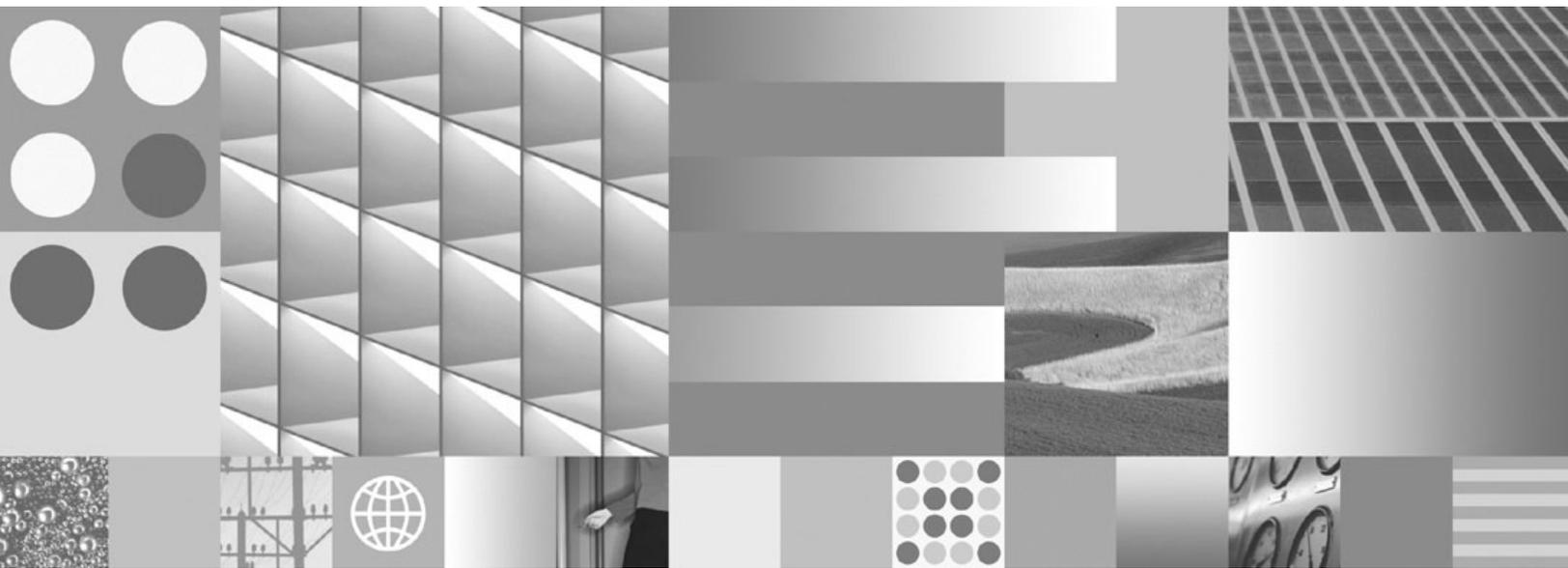


DB2
for Linux, UNIX, and Windows



Version 9 Release 7



DB2 Text Search Guide
Updated July, 2012

Note

Before using this information and the product it supports, read the general information under Appendix E, "Notices," on page 201.

Edition Notice

This document contains proprietary information of IBM. It is provided under a license agreement and is protected by copyright law. The information contained in this publication does not include any product warranties, and any statements provided in this manual should not be interpreted as such.

You can order IBM publications online or through your local IBM representative.

- To order publications online, go to the IBM Publications Center at www.ibm.com/shop/publications/order
- To find your local IBM representative, go to the IBM Directory of Worldwide Contacts at www.ibm.com/planetwide

To order DB2 publications from DB2 Marketing and Sales in the United States or Canada, call 1-800-IBM-4YOU (426-4968).

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright IBM Corporation 2008, 2012.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

| | | | |
|--|-----------|--|-----------|
| Chapter 1. DB2 Text Search | 1 | Starting the DB2 Text Search instance service | 49 |
| Chapter 2. DB2 Text Search overview | 3 | Enabling a database for DB2 Text Search | 49 |
| Key features and concepts | 3 | Installing DB2 Accessories Suite for DB2 Text Search Version 9.7 FP3 or earlier | 50 |
| Scenario: Indexing and searching | 3 | Installing DB2 Accessories Suite for DB2 Text Search Version 9.7 FP4 or later | 52 |
| Text search index creation, updates, and changes | 6 | Uninstalling the DB2 Accessories Suite for DB2 Text Search | 53 |
| Linguistic processing for DB2 Text Search. | 7 | Setting-up DB2 Text Search for rich text document support. | 54 |
| Command-line tools for DB2 Text Search | 8 | Enabling DB2 Text Search for rich text document support. | 55 |
| Issuing text search commands | 9 | Stopping the DB2 Text Search instance service. | 56 |
| Rich text document support | 9 | Disabling a database for DB2 Text Search | 56 |
| Text search solution planning | 10 | Deleting orphaned text search collections | 57 |
| Security considerations for DB2 Text Search | 10 | Disabling DB2 Text Search for rich text document support. | 59 |
| User roles | 10 | Text search index creation | 59 |
| DB2 Text Search capacity planning. | 11 | Creating a text search index | 60 |
| Document formats supported for DB2 Text Search | 22 | Improving the performance of text search indexes | 63 |
| Supported data types | 23 | Text search index maintenance | 63 |
| Conversion of unsupported formats and data types | 23 | Updating a text search index | 64 |
| Supported languages and code pages. | 23 | Clearing text search index events | 65 |
| Document size considerations | 24 | Altering a text search index | 65 |
| Chapter 3. Installing and configuring DB2 Text Search | 27 | Viewing text search index status | 66 |
| Installing and configuring DB2 Text Search with the DB2 Setup Wizard | 28 | Dropping a text search index | 66 |
| Installing DB2 servers (Windows) | 29 | Backing up and restoring text search indexes | 67 |
| Installing DB2 servers using the DB2 Setup wizard (Linux and UNIX) | 32 | Synonym dictionaries for DB2 Text Search | 68 |
| Installing and configuring DB2 Text Search with a response file | 35 | Adding a synonym dictionary for DB2 Text Search | 68 |
| Installing a DB2 product using a response file (Windows). | 36 | Removing a synonym dictionary for DB2 Text Search | 69 |
| Installing a DB2 database product using a response file (Linux and UNIX). | 37 | Chapter 6. Searching with text search indexes | 71 |
| Installing DB2 Text Search using db2_install (Linux and UNIX) | 38 | Search functions for DB2 Text Search | 71 |
| Installing a DB2 database product using the db2_install or doce_install command (Linux and UNIX) | 38 | DB2 Text Search query planning | 73 |
| Configuring DB2 Text Search | 41 | DB2 Text Search arguments | 73 |
| Configuring DB2 Text Search using the Configuration Tool | 42 | DB2 Text Search multiple predicates | 74 |
| Updating the services file on the server for TCP/IP communications | 44 | DB2 Text Search SCORE function | 74 |
| | | DB2 Text Search RESULTLIMIT function. | 75 |
| Chapter 4. Upgrading text search | 45 | Searching in text search indexes | 75 |
| Upgrading DB2 Text Search for administrator or root install. | 45 | Searching text search indexes using SCORE | 76 |
| Upgrading DB2 Text Search for non-root install (Linux and UNIX) | 46 | Enhancing performance for full-text queries | 77 |
| Chapter 5. Administering and maintaining text search indexes. | 49 | Text search argument syntax. | 77 |
| | | Searching XML documents using DB2 Text Search | 80 |
| | | XML search configuration for DB2 Text Search. | 82 |
| | | Search syntax for XML documents. | 83 |
| | | Chapter 7. SQL and XML built-in search functions. | 89 |
| | | CONTAINS function | 89 |
| | | SCORE function | 91 |
| | | xmlcolumn-contains function | 93 |

Chapter 8. Administration commands for DB2 Text Search 99

db2ts START FOR TEXT 100
db2ts STOP FOR TEXT 100
db2ts CLEANUP FOR TEXT 101
db2ts ENABLE DATABASE FOR TEXT 102
db2ts DISABLE DATABASE FOR TEXT 103
db2ts CLEAR COMMAND LOCKS 105
db2ts CREATE INDEX 107
db2ts DROP INDEX 112
db2ts ALTER INDEX 114
db2ts UPDATE INDEX 118
db2ts CLEAR EVENTS FOR INDEX 121
db2ts HELP 123

Chapter 9. DB2 Text Search routines 127

SYSTS_ADMIN_CMD stored procedure - Run text search administration commands 127
SYSTS_ENABLE procedure - Enable current database for text search 128
SYSTS_DISABLE procedure - Disable current database for text search 130
SYSTS_CREATE procedure - Create a text search index on a column 132
SYSTS_DROP procedure - Drop a text search index 139
SYSTS_ALTER procedure - Change the update characteristics of an index 141
SYSTS_UPDATE procedure - Update the text search index. 146
SYSTS_CLEAR_EVENTS procedure - Delete indexing events from an index's event table . . . 148
SYSTS_CLEAR_COMMANDLOCKS procedure - Remove command locks for text search indexes . 151

Chapter 10. Text search administrative views 155

SYSBMTS.TSDEFAULTS view. 155
SYSBMTS.TSLOCKS view 156
SYSBMTS.TSCOLLECTIONNAMES view. . . . 156
SYSBMTS.TSINDEXES view 156
SYSBMTS.TSCONFIGURATION view 157
SYSBMTS.TSEVENT view 158
SYSBMTS.TSSTAGING view 159

Chapter 11. Text Search tools 161

Configuration Tool for DB2 Text Search. 161
Administration tool for DB2 Text Search 163
Synonym Tool for DB2 Text Search 165
Log Formatter Tool for DB2 Text Search 166

Appendix A. DB2 Text Search and Net Search Extender comparison 169

Migration to DB2 Text Search 171

Appendix B. Locales supported for DB2 Text Search 173

Appendix C. DB2 commands. 175

db2iupgrade - Upgrade instance 175
db2icrt - Create instance. 178
db2idrop - Remove instance 182
db2iupdt - Update instances 184

Appendix D. Overview of the DB2 technical information 189

DB2 technical library in hardcopy or PDF format 189
Ordering printed DB2 books 192
Displaying SQL state help from the command line processor. 193
Accessing different versions of the DB2 Information Center 193
Displaying topics in your preferred language in the DB2 Information Center 194
Updating the DB2 Information Center installed on your computer or intranet server. 194
Manually updating the DB2 Information Center installed on your computer or intranet server . . 196
DB2 tutorials 197
DB2 troubleshooting information. 198
Terms and Conditions 198

Appendix E. Notices 201

Index 205

Chapter 1. DB2 Text Search

DB2® Text Search allows you to issue SQL and XQuery statements to perform text search queries on data stored in a DB2 database.

