larger numbers return terms with more variation. We recommend using values between 1 and 3 for moderate error tolerance. The following table describes the expected results for sample settings.

<table>
<thead>
<tr>
<th>Fuzziness level</th>
<th>Description of search results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blank</td>
<td>Only returns the entered term.</td>
</tr>
<tr>
<td>1</td>
<td>Returns slight variations of the entered term.</td>
</tr>
<tr>
<td>4</td>
<td>Returns multiple variations of the entered term.</td>
</tr>
</tbody>
</table>

- **Saved Search** - When you create a saved search, you can use the fuzziness operator (%) and the **Fuzziness Level** menu in the Search Conditions section of the form. The **Fuzziness Level** menu in a saved search uses the same settings as described above. See [Saved search on page 61](#).

  **Note:** The **Fuzziness Level** menu is independent of the fuzziness (%) character that you can enter in the textbox. A search for `appl%` without a **Fuzziness Level** setting may return documents containing `apple` or `apply`, since these terms have the stem `appl` and differ by one character.

Fuzzy searching uses term length and fuzziness level to decide how many % characters to add. This is not a straight level to character match. This means a level 7 fuzziness search doesn't necessarily mean up to 7 additional characters return.

### 5.8.6 Noise words and the alphabet file

The dtSearch engine references a default list of noise words and an alphabet file when it creates a new index. The noise words are excluded in a dtSearch index to improve query performance and prevent unnecessary index growth. These commonly used words are ignored when you run a query. The alphabet file determines how single characters and spaces are handled in a query.

**Note:** If your dtSearches aren't returning the expected results, you may want to ask your administrator about updating the noise word list or alphabet file.

#### 5.8.6.1 Alphabet file

The dtSearch engine uses an alphabet file to define which characters are treated as text, cause word breaks, and are ignored. Administrators can modify the default alphabet file when they create or edit a dtSearch index. See [Making a character searchable on page 50](#).

The alphabet file determines which characters are treated as text, which cause spaces, which cause word breaks, and which are ignored. The categories of items in the alphabet file include:

- **Letters** - all searchable characters, which should include all alphabet characters (a-z and A-Z) and all digits (0-9).
- **Hyphens** - all characters removed during index creation. For example *First-Level* becomes two separate words in a dtSearch index.
- **Spaces** - characters that cause a word break. For example, if you classify the period as a space character, then dtSearch would process U.S.A. as three separate words: U, S, and A.
- **Ignore** - characters that are disregarded in processing text. For example, if you classify the period as ignore instead of space, then dtSearch would process U.S.A. as one word, USA.

**Note:** The underscore (\_) isn't recognized as a space by default. Verify that a given character is defined as causing a word break before using it as a space in a dtSearch.

### 5.8.6.2 Default noise word list

The dtSearch engine is configured with the default noise words listed in the following table. Administrators can modify this list when they create or edit a dtSearch index.

<table>
<thead>
<tr>
<th>Begins with...</th>
<th>Noise words</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>a, about, after, all, also, an, another, any, are, as, and, at</td>
</tr>
<tr>
<td>B</td>
<td>be, because, been, before, being, between, but, both, by</td>
</tr>
<tr>
<td>C</td>
<td>came, can, come, could</td>
</tr>
<tr>
<td>D</td>
<td>did, do</td>
</tr>
<tr>
<td>E</td>
<td>each, even</td>
</tr>
<tr>
<td>F</td>
<td>for, from, further, furthermore</td>
</tr>
<tr>
<td>G</td>
<td>get, got</td>
</tr>
<tr>
<td>H</td>
<td>has, had, he, have, her, here, him, himself, his, how, hi, however</td>
</tr>
<tr>
<td>I</td>
<td>i, if, in, into, is, it, its, indeed</td>
</tr>
<tr>
<td>J</td>
<td>just</td>
</tr>
<tr>
<td>L</td>
<td>like</td>
</tr>
<tr>
<td>M</td>
<td>made, many, me, might, more, moreover, most, much, must, my</td>
</tr>
<tr>
<td>N</td>
<td>never, not, now</td>
</tr>
<tr>
<td>O</td>
<td>of, on, only, other, our, out, or, over</td>
</tr>
<tr>
<td>S</td>
<td>said, same, see, should, since, she, some, still, such</td>
</tr>
<tr>
<td>T</td>
<td>take, than, that, the, their, them, then, there, these, therefore, they, this, those, through, to, too, thus</td>
</tr>
<tr>
<td>U</td>
<td>under, up</td>
</tr>
<tr>
<td>V</td>
<td>very</td>
</tr>
<tr>
<td>W</td>
<td>was, way, we, well, were, what, when, where, which, while, who, will, with, would</td>
</tr>
<tr>
<td>Y</td>
<td>you, your</td>
</tr>
</tbody>
</table>

The following graphic depicts what documents are returned when you include noise words in a dtSearch string:
5.8.6.3 Making a character searchable

**Note:** You can make special characters in a dtSearch index. However, some characters need to be escaped using regular expressions. For more information, see the Regular expression searching - symbols recipe on the Relativity documentation site.

1. Navigate to the dtSearch index.
2. Click **Edit** and scroll down to the Alphabet section.
3. Delete the character from the current category ("hyphen", "spaces", etc). Don’t delete the category heading.
4. Scroll to the top of the Alphabet section and scroll down to the first set of Z's.
5. Create a new line after the Z's and enter the character you want to make searchable four times, separated by spaces. You must also begin with a space.
6. Scroll down in the file to the next line of Z's.
7. Repeat step 5 and save.
8. Perform a full build on the dtSearch index. The characters you added will now be included in your searches.

5.8.7 Phonic searching

Using the dtSearch engine, you can perform phonic searching, which returns documents containing words that sound like the word you're searching for and begins with the same letter. The pound sign (#) is the character used for phonic searches when added to the front of a word. For example, a phonic search for *pear* also finds *pair* and *pare*.

You can also use phonic searching in Dictionary searches.

5.8.8 Stemming

Using the dtSearch engine, you can perform stemming searches, which return documents containing grammatical variations of a root word. The tilde (~) is the character used for stemming searches when added at the end of the root word. For example, a search on *apply*~ returns documents containing the words *apply*, *applying*, *applies*, and *applied*.

Because stemming only works with the root word, it generally doesn't return irregular variations of a verb. For example, a search on *run*~ would not return *ran*. The dtSearch engine only supports stemming for the English language.

After you perform a stemming search, grammatical variations of the root word aren't highlighted in the document result set. For example, the words *applied* or *applying* aren't automatically highlighted in the viewer. You can enter *applied* in the Find Next box, and click the Find Next icon to locate hits or grammatical variations.

5.8.8.1 Using the stemming operator and enable stemming checkbox

In Relativity, you have can use the stemming character (~) or the Enable Stemming checkbox to perform stemming searches. The availability of these search options depends where you're running a dtSearch:

- **Documents tab** - When you select a dtSearch in the Search With option, you can use the stemming character (~). See Running a dtSearch on page 35.

- **Dictionary Search** - When you click the Dictionary link, you can use the stemming character (~) and the Enable Stemming checkbox on the Dictionary Search dialog. See Running a Dictionary search on page 37.
**Saved Search** - When you create a saved search, you can use the stemming character (~) and the Enable Stemming checkbox in the Search Conditions section of the form. See Saved search on page 61.

The Enable Stemming checkbox is independent of the stemming (~) character that you can enter in the Search Terms box or Dictionary Search textbox. A search for apply~ with Enable Stemming checkbox unselected will return apply, applied, applies, or applying. A search for apply with Enable Stemming checkbox selected will return the same results.

The following graphic depicts what documents are returned when you use stemming in your dtSearch string:

In this example, arthroscopic doesn't return because arthroscopic doesn't stem from art in English, rather arthro.

