# SQL Server 2008 Thesaurus Rules of Engagement

In SQL Server, full-text queries can search for synonyms of user-specified terms through the use of a thesaurus. A SQL Server *thesaurus* defines a set of synonyms for a specific language. System administrators can define two forms of synonyms: expansion sets and replacement sets. By developing a thesaurus tailored to your full-text data, you can effectively broaden the scope of full-text queries on that data. Thesaurus matching occurs only for CONTAINS and CONTAINSTABLE queries that specify the FORMSOF THESAURUS clause and for FREETEXT and FREETEXTABLE queries.

Before full-text search queries on your server instance can look for synonyms in a given language, you must define thesaurus mappings (synonyms) for that language. Each thesaurus must be manually configured to define the following:

* Diacritics setting  
  For a given thesaurus, all search patterns are either sensitive or insensitive to diacritical marks (that is, *accent sensitive* or *accent insensitive*). For example, suppose you specify the pattern "café" to be replaced by other patterns in a full-text query. If the thesaurus is accent-insensitive, full-text search replaces the patterns "café" and "cafe". If the thesaurus is accent-sensitive, full-text search replaces only the pattern "café". By default, a thesaurus is accent-insensitive.
* Expansion set  
  An expansion set contains a group of synonyms such as "writer", "author", and "journalist" that are substituted for one another by a full-text query. Queries that contain a match for any synonym in an expansion set are expanded to include every other synonym in the expansion set.   
  For more information, see "XML Structure of an Expansion Set," later in this topic.
* Replacement set  
  A replacement set contains a text pattern to be replaced by a substitution set. For an example, see the section "XML Structure of a Replacement Set" later in this topic.

The following restrictions apply to editing a thesaurus file:

* Only system administrators can update, modify, or delete thesaurus files.
* When editing thesaurus files using text editor tools, the files must be saved in Unicode format, and Byte Order Marks must be specified.
* Thesaurus entries cannot be empty or word break to an empty string.
* Phrases in the thesaurus file must be no longer than 512 characters.
* A thesaurus must not contain any duplicate entries among the <sub> entries of expansion sets and the <pat> elements of replacement sets.

## Location of the Thesaurus Files

The default location of the thesaurus files is:

*SQL\_Server\_install\_path*\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\FTDATA\

This default location contains the following files:

* Language-specific thesaurus files   
  During setup, empty thesaurus files are installed in the above location. A separate file is provided for each supported language. A system administrator can customize these files.  
  The default file names of the thesaurus files use following format:  
  ‘ts’ + <three-letter language-abbreviation> + '.xml'   
  The name of the thesaurus file for a given language is specified in the registry in the following value HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\<instance-name>\MSSearch\<language-abbrev>.
* The global thesaurus file  
  An empty global thesaurus file, tsGlobal.xml.

You can change the location and names of a thesaurus file by changing its registry key. For each language, the location of the thesaurus file is specified in the following value in the registry:

HKLM/SOFTWARE/Microsoft/Microsoft SQL Server/<instance name>/MSSearch/Language/<language-abbreviation>/TsaurusFile

The global thesaurus file corresponds to the Neutral language with LCID 0. This value can be changed by administrators only.

## How Queries Use Thesaurus Files

A thesaurus query uses both a language-specific thesaurus and the global thesaurus. First, the query looks up the language-specific file and loads it for processing (unless it is already loaded). The query is expanded to include the language-specific synonyms specified by the expansion set and replacement set rules in the thesaurus file. These steps are then repeated for the global thesaurus. However, if a term is already part of a match in the language specific thesaurus file, the term is ineligible for matching in the global thesaurus.