DB2 Text Search provides extensive capabilities for searching data in text columns stored in a DB2 table. The search system provides fast query response times and a consolidated, ranked result set that enables you to quickly and easily locate the information that you need. By incorporating the functions of DB2 Text Search in your SQL and XQuery statements, you can create powerful and versatile text-retrieval programs.

DB2 Text Search achieves high performance and scalability by employing data streaming to avoid high resource consumption during search.

DB2 Text Search is supported on the following platforms:

- AIX® (64-bit) platforms
- Linux x64 (64-bit) platforms
- HP-UX on Itanium-based HP Integrity Series (64-bit) platforms
- Solaris on AMD (64-bit) platforms
- Solaris on UltraSPARC (64-bit) platforms
- Windows on x86 (32-bit) platforms
- Windows on x64 (64-bit) platforms
- Linux x86 (32-bit) platforms are supported for development or test purposes only.

Note: DB2 Text Search does not support Microsoft Failover Clustering.

DB2 Text Search includes the following key features:

Tight integration with DB2 Database for Linux, UNIX, and Windows

- A stored procedure interface for administration commands
- Installation and configuration performed by the DB2 installer
- Invisible authentication
- SQL codes for error handling

Document indexing

- Fast indexing of large amounts of data
- pureXML® support
- Multiple document format support
- Incremental and asynchronous index updating

Advanced search technology

- SQL, SQL/XML, and XQuery support
- The CONTAINS and SCORE SQL functions
- Built-in SQL functionality combined with the DB2 Optimizer
- The xmlcolumn-contains XML function
- XML filtering
- Linguistic processing in all supported languages

- Weight, wildcard, and optional term support
- Synonym dictionary support

The hardware requirements for DB2 Text Search are as follows:

Hardware requirements

- Two dual-core 2.66 GHz processors
- 4 GB RAM
- Invisible authentication
- Enough disk space to hold four times the original text size multiplied by the number of text search indexes. For example, if a text search server supports five indexes each with 1 million rows of 1 KB text size, the required disk space needed is about 20 GB.

Chapter 2. DB2 Text Search overview

DB2 Text Search allows you to search text columns stored in a DB2 table. By enabling text search support, you can use the `CONTAINS`, `SCORE`, and `xmlcolumn-contains` functions, which are built into the DB2 engine, to search text search indexes based on the search arguments that you specify.

DB2 Text Search works by collecting data from diverse sources and indexing it for subsequent fast retrieval. Furthermore, the search engine uses linguistic analysis to ensure that it returns only relevant search query results.

As shown in Figure 1, you install the DB2 Text Search server and DB2 servers on the same system, but the DB2 Text Search server runs in its own Java Virtual Machine (JVM). You explicitly start and stop the DB2 Text Search services after starting the DB2 instance. You can install DB2 Text Search and DB2 Net Search Extender on the same DB2 install copy.

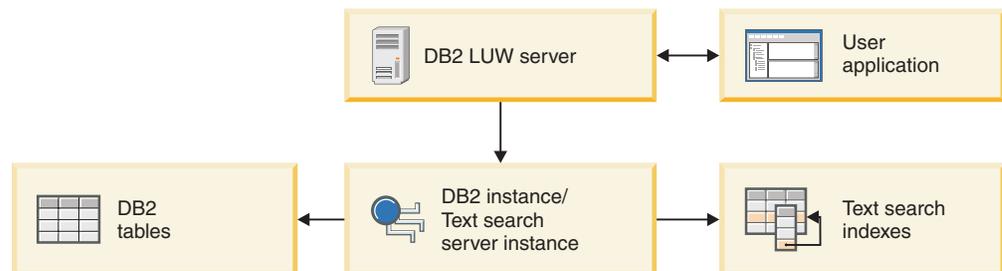


Figure 1. Deployment diagram for the DB2 Text Search server

Starting with Version 9.7 Fix Pack 3 and later fix packs, DB2 DB2 Text Search and Net Search Extender text indexes can coexist on the same table column. You can now switch between either DB2 Text Search or Net Search Extender text indexes by simply toggling the index status.

DB2 Text Search does not have a graphical user interface. Instead, command-line tools are available for tasks such as configuring and administering the DB2 Text Search server, creating a synonym dictionary for a collection, and diagnosing problems. In addition, you can use a stored-procedure interface for some common administrative tasks.

Key features and concepts

Scenario: Indexing and searching

After you have installed and configured DB2 Text Search, there are four steps that you must take before performing searches.

1. Start the DB2 Text Search instance services.

The process that maintains the update schedule of text search indexes is started when you issue the `db2ts START FOR TEXT` command. The text search instance service updates the text search index automatically at the specified times.

2. Prepare the database for use by DB2 Text Search.

You must enable a database for DB2 Text Search only once. Note that you cannot enable Net Search Extender for a database once it has been enabled for DB2 Text Search.

3. Create a text search index on a column that contains, or will contain, text that you want to search.
4. Populate the text search index. This adds data to the empty, newly created text search index.

After a text search index contains data, you can search the index using an SQL statement and can search with XQuery if the index contains XML data.

As Figure 2 shows, you should update existing text search indexes, either manually or automatically, to reflect changes to the text column that the index is associated with.

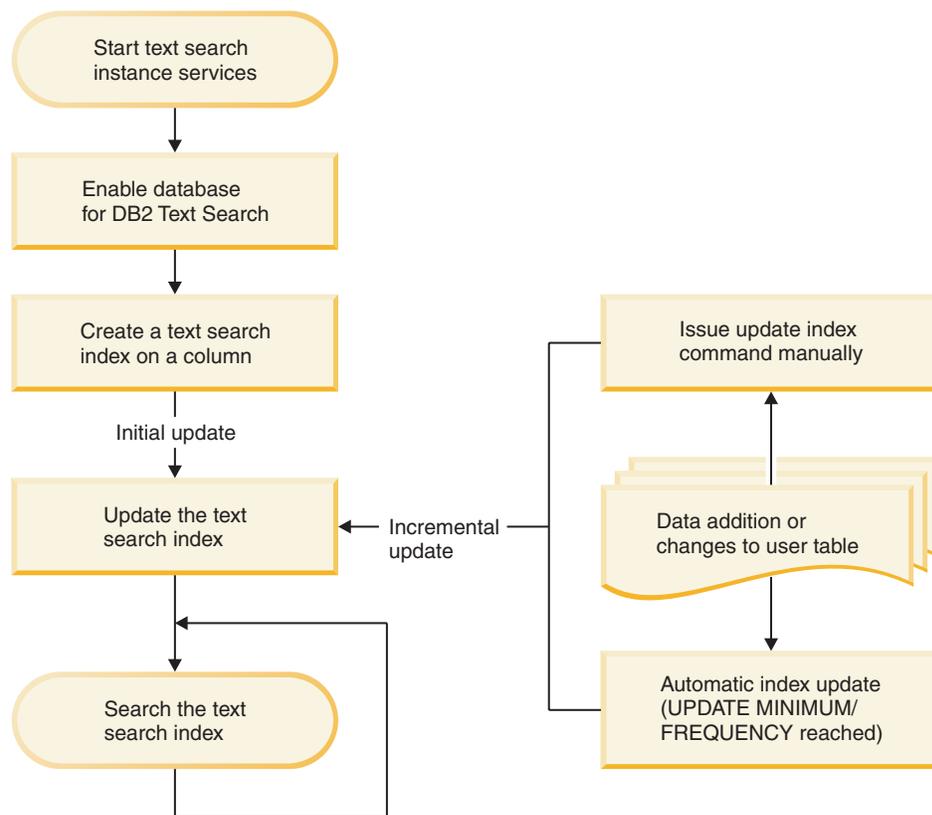


Figure 2. Setting up text search indexes for searching

Starting with Version 9.7 Fix Pack 3 and later fix packs, DB2 Text Search and Net Search Extender text indexes can coexist on the same table column. A DB2 Text Search text index can now be created on the same column for which a Net Search Extender index already exists. You can use the `SYSPROC.SYSTS_ALTER` stored procedure to switch the text index status from `ACTIVE` or `INACTIVE` or vice versa, thereby switching between either DB2 Text Search or Net Search Extender text indexes.

The text index type is determined on query level. This implies that all text indexes used in the predicates of a single query must be of the same index type, that is, either of type DB2 Text Search or Net Search Extender. If several text indexes are

used by multiple predicates in one query, some of which are DB2 Text Search indexes while others are Net Search Extender text indexes, then the search will fail and error message SQL20425N returned.

Note: The text index switch must be accompanied by an application switch to activate the appropriate query syntax.

Simple scenarios

1. Suppose that you want to make the products in the PRODUCT table in the SAMPLE database searchable by DB2 Text Search. Assuming that you already created the sample database (by running the **db2samp1** command) and that you set the **DB2DBDFT** environment variable to SAMPLE, you could issue the following commands:

```
db2ts START FOR TEXT
db2ts ENABLE DATABASE FOR TEXT
db2ts CREATE INDEX myschema.productindex FOR TEXT ON product(name)
db2ts UPDATE INDEX myschema.productindex FOR TEXT
```

The product names and descriptions contained in the NAME column of PRODUCT are now indexed and searchable. If you want to find the product IDs of all the snow shovels, you can issue the following search query:

```
db2 "SELECT pid FROM product WHERE CONTAINS (name, 'snow shovel') = 1"
```

2. Enable a Net Search Extender database for DB2 Text Search and use the index coexistence feature to query the database.

Start the database for text search.

```
db2ts start for text
DB20000I The SQL command completed successfully.
```

Enable the Net Search Extender database for DB2 Text Search.

```
db2ts enable database for text
CIE00001 Operation completed successfully
```

Create and update a DB2 Text Search index on a column which has an existing Net Search Extender index.

```
db2ts "CREATE INDEX db2ts.title_idx FOR TEXT ON books(title)"
CIE00001 Operation completed successfully.
```

```
db2ts "UPDATE INDEX db2ts.title_idx FOR TEXT"
CIE00001 Operation completed successfully.
```

Activate the new DB2 Text Search index.

```
db2 "ALTER INDEX db2ts.title_idx FOR TEXT SET ACTIVE"
CIE00001 Operation completed successfully.
```

Issue a query to use the DB2 Text Search index.

```
db2 "select isbn, title from books where contains(title,'top')=1"
```

| ISBN | TITLE |
|---------------|------------------------------------|
| 123-014014014 | Climber's Mountain Tops |
| 111-223334444 | Top of the Mountain: Mountain Lore |

2 record(s) selected.