**5.8.8.2 Using fuzzy searching and stemming together**

With fuzzy searching and stemming enabled, it will check for a fuzzy match twice, once against the original term, and once comparing the stemmed word with the stemmed word in the index. A match on either will count as a hit.

**5.8.9 Wildcards**

The dtSearch engine supports special characters that you can use as wildcards. It also supports the use of leading wildcards, or those added to the beginning of a word. The following characters represent wildcards in dtSearches:

<table>
<thead>
<tr>
<th>Special character</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Matches any single character. <strong>Note:</strong> If this character is added to the end of a term it limits the search to the number of characters in the term being searched. For example, searching &quot;receivable?&quot; does not return the root term.</td>
</tr>
<tr>
<td>*</td>
<td>Matches any number of characters. <strong>Note:</strong> This character slows searches when used near the beginning or middle of a word.</td>
</tr>
</tbody>
</table>
As illustrated in the following table, you can add wildcards to the root of any word to return matching terms from a dtSearch.

<table>
<thead>
<tr>
<th>Sample search string</th>
<th>Description of search results</th>
</tr>
</thead>
<tbody>
<tr>
<td>appl*</td>
<td>Matches apple, application, and so on.</td>
</tr>
<tr>
<td><em>cipl</em></td>
<td>Matches principle, participle, and so on.</td>
</tr>
<tr>
<td>appl?</td>
<td>Matches apply and apple, but not apples.</td>
</tr>
<tr>
<td>ap*ed</td>
<td>Matches applied, approved, and so on.</td>
</tr>
<tr>
<td>apply~</td>
<td>Matches apply, applied, applies, and so on.</td>
</tr>
</tbody>
</table>

The following graphic depicts what documents are returned when you use the * wildcard in a dtSearch string:

The following graphic depicts what documents are returned when you use the ? wildcard in a dtSearch string:
5.8.10 W/N operator

You can use the W/N (within N words) operator to return documents with two words or phrases occur within a certain proximity of each other. When using Boolean operators in a proximity search with the W/N operator, noise words are included. The N value represents the number of intervening words. For example, the search expression apple W/5 pear returns documents that contain apple only when it occurs within five words of pear. The documents returned by the search must contain the terms within the required proximity, such as five words.

The W/N operator is symmetrical. The search expression apple W/5 pear returns the exact same document as pear W/5 apple.

**Note:** Single characters are treated as full words when this operator is used. For instance, if you search for Harry W/2 Truman, your search retrieves documents that include Harry S Truman or Harry S. Truman.

**Note:** The W/N operator can be interchanged with WI (or wi). For example, the search expression apple W/5 pear returns the same results as apple WI5 pear.

The following graphic depicts what documents are returned when you use the W/# operator in a dtSearch string:
5.8.10.1 NOT W/N

You can use the NOT W/N (not within N words) operator to exclude documents from a result set when two words or phrases are within a certain proximity of each other.

For example, the search expression *apple NOT W/20 pear* returns documents that contain *apple* when it's separated from *pear* by at least 20 words; it also returns documents that don't contain *pear*. Documents that contain *apple* separated from *pear* by fewer than 20 words aren't returned.

The NOT W/N isn't symmetrical. The search expression *apple NOT W/20 pear* doesn't return the same documents as *pear NOT W/20 apple*.

The following graphic depicts what documents are returned when you use the NOT W/# operator in a dtSearch string:
5.8.10.2 Complex expressions

You can create complex expressions with the W/N operator by connecting words or phrases. At least one of these expressions must be a single word, phrase, or group of words and phrases connected by an OR operator as illustrated by the following:

- (apple AND banana) W/10 (pear OR grape)
- (apple AND banana) W/10 (orange tree)

Avoid creating complex expressions that produce ambiguous results as illustrated in the following examples:

- (apple AND banana) W/10 (pear AND grape)
- (apple w/10 banana) w/10 (pear and grape)

**Note:** dtSearch displays a warning message when you enter an ambiguous search request.

5.8.10.3 PRE

You can use the PRE operator to search for a word that appears within a certain number of words before another word.

For example, the search string *apple PRE/5 pear* returns documents where *apple* appears within 5 words before *pear*.

**Note:** Relativity does not use the POST operator. However, you can mimic this functionality by reversing the order of the terms, and using the PRE operator.

The following graphic depicts what documents are returned when you use the PRE/# operator in a dtSearch string:
5.8.11 Numerical patterns