## Understanding the Structure of a Thesaurus File

Each thesaurus file defines an XML container whose ID is Microsoft Search Thesaurus, and a comment, <!-- … -->, that contains a sample thesaurus. The thesaurus is defined in a <thesaurus> element that contains samples of the child elements that define the diacritics setting, expansion sets, and replacement sets, as follows:

* XML Structure of the Diacritical Setting  
  The diacritics setting of a thesaurus is specified in a single <diacritics\_sensitive> element. This element contains an integer value that controls accent sensitivity, as follows:

|  |  |  |
| --- | --- | --- |
| **Diacritics Setting** | **Value** | **XML** |
| Accent insensitive | 0 | <diacritics\_sensitive>0</diacritics\_sensitive> |
| Accent sensitive | 1 | <diacritics\_sensitive>1</diacritics\_sensitive> |

|  |
| --- |
| **ms142491.note(en-us,SQL.100).gifNote:** |
| This setting can only be applied one time in the file, and it applies to all search patterns in the file. This setting cannot be specified for individual patterns. |

* XML Structure of an Expansion Set  
  Each expansion set is enclosed within an <expansion> element. Within this element, you specify one or more substitutions in a <sub> element. In the expansion set, you can specify a group of substitutions that are synonyms of each other.  
  For example, you can edit the expansion section to treat the substitutions "writer", "author", and "journalist" as synonyms. full-text search queries that contain matches in one substitution are expanded to include all other substitutions specified in the expansion set. Therefore, in the preceding example, when you issue a FORMS OF THESAURUS or a FREETEXT query for the word "author", full-text search also returns search results containing the words "writer" and "journalist".  
  This is what the expansion set section would look like for the above example:

[[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl19other');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl19other');)

<expansion>

<sub>writer</sub>

<sub>author</sub>

<sub>journalist</sub>

</expansion>

* XML Structure of a Replacement Set  
  Each replacement set is enclosed within a <replacement> element. Within this element you can specify one or more patterns in a <pat> element and zero or more substitutions in <sub> elements, one per synonym. You can specify a pattern to be replaced by a substitution set. Patterns and substitutions can contain a word, or a sequence of words. If there is no substitution specified for a pattern, it has the effect of removing the pattern from the user query.  
  For example, suppose you want queries for "W2K", the pattern, to be replaced by "Windows 2000" or "XP", the substitutions. If you run a full-text query for "W2K", full-text search only returns search results containing "Windows 2000" or "XP". It does not return results containing "W2K". This is because the pattern "W2K" has been "replaced" by the patterns "Windows 2000" and "XP".  
  This is what the replacement set section would look like for the above example:

[[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl20other');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl20other');)

<replacement>

<pat>W2K</pat>

<sub>Windows 2000</sub>

<sub>XP</sub>

</replacement>

If you have two replacement sets with similar patterns being matched, the longer of the two takes precedence. For example, if you run a FORMS OF THESAURUS query for "Internet Explorer online community" and you have the following replacement sets, the "Internet Explorer" replacement set takes precedence over the "Internet" replacement set. The query will therefore be processed as "IE online community" or "IE 5 online community".

[[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl21other');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl21other');)

<replacement>

<pat>Internet</pat>

<sub>intranet</sub>

</replacement>

and

[[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl22other');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl22other');)

<replacement>

<pat>Internet Explorer</pat>

<sub>IE</sub>

<sub>IE 5</sub>

</replacement>

## To load an updated thesaurus file

Syntax

sys.sp\_fulltext\_load\_thesaurus\_file lcid [ , @loadOnlyIfNotLoaded = action ]

http://i.msdn.microsoft.com/Global/Images/clear.gif Arguments

lcid

Integer mapping the locale identifier (LCID) of the language for which you want to load the thesaurus XML definition. To obtain the LCIDs of languages that are available on a server instance, use the [sys.fulltext\_languages (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms176076.aspx) catalog view.

**@loadOnlyIfNotLoaded** = action

Specifies whether the thesaurus file is loaded into the internal thesaurus tables even if it has already been loaded. action is one of:

|  |  |
| --- | --- |
| **Value** | **Definition** |
| 0 | Load the thesaurus file regardless of whether it is already loaded. This is the default behavior of **sp\_fulltext\_load\_thesaurus\_file**. |
| 1 | Load the thesaurus file only if it is not yet loaded. |

http://i.msdn.microsoft.com/Global/Images/clear.gif Return Code Values

None

http://i.msdn.microsoft.com/Global/Images/clear.gif Result Sets

None

http://i.msdn.microsoft.com/Global/Images/clear.gif Remarks

Thesaurus files are automatically loaded by full-text queries that use the thesaurus. To avoid this first-time performance impact on full-text queries, we recommend that you execute **sp\_fulltext\_load\_thesaurus\_file**.