Issue a query that refers to both DB2 Text Search and Net Search Extender indexes in the table. The title column has an active DB2 Text Search index, while bookinfo column has an active Net Search Extender index. The search will return an error because all text indexes in one query must be of the same index type.

```
db2 "select isbn, title from books where contains(title, 'top')=1
and contains(bookinfo, ' MOUNTAIN ')=1"
```

```
ISBN          TITLE
-----
```

```
SQL20425N Column "BOOKINFO" in table "BOOKS" was specified as an argument to
a text search function, but a text search index does not exist for the column.
SQLSTATE=38H12
```

To avoid this error, create a DB2 Text Search index on the bookinfo column and activate it.

```
db2ts "CREATE INDEX db2ts.bookinfo_idx FOR TEXTON books( bookinfo )"
CIE00001 Operation completed successfully.
```

```
db2ts alter index db2ts.bookinfo_idx for text set active
CIE00001 Operation completed successfully.
```

Text search index creation, updates, and changes

Text search index creation is the process of defining and declaring the properties of an index. After you create a text search index, you can update it by adding new data from the table that it is associated with. You can also alter the properties of the text search index.

Using DB2 Text Search, you can create a text search index to allow the data of a text column to be searched using text search functions. A text search index consists of significant terms that are extracted from text documents. The primary key of a row is used to uniquely identify the text document in that row. When creating the text search index, you can specify various properties for it, including its update frequency and the directory that it will be stored in.

Immediately after its creation, a text search index contains no data. You add data to a text search index using the text-search-specific db2ts **UPDATE INDEX** command or the SYSTS_UPDATE administrative SQL routine. The first index update adds all text documents from a text column to the index. This update is typically known as the *initial update*. All subsequent updates are known as *incremental updates*.

Text search indexes are reorganized automatically as needed; you cannot explicitly trigger a reorganization or re-create an index when you update it. However, you can drop the text search index, create a new one, and perform an initial update on it.

You can change some of the properties of a text search index after creating it, such as the update frequency or the minimum number of changes required to trigger an update.

Example

In this example, a user creates a text search index called myschema.productindex on the PRODUCT table contained in the SAMPLE database. Figure 3 on page 7 shows that the index is empty until an initial update is performed and that as additional product information is added to PRODUCT, an incremental update

must be performed.

Create index for text



Update index for text (initial update)



Update index for text (incremental update)

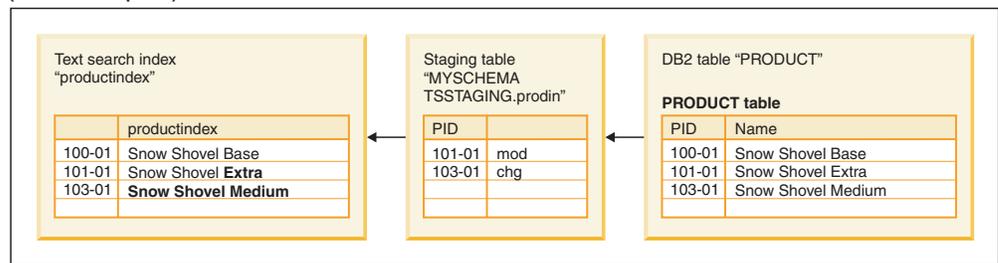


Figure 3. The two types of updates to a text search index

Linguistic processing for DB2 Text Search

DB2 Text Search provides dictionary packs to support the linguistic processing of documents and queries that are not in English. As an alternative to dictionary-based word segmentation, the search engine uses n-gram segmentation support for languages such as Chinese, Japanese, and Korean.

If a text document is in one of the supported languages, linguistic processing is carried out during the tokenization stage, that is when the text is broken up into individual words. For unsupported languages, the document is parsed using white space or n-gram segmentation. Lemmatization (like stemming, this means to find the normalized form of a word, but it also analyzes the word's part of speech) is not performed on unsupported languages.

When you search a text search index, a match is indicated if the indexed document contains the query terms or linguistic variations of the query terms. The variations of a word depend on the language of the query.

Linguistic processing for Chinese, Japanese, and Korean documents

Documents that are in Chinese, Japanese, or Korean are processed using n-gram segmentation.

For a search engine, getting good search results depends in large part on the techniques that are used to process text. After the text is extracted from the document, the first step in text processing is to identify the individual words in the text. Identifying the individual words in the text is referred to as segmentation. For many languages, white space (blanks, the end of a line, and certain punctuation) can be used to recognize word boundaries. However, Chinese, Japanese, and Korean do not use white space between characters to separate words, so other techniques must be used.

Currently DB2 Text Search does not provide dictionary-based word segmentation for Chinese, Japanese, and Korean.

N-gram segmentation

N-gram segmentation avoids the problem of identifying word boundaries, and instead indexes overlapping pairs of characters. Because two characters are used, this technique is also called bi-gram segmentation.

N-gram segmentation always returns all matching documents that contain the search terms; however, this technique can return documents that do not match the query.

By default, DB2 Text Search comes with a pre-configured index that uses n-gram segmentation for Chinese, Japanese, and Korean.

To show how both types of linguistic processing work, examine the following text in a document: election for governor of Kanagawa prefecture. In Japanese, this text contains eight characters. For this example, the eight characters are represented as A B C D E F G H. A sample query that users might enter could be election for governor, which is four characters and are represented as E F G H. (The document text and the sample query share similar characters.)

After the document is indexed, the search engine segments the text election for governor of Kanagawa prefecture into the following sets of characters: AB BC CD DE EF FG GH.

The sample query election for governor is segmented into the following sets of characters: DE EF FG GH. If you search with the sample query election for governor, the document will be found by the query because the tokens for both the document text and the query appear in the same order.

With n-gram segmentation, you will likely see more results but possibly less precise results. For example, in Japanese, if you search with the query Kyoto and a document in your index contains the text City of Tokyo, the query Kyoto will return the document with the text City of Tokyo. The reason is that City of Tokyo and Kyoto share two of the same Japanese characters.

Command-line tools for DB2 Text Search

Four command-line tools are included with DB2 Text Search to facilitate its use.

The Configuration Tool

For performing both the initial and subsequent configurations of DB2 Text Search

The Administration Tool

For performing various administrative tasks related to the DB2 Text Search server

The Synonym Tool

For adding synonym dictionaries to text search indexes and removing synonym dictionaries from text search indexes

The Log Formatter Tool

For viewing and saving system messages and trace messages

Issuing text search commands

You can issue commands by running the **db2ts** command shell or by calling one of the administrative SQL routines that is a stored procedure for DB2 Text Search.

About this task

To use the **db2ts** command shell, pass the command string as a parameter. The **db2ts** command shell acts like the DB2 command shell in that a command must contain the connection information if a remote database is used. Unlike the DB2 command shell, however, **db2ts** does not provide a session; instead, each command is a separate unit and thus must establish a connection separately. You do not have to specify the database connection if you are running the command locally for the default database specified using the **DB2DBDFT** environment variable. Set the **DB2DBDFT** environment variable at the operating system level. If you also set it using the **db2set** command, ensure that the same value is used.

Using an administrative SQL routine enables you to issue administration calls from a DB2 client on which you have not installed DB2 Text Search. You can call either the generic **SYSTS_ADMIN_CMD** administrative SQL routine with a command string as a parameter or the specific administrative SQL routine for that command.

Note: Error messages resulting from **db2ts** commands are written in the client locale, but messages resulting from the administrative routines are written in the locale specified by the message-locale argument or in **en_US** if you do not specify a locale.

Because some commands are not related to a specific database, for example, **START FOR TEXT** and **STOP FOR TEXT**, you can run them only using the **db2ts** command shell.

Rich text document support

DB2 Version 9.7 Text Search supports rich text documents. With this support, DB2 Text Search can index and search rich text documents within a properly configured DB2 Text Search instance.

DB2 Text Search supports **TEXT**, **XML**, and **HTML** text index formats to prepare indexes for full-text search on text data. DB2 Version 9.7 and rich text support adds the **INSO** text index format to prepare indexes for full-text search on rich text data.

Rich text documents are documents that contain text as well as formatting instructions such as bold, italics, font types, font sizes, spacing, and more.

Common and popular office products produce documents that are prime examples of rich text documents. A more specific example is the resume. This type of document is typically created in various office product software suites. DB2 Text Search can now store, index, and search such documents with greater ease and flexibility. Non-English dictionary keywords (for example: street name, email address, and computer languages) can also be correctly searched and retrieved.

For more information about the enablement and configuration of the rich text support feature, see the following topics.

Text search solution planning

Understanding certain key concepts, such as supported document types and languages and user roles, will help you leverage the benefits of DB2 Text Search.

Security considerations for DB2 Text Search

There are a number of steps that you need to take to ensure that confidential data is protected.

Table 1. Security-related issues and solutions

| Issue | Solution |
|---|--|
| Access and audit policies for the base data table are not inherited for the staging table. | The staging table will need to be appropriately protected if confidential data is used in the primary key of the base table |
| If confidential data is indexed, this data is represented in the text index and some data might be readable. | The file and folder access permissions should be checked to verify that they match with the company policy. |
| Text indexes are located on the file system and are accessible to users with read or write privileges on the directory that contains the text index files. Access privileges for text index directories are not automatically restricted by the text search system except for limitations inherited for the default location in the database instance path. | Check (and change, if necessary) the access permissions for custom collection directories to restrict access to the process owner of the text search server process. This process is started by the DB2TS service associated with the DB2 instance and is the "Log on as" user specified for this Windows service. |
| Fenced user ID and instance user ID must be members of the same primary group to grant the instance user ID correct access to files created by the fenced user ID and vice versa. | Assign correct group membership and file permissions. |

User roles

There are different user roles and authorizations for users of DB2 Text Search.

Typical users include the following ones:

- DB2 instance owners
- Database administrators
- User table owners
- Users performing text search queries

Instance owner

The instance owner user ID is used to start and stop the DB2 Text Search instance services on the DB2 server. Note that the DB2 instance and the DB2 Text Search services must be processes of the same user ID.

In Version 9.7, the instance owner must hold DBADM with DATAACCESS authority.

In addition to performing its primary task of controlling the instance services, the DB2 instance owner user ID acts as a central point of control for all database changes triggered by DB2 Text Search. Another primary task is cleaning up any unused text search indexes.