To search for other numerical patterns such as social security numbers, you can use the = wildcard, which matches any single digit. For example, if hyphens are indexed as spaces, then the following search request would find U.S. social security numbers:

```=== == ===```

This searching pattern can return false hits; for example, no valid social security number begins with 9. However, this is the only way to get social security numbers with spaces instead of dashes.

**Note:** dtSearch support notes the `=== == ===` notation is actually more performant than a regular expression for the same pattern assuming you're ok with getting some false hits.

5.8.12 Connector words

The dtSearch connector words are:

- and
- or
- not
- to
- contains

To search for a phrase that contains one of the dtSearch connector words, quote a connector word or the phrase it is in, or put a tilde after the connector. The following search strings work in returning phrases that contain connector words:
- "clear and convincing evidence"
- not~ relevant
- "whether or not John wants to"

Note the following:

- Adding a ~ after a connector word prevents dtSearch from recognizing the word as a connector but does not otherwise affect the search. The ~ character after a word tells dtSearch to apply the stemming rules to it. Because the stemming rules included with dtSearch do not modify short words, the ~ does not change the outcome of a search for and, or, not, or to.

- Connector words such as "and" and "not" are also included by default in the noise word list. All these words are Noise words and you must remove these words from the list to make dtSearch index these files. See Creating a dtSearch index on page 29 for details.

5.8.13 Words and phrases

With a dtSearch, you can use quotation marks to search for a phrase. For example, the phrase fruit salad is included in the search string apple w/5 fruit salad. The following list outlines how dtSearch queries on words or phrases with noise words or punctuation:

- **Phrases with Noise Words** - dtSearch skips any noise words in a phrase. For example, it skips of in the search string Statue of Liberty and retrieves any documents that contains statue, an intervening word, and liberty.

- **Words with Punctuation** - Punctuation inside a word is treated as a space. For example, dtSearch treats the search term can’t as two words, can and t.

- **Numbers and Characters in Parenthesis** - Unexpected results may be returned when numbers or characters in parenthesis are used in a dtSearch. For example, the search term 1843 (c)(8)(ii) is treated as four words.

5.9 dtSearch default alphabet file text

Some of the characters in the alphabet file are not printable, so screenshots were used instead of the actual text. You cannot copy or paste the Spaces or Ignore characters since they are not printable. Instead, use the dtSearchDefaultAlphabetFile configuration value in the Configuration Table Guide to update the dtSearch default alphabet file.

[Letters] // Original letter, lower case, upper case, unaccented
0 0 0 0
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
5 5 5 5
6 6 6 6
7 7 7 7
Relativity | Searching Guide - 60

CJKRanges = 2e80-ac00 ac00-d7af f900-faff fe30-fe4f
6 Saved search

A saved search is a saved set of criteria that returns the latest documents that meet that criteria. For example, if you want to reference documents that contain the terms "confidential" and "property" and are also marked as Relevant, you can create a saved search with that criteria. However, saved searches can be much more complex.

In Relativity, you can create saved searches by defining custom queries and unique views, as well as by selecting public or private security settings, specific folders to query, and nested sort orders. You can also execute a search on the fly, save it for later use, or perform a combination of these tasks.

Since saved searches are executed in real-time, you save the search definition but not the results. Relativity executes the search each time your click on it in the Saved Searches browser and when you return to it after performing other tasks in the workspace. This functionality ensures that only data meeting the search criteria is returned in the result set. (You can set the Requires Manual Rerun option to control this functionality.)

Upon the saved search execution, Relativity first applies the conditions, then the family, then the filters. Note that conditions are on the advanced search page, and filters are applied on the item list. The only time when this order isn’t accurate is when you have nested relational searches. For example, you have Relational Search A, which relies on the results of Relational Search B. In this scenario, Relativity applies the innermost filters (in this case B), then the family that relies on it; then the filters of the outer searches (in this case A), then the family condition; and so on, until the search is complete, and Relativity applies filters as the final step.

You can also use saved searches as the building blocks in other Relativity features. For example, you’re required to select a saved search when you create batches, build a dtSearch or Analytics index, define an imaging set, and perform other tasks in Relativity.

Using saved searches

You need access to all the emails in your workspace that were sent between Jan 1, 2013 and Oct 8, 2014 because they contain many references that are vital to your client’s case. You’re going to reference these documents multiple times throughout review, so it’d be nice to have a way to save them and not have to search for them each time. Relativity’s saved search feature that allows you to do just that.

You go to the saved search browser and create a new saved search. You set the Includes field to Include Family because you need to return files with the same group identifier as the files that meet the field conditions you’re about to enter. For the Conditions field, you select the Sent Date field with an Operator of between. For the two Value choices you select 1/1/2013 and 10/8/2014.
When you click Save & Search, Relativity returns only email that fall in the date range you specified, and any reviewer with permissions to this saved search can easily bring up these documents in the saved search browser at any time.

### 6.1 Navigating the saved searches browser

On the Documents tab, you can click ![view icon](view_icon.png) to view the Saved Searches browser. This browser provides you with features used to create, organize, edit, and perform other tasks with saved searches.
The Search Folder Tree displays the following options:

- **New Search** button - Click to display the Saved Search form. To display this form, you can also click any folder, including the root folder.

- Public or Private - The icons display next to the name of a saved search to indicate its visibility.

- Search List - Displays saved searches in a list view. See [Using search list options on the next page](#).

- Refresh - Refreshes the list view and tree hierarchy.

**Note:** Relativity automatically performs a refresh when you create a new search or modify an existing one. However, you need to click to display new searches or modifications to existing ones made by other users.

- Show Current Path - Displays the location of the search. For example, the path Salt vs. Pepper\Admin\Batch Source indicates that the search is located in a Batch Source sub-folder contained in the Admin folder, which is under the root folder Salt vs. Pepper.

- Email Icon - Opens an email message containing a link to the saved search. The Subject line is pre-populated with the following text: "Relativity Review - <Workspace Name> - <Search Name>."
When the recipient clicks on the link, the saved search is displayed with the current result set.

**Note:** Relativity displays a permissions denied message if the recipient clicks the link to display the search but does not have access rights to it.

- **Edit Search** - Displays the Search Form for updating search criteria.

- **Save Search** - Click to display a pop-up where you can select a new Owner, and modify the search name. See [Creating or editing a saved search on page 68](#).

- **Search Right-click Menu** - Highlight a search in the folder tree to display a right-click menu with the following options:
  - **Edit** - Displays the Saved Search form, where you can modify the current settings for the search.
  - **Copy** - Adds a duplicate of the search to the tree.
  - **Delete** - Permanently removes the search from the database.
  - **Secure** - Available on public searches, this option displays a security page so that you can override the security inherited from the workspace, or parent folder. See the Admin guide for more information on Setting permissions on Relativity objects.
  - **Email Link** - Opens an email message containing a link to the saved search. For additional information, see the description for E-mail Icon.

- **Folder Right-click Menu** - Highlight a folder to display a right-click menu for managing folders. See [Organizing saved searches in folders on page 66](#).

### 6.1.1 Using search list options

To display saved searches in a list view, click in the folder tree.
The search list view provides the following options and information:

- **Search Folder Tree** - Displays saved searches in a tree hierarchy.
- **Edit link** - Click to display the Saved Search form, where you can modify the settings for the search you're currently viewing.
- **Name link** - Click this link to display documents returned by the search in the item list. You can open documents from this list.
- **Path** - Displays the location of the search. For more information, see **Show Current Path** icon.
- **Email** - Opens an email message containing a link to the saved search. The Subject line is pre-populated with the following text: "Relativity Review - <Workspace Name> - <Search Name>." When the recipient clicks on the link, the saved search is displayed with the current result set.