|  |
| --- |
| **Cc280598.note(en-us,SQL.100).gifNote:** |
| When compiling a full-text query that requires a thesaurus for a given language, the Full-Text Engine loads the thesaurus file only if it has not yet been loaded. |

Use [sp\_fulltext\_service](http://msdn.microsoft.com/en-us/library/ms175058.aspx) **'update\_languages'** to update the list of languages registered with full-text search.

http://i.msdn.microsoft.com/Global/Images/clear.gif Permissions

Only members of the **sysadmin** fixed server role or the system administrator can execute the **sp\_fulltext\_load\_thesaurus\_file** stored procedure.

Only system administrators can update, modify, or delete thesaurus files.

http://i.msdn.microsoft.com/Global/Images/clear.gif Examples

### Example A: Load a thesaurus file even if it is already loaded

The following example parses and loads the English thesaurus file:

[[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl27other');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl27other');)

EXEC sys.sp\_fulltext\_load\_thesaurus\_file 1033;

GO

### Example B: Load a thesaurus file only if it is not yet loaded

The following example parses and loads the Arabic thesaurus file, unless it is already loaded:

[[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl28other');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl28other');)

EXEC sys.sp\_fulltext\_load\_thesaurus\_file 1025, @loadOnlyIfNotLoaded = 1;

GO

## To view the tokenization result of a word breaker, thesaurus, and stoplist combination

sys.dm\_fts\_parser (Transact-SQL)

Returns the final tokenization result after applying a given [word breaker](http://msdn.microsoft.com/en-us/library/ms142509.aspx), [thesaurus](http://msdn.microsoft.com/en-us/library/ms142491.aspx), and [stoplist](http://msdn.microsoft.com/en-us/library/ms142551.aspx) combination to a query string input. The tokenization result is equivalent to the output of the Full-Text Engine for the specified query string.

**sys.dm\_fts\_parser** is a dynamic management function.

http://i.msdn.microsoft.com/Global/Images/clear.gif Syntax

sys.dm\_fts\_parser('query\_string', lcid, stoplist\_id, accent\_sensitivity)

http://i.msdn.microsoft.com/Global/Images/clear.gif Arguments

query\_string

The query that you want to parse. query\_string can be a string chain that [CONTAINS](http://msdn.microsoft.com/en-us/library/ms187787.aspx) syntax support. For example, you can include inflectional forms, a thesaurus, and logical operators.

lcid

Locale identifier (LCID) of the word breaker to be used for parsing query\_string.

stoplist\_id

ID of the stoplist, if any, to be used by the word breaker identified by lcid. stoplist\_id is **int**. If you specify 'NULL', no stoplist is used. If you specify 0, the system STOPLIST is used.

A stoplist ID is unique within a database. To obtain the stoplist ID for a full-text index on a given table use the [sys.fulltext\_indexes](http://msdn.microsoft.com/en-us/library/ms186903.aspx) catalog view.

accent\_sensitivity

Boolean value that controls whether full-text search is sensitive or insensitive to diacritics. accent\_sensitivity is **bit**, with one of the following values:

|  |  |
| --- | --- |
| **Value** | **Accent sensitivity is…** |
| 0 | Insensitive  Words such as "café" and "cafe" are treated identically. |
| 1 | Sensitive  Words such as "café" and "cafe" are treated differently. |

|  |
| --- |
| **Cc280463.note(en-us,SQL.100).gifNote:** |
| To view the current setting of this value for a full-text catalog, run the following Transact-SQL statement: SELECT fulltextcatalogproperty('catalog\_name', 'AccentSensitivity');. |