The instance owner is determined differently on UNIX and Windows operating systems:

- On UNIX operating systems, the instance owner user is the name and user ID of the instance specified for the **db2icrt** command.
- On Windows operating systems, the instance owner is the user ID running the DB2 instance service.

Database administrator

The database administrator enables and disables databases for use with DB2 Text Search. Another main task that the database administrator performs is clearing command locks.

This role corresponds to DBADM authorization, which allows the manipulation of all database objects, including text search indexes.

User table owner

The user table owner defines and maintains text search indexes.

Typical user table owner tasks include the following ones:

- Creating text search indexes and defining their characteristics
- Updating text search indexes
- Changing the update characteristics of text search indexes
- Dropping text search indexes
- Clearing the event table periodically

The user table owner role corresponds to the CONTROL privilege for the table on which a text search index is created.

User performing text search queries

Users who perform search queries can use the DB2 Text Search CONTAINS and SCORE functions in an SQL query against a user table. They can also use the xmlcolumn-contains function in an XQuery that references a table with a text search index.

There is no specific DB2 Text Search search authorization. Depending on the access rights that the users are granted on the table that the text search index is created on, the query is permitted or rejected. If users can issue a SELECT statement on a given table, they can also perform a text search on that table.

Users performing the search queries can include the following functionality in their queries:

- Limit the text search to a particular document (using SQL or XQuery)
- Return a score indicating how well a document compares with other matching documents for a given search argument (using SQL)

DB2 Text Search capacity planning

A number of factors influence performance and resource use in DB2 Text Search. When planning system capacity for DB2 Text Search, consider the number of

parallel index updates, the planned size of your text indexes, and the processing time for the documents you are indexing. Various system setup and capacity factors should be considered for full-text search.

DB2 Text Search now enables full-text search queries on most data types within the DB2 database, including support for XML documents and a Rich-Text/Proprietary format feature. Full-text search is supported through a text search server instance that is integrated with the database instance setup and connected through TCP/IP. Full-text indexing and search performance depends on text index specific settings, Text Search Server configuration, and available system resources.

Size of text search indexes

In an DB2 Text Search server setup, an average text search index is about 50-70% the size of the original data. Starting with DB2 Version 9.7 Fix Pack 4, an average text search index is about 50-150% the size of the original data.

The size of the text search index in an integrated text search server depends on the following factors:

- the average size of the document
- the size of the indexed column
- the size of the primary key
- the relationship of sizes for primary key and index data
- the number and distribution of terms

There is no absolute size limit imposed on any text search index. However, when a considerable amount of data is added to or removed from a text search index, the text search index structure is merged to improve query performance. The time for completion depends on the size of the index.

Absolute throughput depends on the data type and the index format. Combining absolute throughput dependencies with completion time dependencies results in practical limits on the total text search index size.

Text search server deployment and configuration

A single text search server is configured for the database instance. The text search server has a recommended minimum memory requirement of 4 GB of memory for production use, which increases according to the number of parallel index updates.

Updating the text search index is resource-intensive, both in terms of disk I/O and CPU/memory requirements. Multiple configuration parameters are available to control the Text Search server resource usage.

Factors affecting throughput

Absolute text index update throughput depends on the data type and the index format. Optimal processing occurs when there is approximately 10-100 KB of text per document. Throughput degrades if text is less than 1 KB and more than 1 MB.

The time required to complete a text index update depends mainly on the following factors:

- the number of documents to be indexed
- the document size

- the index type
- index update parallelism
- text search server configuration

The processing time for each document is the sum of an approximate fixed time and a variable time. The fixed time is influenced by the document type, such as plain text, XML, or INSO. The fixed time is approximate because there can be minor variations in time for memory usage or reuse. The variable time is determined mainly by the document size and linguistic processing variations.

For indexes of INSO documents, handling different MIME types can also affect the processing time.

The number of documents that can be processed in a given timeframe increases for smaller document sizes. However, the total throughput is less for smaller documents than for larger documents due to the fixed cost per document.

For perceived query performance, the biggest impact is due to the number of matching results, not the size of the text search index. For example, a query with a single predicate using a single-term search term on a 100 GB text search index performs similar to a search on an 800 GB text search index if the number of results is the same.

DB2 Text Search server configuration

You can tune your DB2 Text Search configuration by adjusting the queue sizes, heap size, number of indexing threads, and other factors. Balance your adjustments to these different parameters for optimal performance of your system.

It is recommended for the DB2 Text Search server configuration that the number of indexer threads should not exceed the number of CPUs, and the number of parallel updates should not exceed the number of indexer threads.

For an integrated DB2 Text Search server, stop the DB2 Text Search instance services using the **STOP FOR TEXT** command before making any configuration changes, and use the configurationTool to modify the values. For example

```
configTool configureParams -configPath <configPath> -numberOfIndexingThreads 3
```

For your changes to take effect, restart the DB2 Text Search server.

Maximum heap size configuration:

When a document is received by the document ingestion thread, its content is placed in the document queue. Documents placed on the document queue remain there until an active indexing thread indexes it.

In a typical operation, the speed of placing documents on the document queue is faster than the time required to parse and index the document. Therefore, at some point in time, the document queue reaches its capacity, and the document ingestion thread is blocked until another slot is freed from the document queue.

As the document queue fills with unprocessed documents, it consumes heap memory. Further memory is consumed for document processing like parsing and indexing. The combined heap memory consumption must be less than the

maximum heap size of the process. By default, the heap size is configured to be 1500 MB. If you plan to have a continuous number of large documents, adapt the heap size accordingly.

Also, consider the ratio between the input and output queue memory size and the heap memory. The queue size is determined by the memory consumption of the documents in the queue. If you intend to process long documents, like 20 MB each, and decide to increase the queue memory size, consider increasing the heap size.

The *startupHeapSize* variable, also known as *maximumHeapSize* until DB2 Version 9.7 Fix Pack 3, sets the maximum allowed heap size for the integrated DB2 Text Search server. The default startup heap size is 1.5 GB. This value must be a number between 1.5 GB and the maximum amount of memory allowed by your operating system and JVM version. Consider the following examples:

- If you have a Windows system with a 32-bit JVM, then a process can have a maximum heap size of 2 GB. Therefore, your *startupHeapSize* parameter must be set to less than 2 GB. For example, 1.8 GB.
- If you have an AIX system with a 64-bit JVM, then the maximum heap size is limited only by the amount of virtual memory configured on the system. If you have an integrated text search server and many large documents with an average size of 20 MB need to be processed in a continuous flow, increase the *startupHeapSize* to approximately 4 GB.

To configure the maximum heap, issue the following command

- up to DB2 Version 9.7 Fix Pack 3:

```
configTool configureParams -configPath <full-path-to-configuration-folder>  
-maximumHeapSize <value>
```
- starting from DB2 Version 9.7 Fix Pack 4:

```
configTool configureParams -configPath <full-path-to-configuration-folder>  
-startupHeapSize <value>
```

where, *<value>* is the heap size and *<full-path-to-configuration-folder>* is the full path to the *config.xml* file for DB2 Text Search server.

DB2 Text Search indexing threads:

Multiple indexing threads work in parallel to parse and index documents. This usually reduces the total elapsed time for text search index updates.

Indexer threads pick documents from the queue and manage the indexing process. They make use of index preprocessing threads to prepare the document content for indexing and write the result to the text index collection.

Index preprocessing threads extract text, identify the language, tokenize, and analyze the document.

Up to DB2 Version 9.7 Fix Pack 3 the number of preprocessing threads was matched to the number of indexing threads. Starting from DB2 Version 9.7 Fix Pack 4 it is possible to configure indexing threads and preprocessing threads separately. Usually the number of indexer threads and index preprocessing threads is configured to be the same. However, in some scenarios, for example, when large documents are processed, increasing the number of preprocessing threads might provide a performance benefit.

Indexing thread usage

If multiple indexer threads work on the same collection, the effect is reduced by the coordination required to synchronize the processing among the threads. Also, indexing threads that are single threaded perform better while parsing, but there can be a performance hit while merging or writing to disk. For example, four indexing threads working on four different text indexes show better throughput than four indexing threads working on a single text index.

Number of indexing threads

You should have at least two indexing threads and ensure that the number of indexing threads does not exceed the number of available CPUs. The maximum number of parallel index updates should not exceed the number of indexing threads to avoid thread sharing. With too many indexing threads or too many parallel index updates, the overall system performance suffers due to the increased memory usage for process context switches.

For example, if 40 text indexes are frequently updated, and the system contains 8 CPUs, do not use more than eight indexing threads. Also, use a staggered update schedule for the text indexes to minimize contention for index threads.

The default setting for the number of indexer threads is 4, the same default applies to index preprocessing threads.

For DB2 Version 9.7 Fix Pack 3 or earlier fix packs, to configure the number of indexing threads in an integrated text search setup, edit the *full-path-to-configuration-folder/config.xml* file and modify the following parameter:

```
<numberOfIndexers>4</numberOfIndexers>
```

Starting from DB2 Version 9.7 Fix Pack 4, to configure the number of indexing threads, issue the following command:

```
configTool configureParams -configPath full-path-to-configuration-folder  
-numberOfIndexingThreads value
```

where *value* is the number of threads and *full-path-to-configuration-folder* is the full path to the *config.xml* file for the DB2 Text Search server.

To configure the number of preprocessing threads, issue the following command:

```
configTool configureParams -configPath full-path-to-configuration-folder  
-numberOfPreprocessingThreads value
```

where *value* is the number of threads and *full-path-to-configuration-folder* is the full path to the *config.xml* file for the DB2 Text Search server.

DB2 Text Search queue memory size:

The queue memory size for DB2 Text Search must be set properly for optimal index update processing.

For DB2 Version 9.7 Fix Pack 3 or earlier fix packs, for a text search server, configure the document queue size by editing the *<full_path_to_configuration_folder>/config.xml* file, where *<full_path_to_configuration_folder>* is the full path to the folder containing the *config.xml* file. Modifying the following parameter:

```
<documentQueueSize>120</documentQueueSize>
```

This value must be a number between 4 (or the number of indexing threads) and 120. The default is 120.

Starting from DB2 Version 9.7 Fix Pack 4, queue memory assignment can be controlled both for the database and for the text server. The database queue memory determines the number of documents that can be sent to the text server for update processing at any time. To control the size of the database queue memory, update the SYSIBMTS.TSDEFAULTS administration view and set the value for the **DocumentResultQueueSize** parameter. The default value is 30,000. This value is used to limit how much database memory is reserved per update operation for a collection.