  **Note:** Relativity displays a permissions denied message if the recipient clicks the link to display the search but doesn't have access rights to it.

- **Sorting** - Click a column heading to sort in ascending order (**Down** arrow displays), or click twice to sort in descending order (**Up** arrow displays). Click a descending sort to clear it, and to display the original sorting order.
- **Export the list** - Click to create and download an Excel spreadsheet with a list of saved searches.
- Paging - Click on arrows at the bottom of the browser to display the next or previous page of searches, as well as first or last page of the entire list.

- Mass Operations - Select searches for a mass operation. Next, select Copy, Move, or Delete in the drop-down box at the bottom of the browser. Click Go. For more information on mass operations, see the Admin Guide.

### 6.2 Controlling the visibility of saved searches

On the saved search form, you can control the visibility of a search by setting the Owner option. New searches are private by default, making them visible only to you and Relativity administrators. In addition to owner access, users must have permissions to the Saved Searches Browser and at least view permissions for Search on the security page. For more information on workspace permissions, see the Admin Guide.

You can change the visibility of a search by selecting one of these options in the Owner drop-down menu:

- **Public** - Makes the search available to all users with the appropriate permissions.

  Note: You can configure Relativity to make your saved searches public by default. When you create a search, the Owner box will display Public. In My Settings, select Yes in the option Advanced Search Public By Default.

- **User Name** - Select a specific user from the drop-down menu. The search will be visible only to that user and Relativity administrators.

- **Me** - Click this button to reset the visibility on the search to private. Your name appears in the Owner box.

You define the criteria used for saved searches in the Conditions section of the Saved Search form. You can build complex queries using a combination of fields and operators that are set to required values. For information about the operators available for building these queries, as well as specific options for searching batches and developing combined searches, see Defining criteria for saved searches on page 73.

### 6.3 Organizing saved searches in folders

You can organize saved searches by adding them to securable folders that you create and manage in the Saved Searches browser. To work with search folders, you must have the appropriate permissions for Search Folder, Search, and the Saved Searches Browser on the security page. For more information on workspace permissions, see the Admin Guide.

#### 6.3.1 Adding sub-folders to the root

In the Search Folder Tree, right-click on the root folder to add sub-folders to the browser. Click Create to add a new folder, and name it something descriptive of its contents. To update the folder name, right-click on the folder, and click Rename.
6.3.2 Managing subfolders
Right-click on a folder under the root to display the following menu options:

- **Create** - Adds a subfolder to the highlighted folder.
- **Rename** - Makes the folder name editable. Enter new text for the name.
- **Secure** - Displays a security page so that you can override the security inherited from the workspace, or parent folder. For more information on setting permissions on objects, see the Admin Guide.
- **Delete** - Permanently removes all the searches and subfolders that folder contains from the database.
6.3.3 Adding existing searches to folders
To add existing searches to a folder, left click the search and then drag and drop it into the folder. Click OK on the confirmation message.

**Note:** When you move a search, it inherits the security from the parent folder. You may want to check the security on a folder before moving a search into it.

6.4 Creating or editing a saved search
To create or edit a saved search, follow these steps:

1. Click  at the bottom of the browser.
2. Click **New Search** button above the document list manager. To edit a search, right-click on the name, and click **Edit**.
   
   **Note:** If you don't see the New Search button, you may have another saved search selected. Click the top-level folder in the browser to deselect another search.

3. Fill out the fields on the form. See [Fields on the next page](#).
4. Click **Save**.
6.4.1 Fields

The Information fields are:

- **Name** - Enter a title for the search.

- **Includes** - Select an option for returning documents related to hit documents. (Hit documents match the search criteria.) The related documents will be included in the result set, but they do not need to match the search criteria.
  - **Include Duplicates** - Use this setting if you want the result set to include documents with the same MD5 Hash values as the hit documents. (The MD5 Hash value is used as a unique file identifier.)
  - **Include Family** - Use this setting if you want the result set to include documents with the same group identifiers as the hit documents.
  - **Include <Custom Field>** - Your organization may use custom related fields. Contact your Relativity administrator for additional information.

- **Scope** - Select one of these options to designate the document set for the search:
  - **Entire Workspace** - All documents in the workspace are searched.
  - **Selected Folders** - Select this option, and then click **Select Folders**. On the Select Folders popup, choose the checkboxes for the folders that you want to search. Clear the **Include Subfolders** checkbox on the popup if you don't want subfolders included.

- **Requires Manual Rerun** - Select this option if you want to require users to rerun a saved search when they return to it after navigating to other features in Relativity. Selecting this option only affects the search that the current user is running. It doesn't affect any parent or nested searches tied to the current search. If you have a search that has Requires Manual Rerun checked and you include it as the criteria for another search, it will rerun. See [Rerunning out-of-date saved searches on page 72](#).

  **Note:** The Requires Manual Rerun option is for searches that might take a long time to run, and you don't need them to run automatically when you navigate back to the saved searches. This keeps you from having to manually cancel queries before you can navigate away from that search.

The Search Conditions fields are:

- **Search With** - Select Keyword, dtSearch, or Relativity Analytics index for your search engine.

  **Note:** The Search With option is hidden when only a Keyword Search index exists. Additionally, the **Search Text** option appears only for Keyword Searches, while other options are available for dtSearches and Analytics. Your selection for the **Search With** option doesn't affect the availability of Field, Operator, or Value options displayed in the Conditions section of the form.

- **(Keyword Only) Search Text** - Enter search terms or conditions. See [Keyword search on page 18](#).

- **(dtSearch Only) Search Terms, Fuzziness Level, or Enable Stemming** - Enter search terms, and select settings as necessary. See [dtSearch on page 27](#).
■ (Analytics Only) **Keywords, Fuzziness Level, Concepts, Min Concept Rank** - Enter terms and select settings as necessary. See the Analytics Guide.

■ **Sort By Rank** - Select this option to return results in order by relevance. The most relevant documents are listed at the top of the result set. The documents return in the following ways per index:
  - **Analytics** - the rank is the actual conceptual rank (0.00 - 1.00) indicating how similar in concept the document is to the query. The higher the number, the more similar it is to what was searched.
  - **dtSearch and Data Grid** - the rank values indicate a binary hit/no hit for the results. Currently there are no gradations or relative ranking.
  - **Keyword** - the rank is a number between 1-1000 determined by a SQL ranking system generated for the search terms entered.

The Conditions fields are:

In the Conditions section, you set multiple search conditions to create complex queries. To remove a condition, click **Clear** for the associated row. Each row represents a separate criterion. Depending on the type of field you select, there will be different operators available. Each box within the row represents a different part of the query:

■ **Left or Opening Parentheses** - Use the first box in the row to select a single, double, or triple parentheses for grouping criteria or controlling precedence in the query.

■ **Field** - Select a field available in your workspace.

■ **Operator** - Select a search operator. The field type determines the available operators.

■ **Value** - Enter or select a value. The field type determines the available values.

■ **Right or Closing Parentheses** - Use the last box in the row to select a single, double, or triple parentheses to group criteria.

■ **AND or OR Operators** - Use these operators to connect the criterion in each row.

![Conditions](image_url)

**Note:** Using the Extracted Text OR Full Text field as a condition searches across the Extracted text or Full Text fields but only with the operators of Contains or Does Not Contain. This is identical to the way a Keyword search acts.

The **Fields (Required)** are:

■ **Available Fields**: These fields are listed in left box.

■ **Selected Fields**: These fields are listed in the right box, and they will be displayed for your search results. They are ordered based on their position in this box.
Use the following options to move the fields between the Available and Selected Field boxes:

- **Move a Single Field**: Highlight a field and click the Right or Left single arrow. You can also double-click on a field name.