http://i.msdn.microsoft.com/Global/Images/clear.gif Table Returned

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| **keyword** | **varbinary(128)** | The hexadecimal representation of a given keyword returned by a word breaker. This representation is used to store the keyword in the full-text index. This value is not human-readable, but it is useful for relating a given keyword to output returned by other dynamic management views that return the content of a full-text index, such as [sys.dm\_fts\_index\_keywords](http://msdn.microsoft.com/en-us/library/cc280900.aspx) and [sys.dm\_fts\_index\_keywords\_by\_document](http://msdn.microsoft.com/en-us/library/cc280607.aspx).  Cc280463.note(en-us,SQL.100).gifNote:  OxFF represents the special character that indicates the end of a file or dataset. |
| **group\_id** | **int** | Contain an integer value that is useful for differentiating the logical group from which a given term was generated. For example, 'Server AND DB OR FORMSOF(THESAURUS, DB)"' produces the following **group\_id** values in English:  group\_id display\_term  1 Server  2 DB  3 DB |
| **phrase\_id** | **int** | Contains an integer value that is useful for differentiating the cases in which alternative forms of compound words, such as full-text, are issued by the word breaker. Sometimes, with presence of compound words ('multi-million'), alternative forms are issued by the word breaker. These alternative forms (phrases) need to be differentiated sometimes.  For example, 'multi-million' produces the following **phrase\_id** values in English:  phrase\_id display\_term  1 multi  1 million  2 multimillion |
| **occurrence** | **int** | Indicates the order of each term in the parsing result. For example, for the phrase "SQL Server query processor" **occurrence** would contain the following **occurrence** values for the terms in the phrase, in English:  occurrence display\_term  1 SQL  2 Server  3 query  4 processor |
| **special\_term** | **nvarchar(8000)** | Contains information about the characteristics of the term that is being issued by the word breaker, one of:  Exact match  Noise word  End of Sentence  End of paragraph  End of Chapter |
| **display\_term** | **nvarchar(8000)** | Contains the human-readable form of the keyword. As with the functions designed to access the content of the full-text index, this displayed term might not be identical to the original term due to the denormalization limitation. However, it should be precise enough to help you identify it from the original input. |
| **expansion\_type** | **int** | Contains information about the nature of the expansion of a given term, one of:  0 =Single word case  2=Inflectional expansion  4=Thesaurus expansion/replacement  For example, consider a case in which the thesaurus defines run as an expansion of jog:  <expansion>  <sub>run</sub>  <sub>jog</sub>  </expansion>  The term FORMSOF (FREETEXT, run) generates the following output:  run with **expansion\_type**=0  runs with **expansion\_type**=2  running with **expansion\_type**=2  ran with **expansion\_type**=2  jog with **expansion\_type**=4 |
| **source\_term** | **nvarchar(8000)** | The term or phrase from which a given term was generated or parsed. For example, a query on the '"word breakers" AND stemmers' produces the following **source\_term** values in English:  source\_term display\_term  word breakers word  word breakers breakers  stemmers stemmers |

http://i.msdn.microsoft.com/Global/Images/clear.gif Remarks

**sys.dm\_fts\_parser** supports the syntax and features of full-text predicates, such as [CONTAINS](http://msdn.microsoft.com/en-us/library/ms187787.aspx) and [FREETEXT](http://msdn.microsoft.com/en-us/library/ms176078.aspx), and functions, such as [CONTAINSTABLE](http://msdn.microsoft.com/en-us/library/ms189760.aspx) and [FREETEXTTABLE](http://msdn.microsoft.com/en-us/library/ms177652.aspx).

### Using Unicode for Parsing Special Characters

When you parse a query string, **sys.dm\_fts\_parser** uses the [collation](http://msdn.microsoft.com/en-us/library/ms187582.aspx) of the database to which you are connected, unless you specify the query string as [Unicode](http://msdn.microsoft.com/en-us/library/ms187828.aspx). Therefore, for a non-Unicode string that contains special characters, such as ü or ç, the output might be unexpected, depending on the collation of the database. To process a query string independently of the database collation, prefix the string with N, that is, **N'**query\_string**'**.

For more information, see "C. Displaying the Output of a String that Contains Special Characters," later in this topic.

### When to Use sys.dm\_fts\_parser

**sys.dm\_fts\_parser** can be very powerful for debugging purposes. Some major usage scenarios include:

* To understand how a given word breaker treats a given input  
  When a query returns unexpected results, a likely cause is the way that the word breaker is parsing and breaking the data. By using **sys.dm\_fts\_parser**, you discover the result that a word breaker passes to the full-text index. In addition, you can see which terms are stopwords, which are not searched in the full-text index. Whether a term is a stopword for a given language depends on whether it is in the stoplist specified by the stoplist\_id value that is declared in the function.   
  Note as well the accent sensitivity flag, which will allow the user to see how the word breaker will parse the input having in mind its accent sensitivity information.
* To understand how the stemmer works on a given input  
  You can find out how the word breaker and the stemmer parse a query term and its stemming forms, by specifying a CONTAINS or CONTAINSTABLE query containing the following FORMSOF clause:

[[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl51other');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl51other');)

FORMSOF( INFLECTIONAL, query\_term )

The results tell you what terms are being passed to the full-text index.