To change the value issue the following command:

```
db2 "update sysibmts.tsdefaults set defaultvalue='10000'
where lower(defaultname)='documentresultqueuesize'"
```

The second mechanism for queue memory control applies to the text server. Two configuration values determine the use of queue memory.

- *inputQueueMemorySize*:
Specifies the memory size of the input queue on the indexing server. The input queue contains documents that are waiting for preprocessing. A larger memory size will be faster, but will consume more resources. The default size is 15 MB.
- *outputQueueMemorySize*:
Specifies the memory size of the output queue on the indexing server. The output queue contains documents that are waiting to be indexed after preprocessing. A larger memory size will be faster, but will consume more resources. The default size is 15 MB.

Consider the ratio between the input and output queue's memory size and the heap memory. The queue size is determined by the memory consumption of the documents in the queue. If you intend to process long documents, for example 20 MB each, consider increasing the queue memory size and increasing the heap size.

To change, for example, the *inputQueueMemory* size, issue the following command:

```
configTool configureParams -configPath <full-path-to-configuration-folder>
-inputQueueMemorySize <value>
```

where *<value>* is the memory size and *<full-path-to-configuration-folder>* is the full path to the config.xml file for DB2 Text Search.

DB2 Text Search index planning and optimization

Data source characteristics have major impact on performance.

The time required to complete a text index update depends mainly on the following factors:

- the number of documents to be indexed
- the document size
- the index type
- index update parallelism
- text search server configuration

The processing time for each document is the sum of an approximate fixed time and a variable time. The fixed time is influenced by the document type, such as plain text, XML or INSO. The fixed time is approximate because there can be

minor variations in time for memory usage or reuse. The variable time is determined mainly by the document size and linguistic processing variations.

For indexes of INSO documents, handling different MIME types can also affect the processing time.

The number of documents that can be processed in a given timeframe increases for smaller document sizes. However, the total throughput is less for smaller documents than for larger documents due to the fixed cost per document.

DB2 Text Search index source characteristics:

Data source characteristics have major impact on performance.

To enhance performance during indexing or search, use the following techniques:

- For primary key columns, use numeric data types, such as INTEGER, instead of a VARCHAR type. Avoid primary keys that are a compound of multiple VARCHAR columns to minimize traffic for query results.
- Ensure that your system has enough real memory available for the index update operation. Index updates require memory that is in addition to that required for any database buffer pools. If there is insufficient memory, the operating system uses paging space instead which decreases search performance considerably.
- If large numbers of small documents must be processed in text search server index updates, consider reducing the number of parallel index updates and instead increase the queue sizes to increase the maximum flow of documents to the text server. See the capacity planning topics for details.
- Ensure that the content to be indexed is accessible and of proper format, as the performance might decrease during an index update if many error and warning messages are written to the event table.

Asynchronous index updates:

To improve performance, a text search index is not synchronized with its associated user table within the scope of a DB2 transaction that updates, deletes text documents from, or inserts text documents into that table. Instead, text search indexes are updated asynchronously.

To facilitate the asynchronous update of a text search index, a staging table (also known as a log table), is created for each text search index. Triggers on the text table capture any changes, for example, inserts, deletes, or updates, to a text column that the text search index is associated with and write them to the staging table. The updates to the text search index are applied at a later stage, during either a manual update or an automatic update. The update is made to a copy of a very small part of the index. During the update, you can still perform searches on the index, but you cannot access the updated text search index until the synchronization is complete.

When using the `updateautocommit` option, consider that each commit cycle results in significant memory usage, which increases the total time for completing an index update. Setting a `commitcount` is especially useful for initial updates, or updates that involve all rows or a majority of rows, if the total elapsed time is large to avoid losing completed work due to a rollback caused by a system or server failure.

To calculate a `commitcount` value, consider that calculations for text index processing are very different from in-database transactions. Each cycle is treated as a complete index update and therefore executes preparation and closing steps, including necessary index segment merges or index reorganizations. Very short commit cycles will therefore significantly increase total elapsed time and should be avoided.

Examples

Example 1: With an average document size of 10k per row, a throughput on the specific system of 12 GB/hr, and a total of 240 GB to process, an index update without commit cycles requires about 20 hours, processing 1,200,000 rows/h and a `commitcount` might be set between 2000000 and 2500000 for a commit every 2 hours.

Example 2: With an average document size of 100b per row, a throughput of 1 GB/hr on a specific machine, and a total of 60 GB to process, an index update without commit cycles requires about 60 hours, processing 10,000,000 rows/hr and a `commitcount` of 30,000,000 may strike a balance.

Optimizing a DB2 Text Search index:

DB2 Text Search index optimization compacts the text search index and speeds up indexing and searching. Optimization removes deleted documents from the text search index and merges the index segment files on the disk.

Optimization and indexing of the same index cannot be performed in parallel. Take this into account when scheduling optimization and indexing sessions. However, optimization and search can be performed in parallel. Disk space consumption during index optimization can be high, especially if the same index is searched in parallel.

You can optimize the index after you completely index your document set or after incremental index updates. Index optimization can take a long time, depending on the index size. If your incremental updates add documents frequently, perform optimization less frequently to minimize the extra memory usage during the optimization process.

To optimize the index, start the administration tool with the **optimizeIndex** command. For example:

```
adminTool.bat optimizeIndex -configPath c:\mydbinst\db2tss\config  
-collectionName MyCollection
```

You can check the status of the last executed optimization process by running the administration tool with the **optimizeIndexStatus** command.

Disk consumption:

Text index size

The amount of disk space a text search index uses depends highly on the nature of the text in each document. However, there is an approximately linear relationship between the disk space required for the text search index and the disk space required for the original data. Typically, the size of the index on the disk is 50 - 150% of the original text size. For example, the text search index for 100,000 200

KB documents is expected to require about 1100 MB of disk space (100,000 x 20 KB x 55%). The size of the text search index relative to the source documents depends on the following factors:

- the average size of the document
- the size of the indexed column
- the size of the primary key
- the relationship of sizes for primary key and index data
- the number and distribution of terms

During the index update, additional work space is needed. The intermediate space requirements are about a factor 2-3 times the final text search index size, provided the maximum segment size is not reached. The free space required is 2-3 times the maximum segment size. Disk space is reserved even after a segment merge if the old segments have been used in a search.

There is no absolute size limit for text search indexes, however, the combination of throughput factors with completion time dependencies results in practical limits on the total text search index size. For example, when a considerable amount of data is added to or removed from a text search index, the text search index structure is merged to improve query performance, and the time for completion of the merge depends on the size of the index.

Log files

In addition to the `db2diag.log` file, DB2 Text Search generates trace and Configuration tool log files with messages from the DB2 Text Search server.

For a DB2 Text Search server, the default log file location is `db2tss/log` directory. If you want DB2 database and text search logs in the same location, set the location to `<instanceHome>/sql1lib/db2dump/tslog` on UNIX or `<instanceProfilePath>\<instance_name>\db2tss\tslog` on Windows platforms.

Ensure that the target location has sufficient free disk space for the log files. A minimum of 100 MB of free disk space is required. Without sufficient space for the log files, the text search service stops logging and throws a disk full error.

Starting with DB2 Version 9.7 Fix Pack 4, you can now configure logging for the text search server by modifying logging property files that are located in the `db2tss/config` directory. The `ecmts_logging.properties` file determines trace logging properties. The `ecmts_config_logging.properties` file determines command-line tool logging properties. These files are `java.util.Logger` property files. Comments in the files explain how to modify them.

Administrative tables

If you do not specify a table space for the administrative tables for the text index when you run the **CREATE INDEX FOR TEXT** command, the administrative tables are created in the table space that contains the base table. To determine the appropriate location, consider the following information:

- Staging table for the text index

The staging table holds the reference to rows that have been updated in the base table for an incremental update of the text index. This table is automatically cleaned up with each update:

Size =
number of rows for index updates * (primary key length for base table + 18)

- **Event table for the text index**

The event table contains status information about text index processing, including errors and warnings during an index update. In the worst case, if each document is rejected due to a nonfatal error, the number of events is the number of documents plus a few begin and end messages for the update process. The event table is not cleaned automatically, and increases in size until a **CLEAR EVENTS FOR INDEX** operation is issued.

Event table size =
number of events * (length of primary key of base table + 1036)

DB2 Text Search index location:

For an integrated Text Search server, configuration and collection metadata is stored in *instanceHome*/sql/lib/db2tss/config on UNIX or *instanceProfilePath*\instance_name\db2tss\config on Windows.

If you plan to create multiple large indexes, consider storing them on separate or striped disk devices, in particular if concurrent index updates are scheduled.

Index specific parameters for DB2 Text Search index updates:

Starting with DB2 Version 9.7 Fix Pack 4, you can configure the following collection-specific parameters to improve performance:

- **MaxMergeDocs**
- **MergeFactor**
- **BufferSize**

You can modify indexing parameters for a particular collection by editing the \config\collections*collection_name*\collection.xml file. To modify the default settings for future collections that are created, set the values of these parameters in the \config\defaults\collection.xml file.

MaxMergeDocs

The **MaxMergeDocs** parameter defines the largest segment (measured by the number of documents) that can be merged with other segments in the index. There is a trade-off between overall indexing throughput and segment merge time.

If you specify a low value for the **MaxMergeDocs** parameter (for example, 100,000 documents), your segments will be limited in size. In this case, segment merges are quicker and indexing flows more smoothly without time-outs. However, if your content is very large, there will be numerous segments and a degradation in indexing throughput over time.

If you specify a high value for the **MaxMergeDocs** parameter (for example, 100,000,000 or 500,000,000 documents), you get fewer segments (until the index becomes very large) and the overall indexing throughput is better. However, segment merges take more time and you might encounter time-outs during indexing.

Typically the value of **MaxMergeDocs** should be higher for collections of small documents and lower for collections of larger documents.

MergeFactor

The **MergeFactor** parameter defines the number of segments that are merged at a time and also controls the total number of segments that can accumulate in the index. There is a trade-off between frequent, small merges (for example, two at a time) and less frequent, large merges (for example, 10 at a time). You can specify a smaller value for the **MergeFactor** parameter to avoid time-outs. Modifying the merge factor does not typically impact performance.

BufferSize

The **BufferSize** parameter specifies the amount of RAM that can be used for buffering added documents before the documents are flushed as a new segment. There is a trade-off between frequent, small flushes to disk and less frequent, large flushes to disk. In some cases you can improve performance by increasing the value of the **BufferSize** parameter. For example, when you index a single collection of small documents, increasing the buffer size will improve performance, especially for the first 100,000 documents in the index.

DB2 Text Search system tuning

Text index update processing and text search query performance are influenced by various system characteristics.