- **Move Multiple Fields**: Hold the SHIFT key down. Click on a range of fields that you want to move, or drag the mouse over these fields. Click the Right or Left single arrow to move the fields between boxes.

- **Move All Fields**: Click the Right or Left double arrow to move all the fields from one box to another.

- **Change the Selected Fields Order**: Use the Up and Down arrows to reorder the fields. Topmost fields display as a columns on the left of the item list, while fields at the bottom display on the right.

**Note**: You can edit the default list of selected fields displayed in this option. On the Views tab, click **Default Search Views**, and then click **Edit** to update the field list.

The Sort options are:

You can define the default sort order used for the search results. Each row in a sort criterion contains the following options:

- **Sort Field**: Select a field from the left drop-down box. You will sort on this field.

- **Order**: Select ascending or descending from the right drop-down box.

You can create a nested sort order by selecting fields in several drop-down boxes.

The Other fields are:

- **Relativity Applications** - Select one or more applications that you want associated with the saved search. On the details view of an application, you must link the Document object to the application before you see your search in the Saved Search associate list. For more information, see Creating an application in Relativity on the Relativity 9.2 Developers site.

### 6.4.2 Using pop-up pickers

You may have the option to select values from a pop-up picker when you choose certain fields or operators in the Conditions section. For example, popups are available when you select the following operators:

- Any of these
- None of these
- All of these (only for multiple object fields)
- Not all of these (only for multiple object fields)

See [Creating or editing a saved search on page 68](#). For information about setting batch conditions, see [Batch fields as search conditions on page 76](#).

### 6.4.2.1 Select items pop-up picker

Use the following general steps to select items in the picker:
1. Navigate to the Saved Search form or use the Search Conditions feature.
2. Select a Field option for a condition.
3. To display the picker, click \[\ldots\] in the Value box.
4. Select one or more items in the Available Items list. A checkmark indicates an item is selected.
5. Click Add. The items are displayed in the Selected Items list.

**Note:** To remove an item from this list, select it and click Remove. The item is displayed in the Available Items list.

6. Click Set. The items are added to the Value box in the search form.

### 6.4.2.2 System user fields
System user fields include the System Created By and System Last Modified By fields, which you can use in search conditions.

1. Navigate to the Save Search form or use the Search Conditions feature.
2. Select a system user field in the Field option for a condition.
3. Select an operator, and perform one of the following tasks:
   - If you selected any of these or none of these, click the ellipsis \[\ldots\] in the Value box to display the Select Options pop-up. Choose one or more user names, and click OK.
   - Enter the user name in the textbox.
4. Define any additional search criteria as needed.

### 6.4.2.3 Folder name field
You can select Folder Name as a field in a search condition to create more flexible queries than using the Scope section of the Saved Search form. You can combine conditions containing the Folder Name and other fields with AND or OR operators refining your search criteria.

1. Navigate to the Save Search form or use the Search Conditions feature.
2. Select Folder Name in the Field option for a condition.
3. Select an operator, and perform one of the following tasks:
   - If you selected any of these or none of these, click in the Value box to display Select Folder pop-up. Choose one or more folders, and click OK.
   - Enter the folder name in the textbox.
4. Define any additional search criteria as needed.

### 6.4.3 Rerunning out-of-date saved searches
You may need to rerun a saved search when you return to it after navigating to other features in Relativity. Instead of seeing your search results, you see a message indicating that your search is out of date.
**Note:** To enable the Run saved search feature, select the Require Manual Rerun option in the Information section of the Saved Search form.

Perform one of these tasks:

- Click **Run saved search** to reload your search results. You can also click on the saved search in the browser to rerun the search.
- Click **Edit Search** to display the Saved Search form where you modify the search settings.

If you edit an item returned in your saved search, you will also need to rerun it. You must rerun the search even when the edited item still meets the search criteria, and the number of documents returned is unchanged.

### 6.4.4 Copying a saved search to another workspace

You can copy a saved search contained in one workspace to another workspace if the saved search references Keyword Search and is copied to the new workspace as part of a template.

For saved searches that include other saved searches as conditions, these are only copied over as part of a template if all of the searches referenced use Keyword Search.

### 6.5 Defining criteria for saved searches

You define the criteria used for saved searches in the Conditions section of the saved search form. You can build complex queries using a combination of fields and operators that are set to required values. This section provides information about the operators available for building these queries, as well as specific options for searching batches and developing combined searches. For information about building queries, see Creating or editing a saved search on page 68.

#### 6.5.1 Operators

When defining search criteria in the Conditions section of a saved search form, you use operators to determine how a field is queried for the value that you selected or entered. The operators available for a search criterion depend upon the field type:

- Fixed-length, long, or extracted text operators below
- Whole number, decimal, and currency operators on page 75
- User operators on page 75
- Date operators on page 76
- Yes or no operators on page 76
- Single and multiple choice field operators on page 76

#### 6.5.1.1 Fixed-length, long, or extracted text operators

The following operators are available for fixed-length, long, and extracted text field types.
<table>
<thead>
<tr>
<th>Operator</th>
<th>Returns Documents Where...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is like</td>
<td>The field contains all or part of the entered term. Blank values are filtered out, and search values can be enclosed in double quotation marks. A partial match does not require a wildcard (*). Relativity returns an OR operator between terms when a condition uses the &quot;Is like&quot; operator, and the terms are separated by a carriage return into multiple lines, or they are separated by a comma within a single line. For example, Relativity interprets a query as &quot;Field is like Term 1 or Field is like Term 2 or Field is like Term 3&quot;. <strong>Note:</strong> For performance reasons, Relativity recommends using the &quot;Contains&quot; operator rather than building queries with single or multiple uses of the &quot;Is like&quot; operator. See FAQs for Contains and Is Like operators below.</td>
</tr>
<tr>
<td>Is not like</td>
<td>The field does not contain the entered term. Wildcards (*) are already applied at the beginning and end of a term with this operator. The comments about the &quot;Is like&quot; operator also apply to &quot;Is not like&quot;.</td>
</tr>
<tr>
<td>Is</td>
<td>The field value equals any of the entered items. Multiple values can be separated by a comma or carriage return. Blank values are filtered out, and search values can be enclosed in double quotation marks.</td>
</tr>
<tr>
<td>Is not</td>
<td>The field value does not equal the entered term.</td>
</tr>
<tr>
<td>Is set</td>
<td>The field is not empty.</td>
</tr>
<tr>
<td>Is not set</td>
<td>The field is empty.</td>
</tr>
<tr>
<td>Is less than</td>
<td>The field value is less than the entered term.</td>
</tr>
<tr>
<td>Is greater than</td>
<td>The field value is greater than the entered term.</td>
</tr>
<tr>
<td>Contains</td>
<td>The field includes the entered term. This operator is available for long text and fixed length text fields included in the full text index. Blank values are filtered out, and search values can be enclosed in double quotation marks. You can also use the AND/OR operators, and add wildcards (*) to the end of the search. Relativity returns an OR operator between terms when a condition uses the &quot;Contains&quot; operator, and the terms are separated by a comma. (You can also use the OR operator to separate search terms.) For example, Relativity interprets these queries as &quot;Field contains Term 1 or Field contains Term 2 or Field contains Term 3&quot;, and &quot;Field contains Term 1 or Term 2 or Term 3&quot; respectively. Relativity returns an AND operator between terms when a condition uses the &quot;Contains&quot; operator, and the terms are separated by a carriage return into multiple lines, or if terms are separated by a space on a single line so &quot;Field contains Term 1 Term 2 Term 3&quot; is interpreted as &quot;Field contains Term 1 AND Term 2 AND Term 3&quot;. <strong>Note:</strong> The &quot;Contains&quot; operator works identically to keyword search, except that it searches that specific field.</td>
</tr>
<tr>
<td>Does not contain</td>
<td>The field does not contain the entered term. This operator is available for long text and fixed length text fields included in the full text index. Blank values are filtered out, and search values can be enclosed in double quotation marks.</td>
</tr>
</tbody>
</table>

**FAQs for Contains and Is Like operators**

You can improve your searches by understanding the differences between the "Contains" and "Is like" operators.