* To understand how the thesaurus expands or replaces all or part of the input  
  You can also specify:

[[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl52other');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl52other');)

FORMSOF( THESAURUS, query\_term )

The results of this query show how the word breaker and thesaurus interact for the query term. you can see the expansion or replacements from the thesaurus and identify the resulting query that is actually being issued against the full-text index.   
  
Note that if the user issues:

[[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl53other');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl53other');)

FORMSOF( FREETEXT, query\_term )

The inflectional and Thesaurus capabilities will take place automatically.

In addition to the preceding usage scenarios, **sys.dm\_fts\_parser** can help significantly to understand and troubleshoot many other issues with full-text query.

http://i.msdn.microsoft.com/Global/Images/clear.gif Permissions

Requires membership in the **sysadmin** fixed server role and access rights to the specified stoplist.

http://i.msdn.microsoft.com/Global/Images/clear.gif Examples

### A. Displaying the output of a given word breaker for a keyword or phrase

The following example returns the output from using the English word breaker, whose LCID is 1033, and no stoplist on the following query string:

The Microsoft business analysis

Accent sensitivity is disabled.

[[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl61other');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl61other');)

SELECT \* FROM sys.dm\_fts\_parser (' "The Microsoft business analysis" ', 1033, 0, 0)

### B. Displaying the output of a given word breaker in the context of stoplist filtering

The following example returns the output from using the English word breaker, whose LCID is 1033, and an English stoplist, whose ID is 77, on the following query string:

"The Microsoft business analysis" OR "MS revenue"

Accent sensitivity is disabled.

[[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl62other');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl62other');)

SELECT \* FROM sys.dm\_fts\_parser (' "The Microsoft business analysis" OR " MS revenue" ', 1033, 77, 0)

### C. Displaying the Output of a String that Contains Special Characters

The following example uses Unicode to parse the following French string:

**français**

The example specifies the LCID for the French language, **1036**, and the ID of a user-defined stoplist, **5**. Accent sensitivity is enabled.

[[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl63other');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl63other');)

SELECT \* FROM sys.dm\_fts\_parser(N'français', 1036, 5, 1);

## Stopwords and Stoplists

To prevent a full-text index from becoming bloated, SQL Server has a mechanism that discards commonly occurring strings that do not help the search. These discarded strings are called stopwords. During index creation, the Full-Text Engine omits stopwords from the full-text index. This means that full-text queries will not search on stopwords.

|  |
| --- |
| **ms142551.note(en-us,SQL.100).gifImportant:** |
| SQL Server 2005 noise words have been replaced by stopwords. When a database is upgraded to SQL Server 2008 from a previous release, the noise-word files are no longer used in SQL Server 2008. However, the noise-word files are stored in the FTDATA\ FTNoiseThesaurusBak folder, and you can use them later when updating or building the corresponding SQL Server 2008 stoplists. For information about upgrading noise-word files to stoplists, see [Full-Text Search Upgrade](http://msdn.microsoft.com/en-us/library/ms142490.aspx). |

A stopword can be a word with meaning in a specific language, or it can be a token that does not have linguistic meaning. For example, in the English language, words such as "a," "and," "is," and "the" are left out of the full-text index since they are known to be useless to a search.

Although it ignores the inclusion of stopwords, the full-text index does take into account their position. For example, consider the phrase, "Instructions are applicable to these Adventure Works Cycles models". The following table depicts the position of the words in the phrase:

|  |  |
| --- | --- |
| **Word** | **Position** |
| Instructions | 1 |
| are | 2 |
| applicable | 3 |
| to | 4 |
| these | 5 |
| Adventure | 6 |
| Works | 7 |
| Cycles | 8 |
| models | 9 |

The stopwords "are", "to", and "these" that are in positions 2, 4, and 5 are left out of the full-text index. However, their positional information is maintained, thereby leaving the position of the other words in the phrase unaffected.

http://i.msdn.microsoft.com/Global/Images/clear.gif Stoplists

In SQL Server 2008, stopwords are managed in databases using objects called stoplists. A stoplist is a list of stopwords that, when associated with a full-text index, is applied to full-text queries on that index.