Consider the following factors:

- TCP/IP port considerations for Windows
- File descriptors

TCP/IP port considerations for DB2 Text Search and Windows:

On Windows 32-bit operating systems, your ability to handle high query loads is affected by the number of TCP/IP ports and the wait time to reuse a port.

Port assignments on Windows (32-bit)

The integrated DB2 Text Search runs as a separate process on the same host as the database server. The database server and text server communicate through a TCP/IP connection.

The number of available ports for TCP/IP connections is influenced by the number of ports and the wait time to reuse a port after a connection is closed. The default configuration values for these parameters might not be sufficient to provide enough available ports to serve a high query load. If you have too few TCP/IP ports, you might get an CIE00756 Connection failed error.

If a CIE00756 Connection failed error occurs, run the following commands to view port usage on the server:

```
netstat -n  
netstat -n | c:\windows\system32\find /I port_number
```

If the output shows many TCP/IP connections and local addresses `127.0.0.1:port_number` in `TIME_WAIT` state, the server is likely running out of TCP/IP ports.

You can determine the DB2 Text Search port numbers by issuing the following command:

```
configTool printAdminHTTPPort -configPath %INSTPROF%\%DB2INSTANCE%\db2tss\config
```

where, INSTPROF is set to the value of the **DB2INSTPROF** registry variable applicable to integrated DB2 Text Search server setups.

Port settings

Port settings are controlled by the following registry entries that are found in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TCPIP\Parameters:

- **TcpTimedWaitDelay**

A DWORD value, in the range 30 - 300, that determines the time in seconds that elapses before TCP/IP can release a closed connection and reuse its resources. Set the **TcpTimedWaitDelay** value to a low value to reduce the amount of time that sockets stay in TIME_WAIT state.

- **MaxUserPort**

A DWORD value that determines the highest port number that TCP/IP can assign when an application requests an available user port. Set **MaxUserPort** to a high value to increase the total number of sockets that can be connected to the port.

A system making many connection requests might perform better if **TcpTimedWaitDelay** is set to 30 seconds, and **MaxUserPort** is set to 32678.

After adding or changing the registry entries, reboot the Windows machine to reflect the changes.

DB2 Text Search file descriptors:

For DB2 Text Search index updates and queries, system resources such as file descriptors are consumed to handle multiple index update and search requests.

In a typical system, the number of open file descriptors per process may be limited to a relatively small number like 1024, which can result in the text search server running out of file descriptors. If this occurs, the search and update requests will fail.

To resolve this error

- Check the server logs for an exception with the message string similar to too many open files.
- On UNIX systems, check the system limits with `ulimit -a`.

To increase file descriptors, follow these steps:

1. Shut down the text search server.
2. Increase the number of file descriptors per process by following your operating system manual. This increase in file descriptors must be sufficient to accommodate all requests across login sessions.
3. Restart the text search server.

Document formats supported for DB2 Text Search

You must specify the format (or type) of text documents that you intend to search using DB2 Text Search. This information is necessary for indexing text documents.

The text column data can be plain text, rich text, an HTML document, or an XML document. Documents are parsed to extract only the relevant parts for indexing, thus making them searchable. For example, tags and metadata in an HTML document are not extracted because they are not indexed.

Supported data types

The data types in the text columns that you want to index and search can be either binary or character.

DB2 Text Search supports the following data types:

- CHAR
- VARCHAR
- LONG VARCHAR
- CLOB
- DBCLOB
- BLOB
- GRAPHIC
- VARGRAPHIC
- LONG VARGRAPHIC
- XML

Conversion of unsupported formats and data types

You can use your own function to convert an unsupported format or data type into a supported format or data type.

By creating the text index using a user-defined function (UDF), you can convert an unsupported format to a supported format that can be processed during indexing by filtering the unsupported characters.

You can also use this approach for indexing documents that are stored in external unsupported data stores. In this case, where a DB2 column contains document references, you can use a UDF to return the content of documents that have the relevant document reference.

Supported languages and code pages

You can specify that the text documents be parsed using a particular language when you first create a text search index. You can also specify that the query terms be interpreted in a particular language while searching. In addition, you can specify a code page when you create a text search index on a binary data type column.

Language specification

A *locale* is a combination of language and territory (region or country) information and is represented by a five-character locale code. You define the message locale for a text search administration procedure by passing the procedure the locale code. Refinements of these locale codes are possible depending on the locales installed on the DB2 server.

There is an important difference between specifying a language when you create a text search index and specifying a language when you issue a search query:

- The locale that you specify in your **db2ts CREATE INDEX** command determines the language used to tokenize or analyze documents for indexing. If you know that all documents in the column to be indexed use a specific language, specify the applicable locale when you create the text search index. If you do not specify a locale, the database territory will be used to determine the default setting for **LANGUAGE**. To have your documents automatically scanned to determine the locale, in the **SYSIBMTS.TSDEFAULTS** view, set the **LANGUAGE** attribute to **AUTO**. The **SYSIBMTS.TSDEFAULTS** view describes database defaults for text search using attribute-value pairs.
- The locale that you specify in a search query is used to perform linguistic processing on the query and to help identify the base forms of the query term. After the locale of the base form has been identified, the locale does not play any part in the search process itself. Thus, you could use the English language for a query and obtain German documents in the search result if the search term in its base form is present in the documents.

The list of supported locales can be found [here](#).

Code page specification

You can index documents if they use one of the supported DB2 code pages. Although specifying the code page when creating a text search index is optional, doing so helps to identify the character encoding of binary columns. If you do not specify a code page for binary columns, the code page from the column property is used. .

Document size considerations

DB2 Text Search has limits on the size of the document that can be indexed and on the number of characters within that document.

The maximum size of documents that can be processed successfully is 100 MB. If a document exceeds the size limit, that document is rejected and an entry is created in the event table with that information, including the primary key to identify it. Processing continues for other documents that are a part of that update operation.

Document truncation

DB2 Text Search limits the number of Unicode characters that you can index for each text document. Sometimes, this character limit results in the truncation of large text documents in the text search index.

The default value for the number of Unicode characters allowed for each text document is 10 000 000. For a rich text document, this limit is applied after the document is transformed to plain text.

If a text document is truncated during the parsing stage, you receive a warning that some text was not processed correctly or completely. Text that is incorrectly or incompletely processed is not indexed and cannot be searched. Details about the warning are written to the event table that was created for the text search index.

You might want to remove a truncated document to avoid unexpected behavior during search processing. You can remove the document by removing it from the DB2 table or by changing the value for the document to empty or null.

The maximum number of characters indexed for XML documents is the same as for non-XML documents, but the count includes not only the plain text but also tag

names, attribute names, and attribute values. XML directives and comments are not indexed and do not count toward the truncation limit.

Chapter 3. Installing and configuring DB2 Text Search

DB2 Text Search is an optionally installable component whose installation and configuration are fully integrated with the installation of all DB2 database server products.

You can have the DB2 installer automatically install and configure DB2 Text Search. The steps that you must take are platform dependent. Figure 4 describes the installation and configuration process on Windows operating systems, and Figure 5 on page 28 describes the process on Linux and UNIX operating systems.

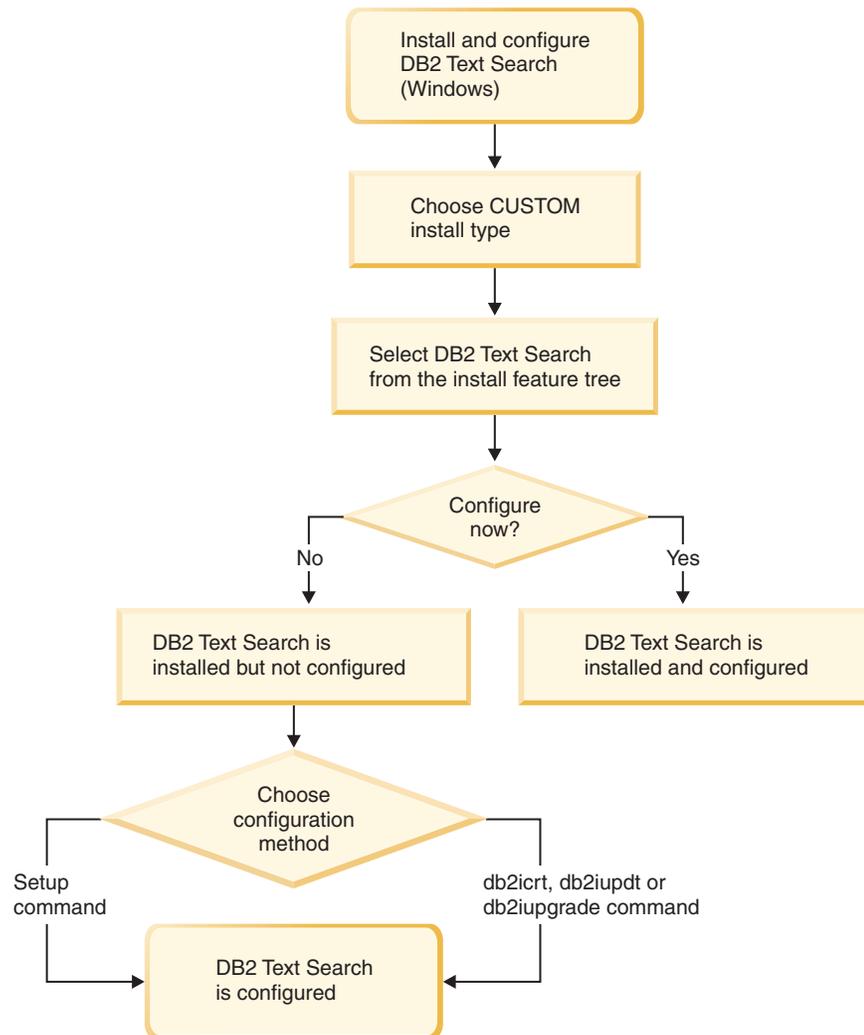
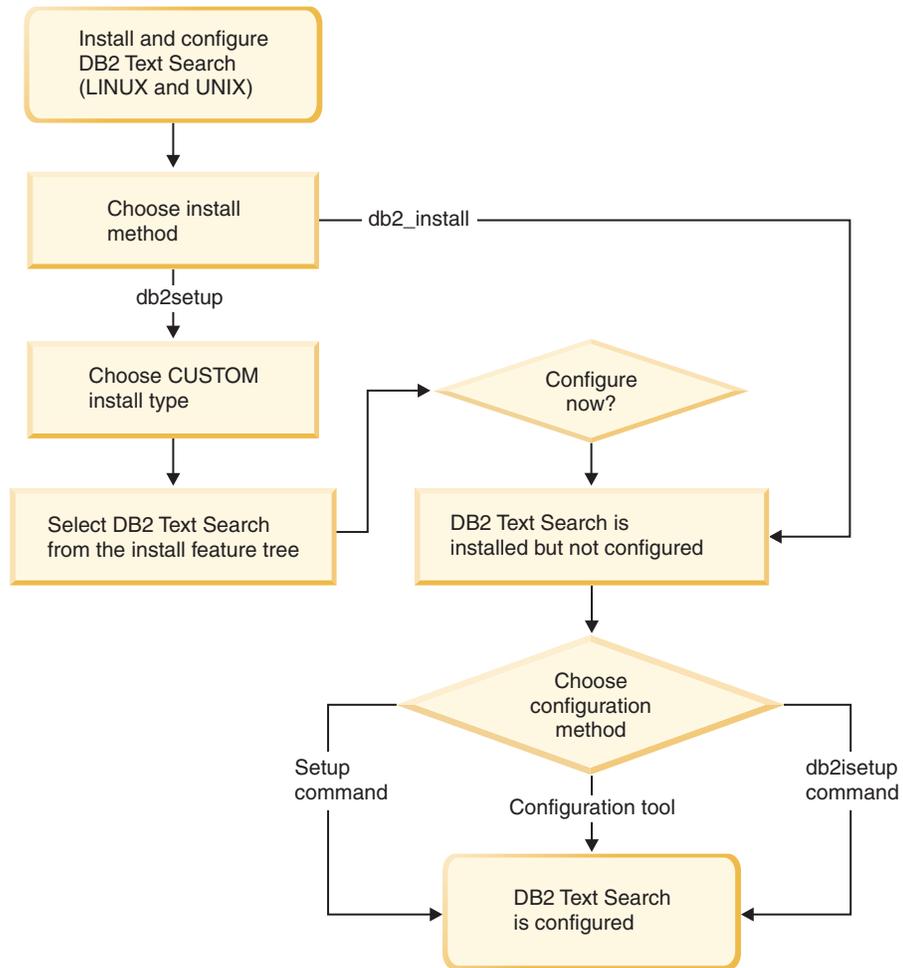