Why do searches using the "Is like" operator tend to run slowly?
The "Is like" operator can slow the performance of your system because it queries every document for the field specified in the condition. For performance reasons, we don't recommend building queries with single or multiple uses of the "Is like" operator. Instead, you can use the "Contains" operator, which improves performance by querying only the fields in the full-text index.

Are there any special requirements for using the "Contains" operator?
Yes, the full text-index must include the field that you want to search with the "Contains" operator.

How does query execution differ for searches with the "Contains" and "Is like" operators?
The "Contains" operator queries the SQL full text catalog, while the "Is like" operator queries the database table inside the catalog. The "Is like" operator prevents other queries from editing the table until it completes, which can negatively affect performance.

Why are different search results returned by queries using the "Contains" versus "Is like" operators?
The difference in result sets is caused by the way SQL interprets queries using these operators. An "Is like" statement appends a wildcard to the front of each query, which sometimes causes it to return more items than a query with the "Contains" operator. You must evaluate these additional items to determine if they're actually part of your expected result set or if they represent false hits for the items that you want to return.

Is there any way to enhance the performance of queries using the "Is like" operator?
Yes, you can make queries that use the "Is like" operator more efficient by creating a SQL index on the table column referenced by the query. The query can point to this index, and avoid accessing the table. For more information, contact your system admin.

How can I use the "Contains" operator to facilitate document reviews?
You can use the "Contains" operator to search fields in email headers for email addresses and other pertinent header information. To perform these searches, the Author, TO, CC, and BCC fields in email headers must be added to the full text index.

6.5.1.2 Whole number, decimal, and currency operators
The following operators are available for whole number, decimal, and currency field types.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Returns Documents Where...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is</td>
<td>The entered number is equal to the field value.</td>
</tr>
<tr>
<td>Is not</td>
<td>The entered number is not equal to the field value.</td>
</tr>
<tr>
<td>Is set</td>
<td>The field is not empty.</td>
</tr>
<tr>
<td>Is not set</td>
<td>The field is empty.</td>
</tr>
<tr>
<td>Is less than</td>
<td>The field value is less than the entered number.</td>
</tr>
<tr>
<td>Is greater than</td>
<td>The field value is greater than the entered number.</td>
</tr>
</tbody>
</table>

6.5.1.3 User operators
The following operators are available for user field types.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Returns Documents Where...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is logged in user</td>
<td>The logged in user is equal to the field value.</td>
</tr>
<tr>
<td>Any of these</td>
<td>Any of the selected users match the field value.</td>
</tr>
<tr>
<td>None of these</td>
<td>The selected users do not match the field value.</td>
</tr>
<tr>
<td>Is set</td>
<td>The field is not empty.</td>
</tr>
</tbody>
</table>
### 6.5.1.4 Date operators

The following operators are available for date field types.

You can search on date and time but time is not displayed by default when you select a date on the calendar pop-up. For example, you can search on 10/16/2001 3:57 PM by typing in the time after your selected the date. You can also search for 2/3/10 between 4/3/10 and both 2/3/10 and 4/3/10 return in the results.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Returns Documents Where...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is not set</td>
<td>The field is empty.</td>
</tr>
<tr>
<td>Is not</td>
<td>The entered date is not equal to the field value.</td>
</tr>
<tr>
<td>Is set</td>
<td>The field is not empty.</td>
</tr>
<tr>
<td>Is not set</td>
<td>The field is empty.</td>
</tr>
<tr>
<td>Is before</td>
<td>The field value is before the entered date.</td>
</tr>
<tr>
<td>Is before or on</td>
<td>The field value is before or on the entered date.</td>
</tr>
<tr>
<td>Is after</td>
<td>The field value is after the entered date.</td>
</tr>
<tr>
<td>Is after or on</td>
<td>The field value is on or after the entered date.</td>
</tr>
<tr>
<td>Between</td>
<td>The field value is between the two entered dates.</td>
</tr>
<tr>
<td>Is in</td>
<td>The field value is within the selected range.</td>
</tr>
</tbody>
</table>

### 6.5.1.5 Yes or no operators

The following operators are available for Yes/No field types.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Returns Documents Where...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is</td>
<td>The selected value (Yes or No) is equal to the field value.</td>
</tr>
<tr>
<td>Is not</td>
<td>The selected value (Yes or No) is not equal to the field value.</td>
</tr>
<tr>
<td>Is set</td>
<td>The field is not empty.</td>
</tr>
<tr>
<td>Is not set</td>
<td>The field is empty.</td>
</tr>
</tbody>
</table>

### 6.5.1.6 Single and multiple choice field operators

The following operators are available for single and multiple choice field types.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Returns Documents Where...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any of these</td>
<td>Any of the selected choices are present in the field.</td>
</tr>
<tr>
<td>None of these</td>
<td>None of the selected choices is present in the field.</td>
</tr>
<tr>
<td>All of these</td>
<td>All of the selected choices are present in the field.</td>
</tr>
<tr>
<td>Not all of these</td>
<td>The selected choices are not present in the field.</td>
</tr>
<tr>
<td>Is set</td>
<td>The field is not empty.</td>
</tr>
<tr>
<td>Is not set</td>
<td>The field is empty.</td>
</tr>
</tbody>
</table>

### 6.5.2 Batch fields as search conditions

Relativity includes several fields related to batching that you can use as conditions for searching across batch sets. To use batch fields as search conditions, follow these steps:
1. Create a new saved search. See Creating or editing a saved search on page 68.

2. In the Conditions section of the saved search form, select **Batch** in the Field drop-down menu.

3. Choose the **Operator** that you want to use.

4. Click the blue dot in the **Value** column to display the Select Batch Criteria popup.

5. Select one of the following batch related fields:
   - **Batch**
   - **Batch::Batch Set**
   - **Batch::Assigned To**
   - **Batch::Status**

   As illustrated below, you can use the **Assigned To** field to search for batches set to a specific user.

See Operators on page 73 for more information.

### 6.5.3 Using saved searches as conditions (combined searches)

You can combine searches by selecting previously created saved searches as Field conditions. To use a saved search as a search condition, follow these steps:

1. Create a new saved search. See Creating or editing a saved search on page 68.

2. In the Conditions section of the saved search form, select *(Saved Search)* from the top of this list in the **Field** drop-down menu.

3. Select an option in the **Operator** drop-down menu.

4. Click the blue dot in the **Value** column to display a pop-up window.

5. Select a saved search and click **OK**. As illustrated below, you can use multiple save searches as Field conditions. Including multiple searches in a query may affect search performance.

**Note:** Beginning with the Relativity 9.1.137.12 release, you can delete a saved search that is referenced by another saved search through the mass delete operation.
6.5.3.1 Preventing circular references
Relativity prevents you from creating recursive searches when you combine multiple searches as conditions in a query. For example, you might create a combined search using Saved Searches 1, 2, and 3 as follows:

- Saved Search 2 uses Saved Search 1 as a Field condition.
- Saved Search 3 uses Saved Search 2 as a Field condition.

When you edit Saved Search 1, you can't select Saved Search 2 or Saved Search 3 as Field conditions. Relativity prevents you from creating a circular reference by not listing these searches in the Value pop-up window.

6.5.3.2 Including related items in combined searches
You can combine saved searches to check for conflicts within related item groups. For example, a review manager may use a combined search for quality control when preparing to produce responsive documents for a case.

Use the following steps to confirm that a search for responsive documents doesn't include any privileged documents.

1. Create a saved search called Responsive Check that uses the Includes Family option and sets a condition on a field, such as Designation to Responsive. This search specifies production criteria that return only responsive documents.

2. Create a second saved search called Privilege Check that uses the Includes Family option and sets a condition on a field, such as Privilege Description to Privileged, Attorney Client, and so on. This search is used for evaluation purposes.

3. Create a combined search called Conflict Check that uses the Includes Family option and sets conditions for the Responsive Check and Privilege Check searches. This quality control search determines if any privileged documents are included in the production-eligible saved search.

6.5.4 Lists as search conditions
If you've created saved lists using the Lists feature, you can add lists as criteria in a saved search.

Follow these steps to create a new saved search using a saved list of documents as the search criteria:

1. On the Documents tab, click to open the Saved Searches browser.