### Creating a Stoplist

You can create a stoplist in any of the following ways:

* Using the system-supplied stoplist in the database. SQL Server ships with a system stoplist that contains the most commonly used stopwords for each supported language, that is for every language that is associated with given word breakers by default. The system stoplist contains common stopwords for all supported languages. You can copy the system stoplist, and customize your copy by adding and removing stopwords.  
  The system stoplist is installed in the [Resource](http://msdn.microsoft.com/en-us/library/ms190940.aspx)database.
* Creating your own stoplist, and then adding stopwords to it for any language that you specify. You can also drop stopwords from your stoplist when necessary.
* Using an existing custom stoplist from any other database in the current server instance and then adding and dropping stopwords as necessary.
* CREATE FULLTEXT STOPLIST (Transact-SQL)
* Creates a new full-text stoplist in the current database.
* In SQL Server 2008 and later versions, stopwords are managed in databases by using objects called stoplists. A stoplist is a list of stopwords that, when associated with a full-text index, is applied to full-text queries on that index. For more information, see [Stopwords and Stoplists](http://msdn.microsoft.com/en-us/library/ms142551.aspx).

|  |
| --- |
| **Cc280405.note(en-us,SQL.100).gifImportant:** |
| CREATE FULLTEXT STOPLIST, ALTER FULLTEXT STOPLIST, and DROP FULLTEXT STOPLIST are supported only under compatibility level 100. Under compatibility levels 80 and 90, these statements are not supported. However, under all compatibility levels the system stoplist is automatically associated with new full-text indexes. |

* Topic link icon[Transact-SQL Syntax Conventions](http://msdn.microsoft.com/en-us/library/ms177563.aspx)
* http://i.msdn.microsoft.com/Global/Images/clear.gif Syntax
* CREATE FULLTEXT STOPLIST stoplist\_name
* [ FROM { [ database\_name. ] source\_stoplist\_name } | SYSTEM STOPLIST ]
* [ AUTHORIZATION owner\_name ]
* ;
* http://i.msdn.microsoft.com/Global/Images/clear.gif Arguments
* stoplist\_name
* Is the name of the stoplist. stoplist\_name can be a maximum of 128 characters. stoplist\_name must be unique among all stoplists in the current database, and conform to the rules for [identifiers](http://msdn.microsoft.com/en-us/library/ms175874.aspx).
* stoplist\_name will be used when the full-text index is created.
* database\_name
* Is the name of the database where the stoplist specified by source\_stoplist\_name is located. If not specified, database\_name defaults to the current database.
* source\_stoplist\_name
* Specifies that the new stoplist is created by copying an existing stoplist. If source\_stoplist\_name does not exist, or the database user does not have correct permissions, CREATE FULLTEXT STOPLIST fails with an error. If any languages specified in the stop words of the source stoplist are not registered in the current database, CREATE FULLTEXT STOPLIST succeeds, but warning(s) are returned and the corresponding stop words are not added.
* SYSTEM STOPLIST
* Specifies that the new stoplist is created from the stoplist that exists by default in the [Resource database](http://msdn.microsoft.com/en-us/library/ms190940.aspx).
* AUTHORIZATION owner\_name
* Specifies the name of a database principal to own of the stoplist. owner\_name must either be the name of a principal of which the current user is a member, or the current user must have IMPERSONATE permission on owner\_name. If not specified, ownership is given to the current user.
* http://i.msdn.microsoft.com/Global/Images/clear.gif Remarks
* The creator of a stoplist is its owner.
* http://i.msdn.microsoft.com/Global/Images/clear.gif Permissions
* To create a STOPLIST requires CREATE FULLTEXT CATALOG permissions. The stoplist owner can grant CONTROL permission explicitly on a stoplist to allow users to add and remove words and to drop the stoplist.