Figure 4. Installing and configuring on Windows platforms



1. If you run **db2setup** as a non-root user, have your system administrator (who has SYSADM authority) run the **db2rfe** command afterwards to reserve the port number that you want in the services file.

Figure 5. Installing and configuring on Linux and UNIX platforms

Restrictions

DB2 Text Search does not support a partitioned database environment.

Installing and configuring DB2 Text Search with the DB2 Setup Wizard

You can install DB2 Text Search with the DB2 Setup Wizard as a part of a custom installation of your DB2 database product.

About this task

Perform a custom installation of your DB2 database product and select DB2 Text Search from the feature tree. You can have DB2 Text Search automatically configured, or you can manually configure it later.

Procedure

To perform a custom installation of DB2 Text Search using **setup** or **db2setup**:

1. Install the DB2 server using the instructions for your platform:

- "Installing DB2 servers using the DB2 Setup wizard (Windows)" in *Installing DB2 Servers*
- "Installing DB2 servers using the DB2 Setup wizard (Linux and UNIX)" in *Installing DB2 Servers*

You can select the DB2 Text Search component from the feature tree. During the installation, you have the option to configure DB2 Text Search for the default instance. If you do not want to configure DB2 Text Search, skip step 2.

2. To configure DB2 Text Search yourself, provide a valid service name and port number if these fields do not already have values. You do not have to configure DB2 Text Search immediately after installing it; you can configure it later. For instructions on how to perform the configuration later, see "Configuring DB2 Text Search" on page 41.

Installing DB2 servers (Windows)

This task describes how to start the DB2 Setup wizard on Windows. You will use the DB2 Setup wizard to define your installation and install your DB2 database product on your system.

Before you begin

Before you start the DB2 Setup wizard:

- If you are planning on setting up a partitioned database environment, refer to "Setting up a partitioned database environment".
- Ensure that your system meets installation, memory, and disk requirements.
- If you are planning to use LDAP to register the DB2 server in Windows operating systems Active Directory, you should extend the directory schema before you install, otherwise you will need to manually register the node and catalog the databases. For more information, see the "Extending the Active Directory Schema for LDAP directory services (Windows)" topic.
- If you are planning to use the IBM® Tivoli® Monitoring for Databases: DB2 Agent, before installing your DB2 product see topic "Installing IBM Tivoli Monitoring for Databases: DB2 Agent with the DB2 installer" for details, limitations and restrictions.
- You must have a local Administrator user account with the recommended user rights to perform the installation. In DB2 database servers where LocalSystem can be used as the DAS and DB2 instance user and you are not using the database partitioning feature, a non-administrator user with elevated privileges can perform the installation.

Note: If a non-Administrator user account is going to do the product installation, then the VS2005 runtime library must be installed before attempting to install a DB2 database product. The VS2005 runtime library is needed on the operating system before the DB2 database product can be installed. The VS2005 runtime library is available from the Microsoft runtime library download website. There are two choices: choose `vcredist_x86.exe` for 32-bit systems or `vcredist_x64.exe` for 64-bit systems.

- Although not mandatory, it is recommended that you close all programs so that the installation program can update any files on the computer without requiring a reboot.
- Installing DB2 products from a virtual drive or an unmapped network drive (such as `\\hostname\sharename` in Windows Explorer) is not supported. Before attempting to install DB2 products, you must map the network drive to a Windows drive letter (for example, Z:).

About this task

Restrictions

- You cannot have more than one instance of the DB2 Setup wizard running in any user account.
- The DB2 copy name and the instance name cannot start with a numeric value. The DB2 copy name is limited to 64 English characters consisting of the characters A-Z, a-z and 0-9.
- The DB2 copy name and the instance name must be unique among all DB2 copies.
- The use of XML features is restricted to a database that has only one database partition.
- No other DB2 database product can be installed in the same path if one of the following product is already installed:
 - IBM Data Server Runtime Client
 - IBM Data Server Driver Package
 - *DB2 Information Center*
- The DB2 Setup wizard fields do not accept non-English characters.
- If you enable extended security on Windows Vista or Windows 2008, or higher, users must belong to the DB2ADMNS or DB2USERS group to run local DB2 commands and applications because of an extra security feature (User Access Control) that limits the privileges that local administrators have by default. If users do not belong to one of these groups, they will not have read access to local DB2 configuration or application data.

Procedure

To start the DB2 Setup wizard:

1. Log on to the system with the local Administrator account that you have defined for the DB2 installation.
2. If you have the DB2 database product DVD, insert it into the drive. If enabled, the autorun feature automatically starts the DB2 Setup Launchpad. If the autorun does not work, use Windows Explorer to browse the DB2 database product DVD and double-click the **setup** icon to start the DB2 Setup Launchpad.
3. If you downloaded the DB2 database product from Passport Advantage®, run the executable file to extract the DB2 database product installation files. Use Windows Explorer to browse the DB2 installation files and double-click on the **setup** icon to start the DB2 Setup Launchpad.
4. From the DB2 Setup launchpad, you can view installation prerequisites and the release notes, or you can proceed directly to the installation. You may want to review the installation prerequisites and release notes for late-breaking information.
5. Click **Install a Product** and the Install a Product window will display the products available for installation.

If you have no existing DB2 database products installed on your computer, launch the installation by clicking **Install New**. Proceed through the installation following the DB2 Setup wizard's prompts.

If you have at least one existing DB2 database product installed on your computer, you can:

- Click **Install New** to create a new DB2 copy.

- Click **Work with Existing** to update an existing DB2 copy, to add function to an existing DB2 copy, upgrade an existing DB2 Version 8, Version 9.1, or Version 9.5 copy, or to install an add-on product.
6. The DB2 Setup wizard will determine the system language, and launch the setup program for that language. Online help is available to guide you through the remaining steps. To invoke the online help, click **Help** or press **F1**. You can click **Cancel** at any time to end the installation.

Results

Your DB2 database product will be installed, by default, in the *Program_Files\IBM\sql1lib* directory, where *Program_Files* represents the location of the Program Files directory.

If you are installing on a system where this directory is already being used, the DB2 database product installation path will have *_xx* added to it, where *xx* are digits, starting at 01 and increasing depending on how many DB2 copies you have installed.

You can also specify your own DB2 database product installation path.

What to do next

- Verify your installation.
- Perform the necessary post-installation tasks.

For information about errors encountered during installation, review the installation log file located in the *My Documents\DB2LOG* directory. The log file uses the following format: *DB2-ProductAbbrev-DateTime.log*, for example, *DB2-ESE-Tue Apr 04 17_04_45 2008.log*.

If this is a new DB2 product installation on Vista 64-bit, and you will use a 32-bit OLE DB provider, you must manually register the *IBMDADB2* DLL. To register this DLL, run the following command:

```
c:\windows\SysWOW64\regsvr32 /s c:\Program_Files\IBM\SQLLIB\bin\ibmdadb2.dll
```

where *Program_Files* represents the location of the Program Files directory.

If you want your DB2 database product to have access to DB2 documentation either on your local computer or on another computer on your network, then you must install the *DB2 Information Center*. The *DB2 Information Center* contains documentation for the DB2 database system and DB2 related products. By default, DB2 information will be accessed from the web if the *DB2 Information Center* is not locally installed.

DB2 Express® Edition and DB2 Workgroup Server Edition memory limits

If you are installing DB2 Express Edition, the maximum allowed memory for the instance is 4 GB.

If you are installing DB2 Workgroup Server Edition, the maximum allowed memory for the instance is 64 GB.

The amount of memory allocated to the instance is determined by the **INSTANCE_MEMORY** database manager configuration parameter.

Important notes when upgrading from Versions 9.1 or 9.5:

- If the memory configuration for your Version 9.1 DB2 database product exceeds the allowed limit, the DB2 database product might not start after upgrading to the current version.
- The self tuning memory manager will not increase your overall instance memory limit beyond the license limits.

Installing DB2 servers using the DB2 Setup wizard (Linux and UNIX)

This task describes how to start the DB2 Setup wizard on Linux and UNIX operating systems. The DB2 Setup wizard is used to define your installation preferences and to install your DB2 database product on your system.