2. Click New Search.

3. Type a name for the saved search in the Name field.

4. Add a condition with the following column settings:
   - Field - select Lists.
   - Operator - select these conditions.
   - Value - select the following value criteria:
     - Field - select Lists.
     - Operator - select any of these.
- **Value** - select one or more saved lists to include in the search criteria.

5. In the **Fields** category, select the fields you want to include when viewing your saved search results. Select the **Lists** field to show the list(s) with which an object is associated.

6. Click **Save** to save the search. Or, click **Save & Search** to save and execute the search.

### 6.6 Saving searches on the Documents tab

On the Documents tab, you can click **Save as Search** to save a search based on the criteria that you've already set for filtering, Keyword searches, dtSearches, or Analytics. You can enter a unique name for the search, select an owner, and choose a folder for it. Relativity also saves any settings that you have selected for the view, sort order, or other features that control how your results appear.

After you save your search, it appears in the saved searches browser, and you can modify it using the same options available for saved searches. See [Creating or editing a saved search on page 68](#).

**Note:** To use **Save as Search**, you must have add permissions for **Search**, and access to the **Saved Searches Browser** assigned to you through the security page. For more information on workspace conditions, see the Admin guide.

To create a saved search on the Documents tab:

1. Navigate to the Documents tab.

2. (Optional) In the **Browser** menu, select the **Folders**, **Field Tree**, or **Clusters** option. The item list for the selected browser is displayed.

3. Filter on the documents in the list or select a keyword or other search option. To set search criteria, see the specific instructions for running Keyword searches, dtSearches, Analytics, or filters. Your search results will be displayed in the item list on the Documents tab. Any folder, tag, or cluster selected in the browser will be included as a condition when the search is saved. The currently displayed columns and sort order will also be saved.

   **Note:** If you select an option in the Include Related Documents drop-down menu, and also set the filters, you will not be able to save this search with **Save as Search**. You can set the Include Related Documents drop-down menu to blank or clear all filters, and then use this link to save the search.

4. Click **Save**.

5. Select or enter the following information:
   - **Owner** - Select **Public** to make the search available to all users or choose a specific user from the list. Click **Me** to select your name from the list, making the search private. (Users must have the appropriate privileges to view searches.) See [Controlling the visibility of saved searches on page 66](#).
   - **Name** - Enter a title for the search. The title appears in the saved searches browser.
- **Folder List** - Click the Plus button to expand the folders in the list. Highlight the folder where you want to add the search.

6. Click **Save**.

Your search criteria are saved, and the search is added to the saved searches browser. You can click the button to display this browser.

To update the search, follow the same steps as those used to edit a saved search. Right-click on the search in the saved searches browser, and click **Edit**. The saved search form pre-populates with information used in your search.

### 6.7 Saved search history

On the History tab, you can view the audit records for saved searches, unsaved searches, and queries performed on views (such as filtering on a column).

<table>
<thead>
<tr>
<th>Name</th>
<th>Object Type</th>
<th>User Name</th>
<th>Timestamp</th>
<th>Execution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documents</td>
<td>View</td>
<td>Smith, Jane</td>
<td>10/9/2014 5:07:13 PM CDT</td>
<td>2202</td>
</tr>
</tbody>
</table>

Use the following guidelines to view search history records:

- **Saved Searches** - the **Name** column displays the name of the search, and the **Object Type** column displays "Search." Click the **Query** link to display a pop-up window with the SQL statement for the query.
- **Unsaved Searches or Views** - the Name column displays the name of the view in which the search was performed, and the Object Type column displays "View." Click the name link to display the details page for the view, or click the Query link to display a pop-up window with the SQL statement for the query.
7 Searching with Regular Expressions (RegEx)

A regular expression is a form of advanced searching that looks for specific patterns, as opposed to certain terms and phrases. With RegEx you can use pattern matching to search for particular strings of characters rather than constructing multiple, literal search queries.

RegEx uses metacharacters in conjunction with a search engine to retrieve specific patterns. Metacharacters are the building blocks of regular expressions. For example, “\d” in a regular expression is a metacharacter that represents a digit character, but “d” stands for the literal character, “d.” You can use regular expressions to search for social security numbers, patent numbers, URLs, email addresses, Bates numbers, and other strings that follow a specific pattern.

There are several implementations of RegEx. The differences in implementations usually include the way special characters are handled and how character classes are treated.

7.1 Use cases for Regular expressions

RegEx can help you in cases where you need to find different numbers that contain the same pattern, for example, the following serial numbers:

- XFRD-8324-ERWH-3231
- GHSR-3413-KBKV-8173
- MPFS-1357-QEGT-9376

Instead of writing three literal search strings to match each serial number, you can construct one regular expression to match the serial numbers’ pattern. This single RegEx returns any document that contains any of the three serial numbers.

- Pattern: 4 letters-4 digits-4 letters-4 digits
- RegEx: `[a-z]{4}-[0-9]{4}-[a-z]{4}-[0-9]{4}`

**Note:** Think of each RegEx as a phrase when you construct your search string. If you switch the order of the string you won’t receive the same results.

You cannot use capital letters when constructing a RegEx in dtSearch. Thus, if you are searching for varying strings that all begin with NLRT, such as:

- NLRT-0381
- NLRT-6334
- NLRT-9167

- The proper Relativity RegEx is: "##nlrt-\d{4}".

7.2 Regular expression metacharacters

Metacharacters are the building blocks of regular expressions. Characters in RegEx are understood to be either a metacharacter with a special meaning or a regular character with a literal meaning.
The following are some common RegEx metacharacters and examples of what they would match or not match in RegEx.

<table>
<thead>
<tr>
<th>Metacharacter</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>\d</td>
<td>Whole Number 0 - 9</td>
<td>\d\d\d = 327</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\d\d = 81</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\d = 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\d\d\d ≠ 24631</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\d\d\d doesn’t return 24631 because 24631 contains 5 digits. \d\d\d only matches for a 3-digit string.</td>
</tr>
<tr>
<td>\w</td>
<td>Alphanumeric Character</td>
<td>\w\w\w = dog</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\w\w\w\w = mule</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\w\w = to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\w\w\w = 467</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\w\w\w\w = 4673</td>
</tr>
<tr>
<td></td>
<td></td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\w\w\w ≠ boat</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\w\w\w doesn’t return boat because boat contains 4 characters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\w ≠ !</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\w doesn’t return the exclamation point ! because it is a non-alphanumeric character.</td>
</tr>
<tr>
<td>\W</td>
<td>Symbols</td>
<td>\W = %</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\W = #</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\W\W\W = @#%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\W\W\W\W ≠ dog8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\W\W\W\W doesn’t return dog8 because d, o, g, and 8 are alphanumeric characters.</td>
</tr>
<tr>
<td>[a-z]</td>
<td>Character set, at least one of</td>
<td>pand[ora] = panda</td>
</tr>
<tr>
<td>Metacharacter</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>----------</td>
</tr>
</tbody>
</table>
| [0-9]         | which must be a match, but no more than one unless otherwise specified. The order of the characters does not matter. | pand[ora] = pando
pand[ora] ≠ pandora
pand[ora] doesn't bring back pandora because it is implied in pand[ora] that only 1 character in [ora] can return.

(Quantifiers that allow pand[ora] to match for pandora is discussed below.) |
| (abc) (123)   | Character group, matches the characters abc or 123 in that exact order. | pand(ora) = pandora
pand(123) = pand123
pand(oar) ≠ pandora
pand(oar) does not match for pandora because it's looking for the exact phrase pandoar. |
| | Alternation - allows for alternate matches. | pand(abc|123) = pandora OR pand123 |
| | Question mark matches when the character preceding ? occurs 0 or 1 time only, making the character match optional. | colou?r = colour (u is found 1 time)
colou?r = color (u is found 0 times) |
| | Asterisk matches when the character preceding * matches 0 or more times. | tre* = tree (e is found 2 times)
tre* = tre (e is found 1 time)
tre* = tr (e is found 0 times)

tre* ≠ trees
tre* doesn't match the term trees because although "e" is found 2 times, it is followed by "s", which is not accounted for in the RegEx. |
<table>
<thead>
<tr>