|  |
| --- |
| **Cc280405.note(en-us,SQL.100).gifNote:** |
| Using a stoplist with a full-text index requires REFERENCE permission. |

* http://i.msdn.microsoft.com/Global/Images/clear.gif Examples

### A. Creating a new full-text stoplist

* The following example creates a new full-text stoplist named myStoplist.
* [[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl26other');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl26other');)
* CREATE FULLTEXT STOPLIST myStoplist;
* GO

### B. Copying a full-text stoplist from an existing full-text stoplist

* The following example creates a new full-text stoplist named myStoplist2 by copying an existing AdventureWorks stoplist named Customers.otherStoplist.
* [[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl27other');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl27other');)
* CREATE FULLTEXT STOPLIST myStoplist2 FROM AdventureWorks.otherStoplist;
* GO

### C. Copying a full-text stoplist from the system full-text stoplist

* The following example creates a new full-text stoplist named myStoplist3 by copying from the system stoplist.
* [[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl28other');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl28other');)
* CREATE FULLTEXT STOPLIST myStoplist3 FROM SYSTEM STOPLIST;
* GO

For full details on stop lists see <http://msdn.microsoft.com/en-us/library/cc280405.aspx>.

## Overview of CONTAINS and FREETEXT Searches

Querying SQL Server Using Full-Text Search

SQL Server provides a suite of full-text predicates (CONTAINS and FREETEXT) and rowset-valued functions (CONTAINSTABLE and FREETEXTTABLE) for writing full-text queries. This section introduces these predicates and functions and describes the different types of searches you can perform using them. It also discusses how to tune and optimize full-text queries.

http://i.msdn.microsoft.com/Global/Images/clear.gif In This Section

[Full-Text Predicates and Functions Overview](http://msdn.microsoft.com/en-us/library/ms142583.aspx)

Introduces and compares the full-text predicates (CONTAINS and FREETEXT) and functions (CONTAINSTABLE and FREETEXTTABLE).

[Supported Forms of Query Terms (Full-Text Search)](http://msdn.microsoft.com/en-us/library/cc879300.aspx)

Contains information about the forms of queries by full-text search and the support provided for each form by the full-text predicates and rowset-valued functions.

[Limiting Ranked Result Sets (Full-Text Search)](http://msdn.microsoft.com/en-us/library/cc879245.aspx)

Discusses the optional top\_n\_by\_rank parameter to return only a subset of rows that satisfy a query using the CONTAINSTABLE or FREETEXTABLE function. Also, contains information about how full-text search generates the rank values returned by a query

[Querying varbinary(max) and xml Columns (Full-Text Search)](http://msdn.microsoft.com/en-us/library/ms142531.aspx)

Discusses the use of CONTAINS and FREETEXT predicates to search full-text indexes on columns of **varbinary**, **varbinary(max), xml,** or **image** data.

[Querying Multiple Columns (Full-Text Search)](http://msdn.microsoft.com/en-us/library/ms142488.aspx)

Discusses using the CONTAINS predicate to query multiple columns.

[Querying Linked Servers (Full-Text Search)](http://msdn.microsoft.com/en-us/library/ms142529.aspx)

Discusses the requirements for using four-part names in CONTAINS or FREETEXT full-text predicates to execute queries against linked servers.

[Performance Tuning and Optimization of Full-Text Queries](http://msdn.microsoft.com/en-us/library/cc879244.aspx)

Discusses recommendations to increase full-text query performance.

http://i.msdn.microsoft.com/Global/Images/clear.gif See Also

#### Concepts

[Full-Text Search Overview](http://msdn.microsoft.com/en-us/library/ms142547.aspx)

#### Other Resources

[CONTAINS (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms187787.aspx)  
[CONTAINSTABLE (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms189760.aspx)  
[FREETEXT (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms176078.aspx)  
[FREETEXTTABLE (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms177652.aspx)  
[Full-Text Search Administrator InfoCenter](http://msdn.microsoft.com/en-us/library/ms142496.aspx)  
[Full-Text Search Developer InfoCenter](http://msdn.microsoft.com/en-us/library/ms142519.aspx)

#### Help and Information

[Getting SQL Server 2008 Assistance](http://msdn.microsoft.com/en-us/library/ms166016.aspx)