<th>Metacharacter</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
</table>
| +             | Plus sign matches when the character preceding + matches 1 or more times. The + sign makes the character match mandatory. | tre+ = tree (e is found 2 times)  
tre+ = tre (e is found 1 time)  
 tre+ ≠ tr (e is found 0 times)  
tre+ doesn't match for tr because e is found zero times in tr. |
| . (period)    | The period matches any alphanumeric character or symbol. | ton. = tone  
ton. = ton#  
ton. = ton4  
 ton. ≠ tones  
ton. doesn't match for the term tones because . by itself will only match for a single character, here, in the 4th position of the term. In tones, s is the 5th character and is not accounted for in the RegEx. |
| .*            | Combine the metacharacters . and *, in that order .* to match for any character 0 or more times. | tr.* = tr  
tr.* = tre  
tr.* = tree  
tr.* = trees  
tr.* = trough  
tr.* = treadmill |

### 7.2.1 RegEx quantifiers

RegEx use quantifiers to indicate the scope of a search string. You can use multiple quantifiers in your search string. The following table gives examples of the quantifiers you can use in your RegEx:
<table>
<thead>
<tr>
<th>Quantifier</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>{n}</td>
<td>Matches when the preceding character, or character group, occurs n times exactly.</td>
<td>\d{3} = 836 \n\d{3} = 139 \n\d{3} = 532 \npand[ora] \n{2} = pandar \npand[ora] \n{2} = pandoo \npand(ora) \n{2} = pandoraora \npand[ora] \n{2} ≠ pandora \npand[ora] \n{2} doesn’t match for pandora because the quantifier \n{2} only allows for 2 letters from the character set [ora].</td>
</tr>
<tr>
<td>{n,m}</td>
<td>Matches when the preceding character, or character group, occurs at least n times, and at most m times.</td>
<td>\d{2,5} = 97430 \n\d{2,5} = 9743 \n\d{2,5} = 97</td>
</tr>
</tbody>
</table>

**Quantifier**

- \{n\}
- \{n,m\}
<table>
<thead>
<tr>
<th>Quantifier</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>\d{2,5} ≠ 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9 does not match because it is 1 digit, thus outside of the character range.</td>
</tr>
</tbody>
</table>

### 7.2.2 Escaping RegEx Metacharacters

When using RegEx to search for a character that is a reserved metacharacter, use the backslash \ to escape the character so it can be recognized. The following table gives an example on how to escape a reserved metacharacter when searching.

<table>
<thead>
<tr>
<th>Search For</th>
<th>RegEx</th>
<th>Match Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>UK phone number</td>
<td>+[0-9]{11}</td>
<td>+14528280001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+38119930978</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the + sign is not escaped with a backslash, RegEx treats + as a quantifier instead of the literal plus sign character.</td>
</tr>
</tbody>
</table>

### 7.3 Using regular expressions with dtSearch

You can use RegEx with your dtSearch index to search for things like Bates numbers, zip codes, and phone numbers. You can use RegEx in conjunction with proximity, stemming, and fuzzy searching in dtSearch.

**Using Regular Expressions**

Your case team needs to find documents containing a variety of serial numbers that all match the same pattern. The pattern consists of five letters, a hyphen, then four numbers: ABCDE-1234. Instead of searching for every possible serial number, you decide to use a regular expression. By using the regular expression you know you'll be able to find all instances of the
serial number in your document set and save your case team a lot of time. You use the following
RegEx search string: "##[a-z]{5}-[0-9]{4}" and find all instances of the serial numbers for your
case team.

7.3.1 RegEx search strings
You activate RegEx in dtSearch by starting your search string with ##.

The syntax for running a RegEx search in Relativity is as follows:
"##RegularExpression"

"##" signals to Relativity that the string following ##, and encapsulated by double quotes, should be
interpreted as RegEx. When adding double quotes to your RegEx, ensure you use straight quotes (""').
Curly quotes (""") cause the RegEx to fail. You also want to avoid using capital letters in your RegEx
because all characters in a dtSearch index are normalized to lowercase.

Once you run your RegEx search and your results return, it’s important to know the terms are not
highlighted in the document viewer. Any document that Relativity returns contains content that matches
your RegEx search. Open each document to review the content.

7.3.1.1 RegEx caveats in dtSearch
There are a few caveats to consider when using RegEx in dtSearch. Consider the following caveats before
constructing your RegEx.

- The metacharacter \s never matches a whitespace character in Relativity, because whitespace
characters don’t exist in a dtSearch index. Instead, spaces are word breaks in dtSearch.

- You cannot use capital letters when constructing a RegEx in dtSearch. Thus, if you are searching
for varying strings that all begin with NLRT, such as:
  - NLRT-0381
  - NLRT-6334
  - NLRT-9167
  - The proper Relativity RegEx is: "##nlrt-\d{4}".

7.3.2 Common dtSearch RegEx examples
The following table includes examples of dtSearch RegEx you can use to search for patterns in dtSearch
<table>
<thead>
<tr>
<th>Type</th>
<th>Regular Expression</th>
<th>Match Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bates numbers</td>
<td>&quot;##rel[0-9]{6}&quot; &quot;##rel\d{7}&quot;</td>
<td>REL0000331 REL3728948</td>
</tr>
<tr>
<td>Zip codes</td>
<td>&quot;##[a-z]{2}&quot; &quot;## [0-9]{5}&quot;</td>
<td>IL 60606 MD 21218 CA 94115</td>
</tr>
<tr>
<td>United States Phone numbers</td>
<td>&quot;##[0-9]{3}-[0-9]{4}&quot; &quot;##\d{3}-\d{4}&quot;</td>
<td>373-8837 463-9391 819-3814</td>
</tr>
<tr>
<td>United States Phone numbers with or without area codes</td>
<td>&quot;##([0-9]{3}-)?[0-9]{3}-[0-9]{4}&quot;</td>
<td>312-483-8372 463-9391 819-3814 410-396-2248 708-184-4789</td>
</tr>
<tr>
<td>Patent numbers</td>
<td>&quot;##\d.,\d{3}&quot; &quot;##[0-9],[0-9]{3},[0-9]{3}&quot;</td>
<td>5,394,233 8,303,134 9,037,193</td>
</tr>
<tr>
<td>Serial numbers</td>
<td>&quot;##[a-z]{4}-[0-9]{4} [a-z]{4}-[0-9]{4}&quot; &quot;##[a-z]{4}-\d{4} [a-z]{4}-\d{4}&quot;</td>
<td>XRFD-8324-ERWF-3231 GHSR-3413-KWEJ-8173 MPFS-1357-QEGT-9376</td>
</tr>
<tr>
<td>Dates</td>
<td>&quot;##[0-9]{1,2}/[0-9]{1,2}/[0-9]{2,4}&quot;</td>
<td>10/17/2015 3/6/98 4/25/2006 12/04/87</td>
</tr>
<tr>
<td>Type</td>
<td>Regular Expression</td>
<td>Match Results</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Email addresses</td>
<td>&quot;##([w_\w]+@([w_\w]+.([w_\w]{2,6}))&quot;</td>
<td><a href="mailto:Joe.Smith426@example.com">Joe.Smith426@example.com</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="mailto:743.MaryJane@example.com">743.MaryJane@example.com</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="mailto:Brian.23.Voltaire@example.net.uk">Brian.23.Voltaire@example.net.uk</a></td>
</tr>
<tr>
<td>URLs</td>
<td>&quot;##(https?:/?)?([da-z\w-]+.([a-z\w]{2,6})([/\w -]<em>)</em>/?)&quot;</td>
<td><a href="https://www.relativity.com">https://www.relativity.com</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="https://docs.google.com">https://docs.google.com</a></td>
</tr>
</tbody>
</table>
Optimized indexing requires some knowledge of your data. The time it takes to scrub your data before indexing will be rewarded in time saved when creating an index and returning search results. Consider the following when creating an index:

- Consider removing file types that have no searchable content, such as system or program files.
- Use a separate index for searching database files and large Excel files.
  - Even if your database has only a small number of these files, creating an index without them will improve searching speed, especially numeric range searching.
- Set up multiple dtSearch indexes, including one with a smaller document set based on one or more of the following criteria:
  - date ranges
  - custodians
  - text size (extracted or OCR text)
    - Small (< 2 MB)
    - Medium (> 2 MB and < 10 MB)
    - Large (> 10 MB and < 25 MB)
    - Very large (> 25 MB)
- Set a dtSearch index to recognize and/or ignore words, characters, and digits as necessary.
  - Noise Words (Ex: Include “sample” as a part of the noise world list for a dtSearch index containing Excel documents.)
  - Alphabet file list (Ex: Index the character “£.”)
  - Remove numbers from the alphabet file list if only searching for words – this will reduce the size of the index and will disable numeric range searching.
- Enable dtSearch indexes to automatically recognize dates, email addresses, and credit card numbers only when necessary. Enabling this setting increases build time.
- Consider using a pair of dtSearch indexes when adding new data. You can have one index updated in the background and then swap out the outdated index with the current one.
Proprietary Rights

This documentation ("Documentation") and the software to which it relates ("Software") belongs to Relativity ODA LLC and/or Relativity's third party software vendors. Relativity grants written license agreements which contain restrictions. All parties accessing the Documentation or Software must: respect proprietary rights of Relativity and third parties; comply with your organization’s license agreement, including but not limited to license restrictions on use, copying, modifications, reverse engineering, and derivative products; and refrain from any misuse or misappropriation of this Documentation or Software in whole or in part. The Software and Documentation is protected by the Copyright Act of 1976, as amended, and the Software code is protected by the Illinois Trade Secrets Act. Violations can involve substantial civil liabilities, exemplary damages, and criminal penalties, including fines and possible imprisonment.

©2019. Relativity ODA LLC. All rights reserved. Relativity® are registered trademarks of Relativity ODA LLC.