Before you begin

Before you start the DB2 Setup wizard:

- If you are planning on setting up a partitioned database environment, refer to “Setting up a partitioned database environment” in *Installing DB2 Servers*
- Ensure that your system meets installation, memory, and disk requirements.
- Ensure you have a supported browser installed (Firefox 2.0+, Mozilla 1.7+, or SeaMonkey 1.1.4).
- You can install a DB2 database server using either root or non-root authority. For more information about non-root installation, see “Non-root installation overview (Linux and UNIX)” in *Installing DB2 Servers*.
- The DB2 database product image must be available. You can obtain a DB2 installation image either by purchasing a physical DB2 database product DVD, or by downloading an installation image from Passport Advantage.
- If you are installing a non-English version of a DB2 database product, you must have the appropriate National Language Packages.
- The DB2 Setup wizard is a graphical installer. You must have X windows software capable of rendering a graphical user interface for the DB2 Setup wizard to run on your machine. Ensure that the X windows server is running. Ensure that you have properly exported your display. For example, export `DISPLAY=9.26.163.144:0`.
- If you are using security software in your environment, you must manually create required DB2 users before you start the DB2 Setup wizard.
- If you are planning to use the IBM Tivoli Monitoring for Databases: DB2 Agent, before installing your DB2 database product see topic “Installing IBM Tivoli Monitoring for Databases: DB2 Agent with the DB2 installer” for details, limitations and restrictions.

About this task

Restrictions

- You cannot have more than one instance of the DB2 Setup wizard running in any user account.
- The use of XML features is restricted to a database that is defined with the code set UTF-8 and has only one database partition.
- The DB2 Setup wizard fields do not accept non-English characters.
- For HP-UX 11i V2 on Itanium based HP Integrity Series Systems, users created with Setup Wizard for DB2 instance owner, fenced user, or DAS cannot be accessed with the password specified on DB2 Setup Wizard. After the setup

wizard is finished, you need to reset the password of those users. This does not affect the instance or DAS creation with the setup wizard, therefore, you do not need to recreate the instance or DAS.

Procedure

To start the DB2 Setup wizard:

1. If you have a physical DB2 database product DVD, change to the directory where the DB2 database product DVD is mounted by entering the following command:

```
cd /dvdrom
```

where */dvdrom* represents the mount point of the DB2 database product DVD.

2. If you downloaded the DB2 database product image, you must extract and untar the product file.

- a. Extract the product file:

```
gzip -d product.tar.gz
```

where *product* is the name of the product that you downloaded.

- b. Untar the product file:

On Linux operating systems

```
tar -xvf product.tar
```

On AIX, HP-UX, and Solaris operating systems

```
guntar -xvf product.tar
```

where *product* is the name of the product that you downloaded.

- c. Change directory:

```
cd ./product
```

where *product* is the name of the product that you downloaded.

Note: If you downloaded a National Language Package, untar it into the same directory. This will create the subdirectories (for example *./nlpack*) in the same directory, and allows the installer to automatically find the installation images without prompting.

3. Enter the **./db2setup** command from the directory where the database product image resides to start the DB2 Setup wizard.
4. The IBM DB2 Setup Launchpad opens. From this window, you can view installation prerequisites and the release notes, or you can proceed directly to the installation. You can also review the installation prerequisites and release notes for late-breaking information.
5. Click **Install a Product** and the **Install a Product** window will display the products available for installation.

Launch the installation by clicking **Install New**. Proceed through the installation following the DB2 Setup wizard's prompts.

Once you have initiated the installation, proceed through the DB2 Setup wizard installation panels and make your selections. Installation help is available to guide you through the remaining steps. To invoke the installation help, click **Help** or press F1. You can click **Cancel** at any time to end the installation.

Results

For non-root installations, DB2 database products are always installed in the `$HOME/sql11ib` directory, where `$HOME` represents the non-root user's home directory.

For root installations, DB2 database products are installed, by default, in one of the following directories:

AIX, HP-UX, and Solaris

`/opt/IBM/db2/V9.7`

Linux `/opt/ibm/db2/V9.7`

If you are installing on a system where this directory is already being used, the DB2 database product installation path will have `_xx` added to it, where `_xx` are digits, starting at 01 and increasing depending on how many DB2 copies you have installed.

You can also specify your own DB2 database product installation path.

DB2 installation paths have the following rules:

- Can include lowercase letters (a-z), uppercase letters (A-Z), and the underscore character (_)
- Cannot exceed 128 characters
- Cannot contain spaces
- Cannot contain non-English characters
- Cannot be symbolic links

The installation log files are:

- The DB2 setup log file. This file captures all DB2 installation information including errors.
 - For root installations, the DB2 setup log file name is `db2setup.log`.
 - For non-root installations, the DB2 setup log file name is `db2setup_username.log`, where `username` is the non-root user ID under which the installation was performed.
- The DB2 error log file. This file captures any error output that is returned by Java (for example, exceptions and trap information).
 - For root installations, the DB2 error log file name is `db2setup.err`.
 - For non-root installations, the DB2 error log file name is `db2setup_username.err`, where `username` is the non-root user ID under which the installation was performed.

By default, these log files are located in the `/tmp` directory. You can specify the location of the log files.

There is no longer a `db2setup.his` file. Instead, the DB2 installer saves a copy of the DB2 setup log file in the `DB2_DIR/install/logs/` directory, and renames it `db2install.history`. If the name already exists, then the DB2 installer renames it `db2install.history.xxxx`, where `xxxx` is 0000-9999, depending on the number of installations you have on that machine.

Each installation copy has a separate list of history files. If an installation copy is removed, the history files under this install path will be removed as well. This

copying action is done near the end of the installation and if the program is stopped or aborted before completion, then the history file will not be created.

What to do next

- Verify your installation.
- Perform the necessary post-installation tasks.

National Language Packs can also be installed by running the `./db2setup` command from the directory where the National Language Pack resides, after a DB2 database product has been installed.

On Linux x86, if you want your DB2 database product to have access to DB2 documentation either on your local computer or on another computer on your network, then you must install the *DB2 Information Center*. The *DB2 Information Center* contains documentation for the DB2 database system and DB2 related products.

DB2 Express Edition and DB2 Workgroup Server Edition memory limits

If you are installing DB2 Express Edition, the maximum allowed memory for the instance is 4 GB.

If you are installing DB2 Workgroup Server Edition, the maximum allowed memory for the instance is 64 GB.

The amount of memory allocated to the instance is determined by the `INSTANCE_MEMORY` database manager configuration parameter.

Important notes when upgrading from Versions 9.1 or 9.5:

- If the memory configuration for your Versions 9.1 or 9.5 DB2 database product exceeds the allowed limit, the DB2 database product might not start after upgrading to the current version.
- The self tuning memory manager will not increase your overall instance memory limit beyond the license limits.

Installing and configuring DB2 Text Search with a response file

You can install and configure DB2 Text Search as a part of custom silent installation of your DB2 database product. This type of installations uses the `setup` or `db2setup` command with a response file.

About this task

Perform a custom installation of your DB2 database product to install DB2 Text Search. You must add a number of keywords to your response file to have DB2 Text Search installed and configured.

Procedure

To perform a custom installation:

1. Add the following line to the response file that you are using to install your DB2 database product:
`COMP = TEXT_SEARCH`
2. To configure DB2 Text Search during the installation, add the following lines to the response file:
 - For root installations only:
`db2inst_name.TEXT_SEARCH_HTTP_SERVICE_NAME = db2j_db2inst_name`

where *db2inst_name* is the name of the DB2 instance and *db2j_db2inst_name* is the service name.

- For root installations and non-root installations:

```
db2inst_name.TEXT_SEARCH_HTTP_PORT_NUMBER = port-number
```

If you provide a value for the **TEXT_SEARCH_HTTP_SERVICE_NAME** keyword for a non-root installation, an error will be returned.

You can specify any valid service name and port number that are not in use. If you do not provide any values, default values are used for configuration if the response file keyword *db2inst_name.CONFIGURE_TEXT_SEARCH* is set to YES.

3. Install the DB2 database product using the instructions for your platform:
 - "Installing a DB2 product using a response file (Windows)" in Installing DB2 Servers
 - "Installing a DB2 product using a response file (Linux and UNIX)" in Installing DB2 Servers

What to do next

You do not have to configure DB2 Text Search immediately after installing it; you can configure it later. For instructions on how to perform the configuration later, see "Configuring DB2 Text Search" on page 41.

Installing a DB2 product using a response file (Windows)

Before you begin

Before you begin the installation, ensure that:

- Your system meets all of the memory, hardware, and software requirements to install your DB2 product.

About this task

To perform an installation from the workstation where the DB2 product will be installed:

Procedure

1. Connect to the shared directory of the network drive or DVD drive containing the DB2 installation files by entering the following command from the command prompt:

```
net use x: \\computer_name\directory_sharename /USER:domain\username
```

where:

- *x*: represents the shared directory on the local drive.
- *computer_name* represents the computer name of the remote server where the DB2 installation files are located.
- *directory_sharename* represents the share name of the directory on the network drive or DVD drive where the DB2 installation files are located.
- *domain* represents the domain where the account is defined.
- *username* represents a user that has access to this system.

For example, to use the remote db2prods directory, which was shared as DB2_Installation_Images and is located on the remote server Server, as the local x: drive, enter the following command:

```
net use x: \\Server\DB2_Installation_Images
```

Depending on how security is set up across your network, you might have to specify the */USER* parameter.

2. Enter the **setup** command as follows:

```
setup -u response_file
```

where *response_file* represents the full path and file name of the response file to use.

If you are using a response file that was created using the response file generator, ensure that all the instance profiles are located in the same drive and directory as the response file that you specify.

3. Check the messages in the log file when the installation finishes. For information about errors encountered during installation, review the installation log file located in the My Documents\DB2LOG\ directory. The log file uses the following format: DB2-ProductAbbrrev-DateTime.log, for example, DB2-ESE-Tue Apr 04 17_04_45 2007.log.

What to do next

If you want your DB2 product to have access to DB2 documentation either on your local computer or on another computer on your network, you must install the DB2 Information Center.

Installing a DB2 database product using a response file (Linux and UNIX)

Before you begin

Before you begin the installation, ensure that:

- For root installations, log on with the user ID that has root privileges. For non-root installations, log on with the user ID that is to own the DB2 installation.
- Your system meets all of the memory, hardware, and software requirements to install your DB2 database product.
- You can install a DB2 database product using a response file with either root or non-root authority.
- All DB2 processes associated with the copy of DB2 you are working with are stopped.

Procedure

To perform a response file installation:

1. Enter the **db2setup** command as follows:

```
dvd/db2setup -r responsefile_directory/response_file
```

where:

- *dvd* represents the location of the DB2 installable image;
 - *responsefile_directory* represents the directory where the customized response file is located; and
 - *response_file* represents the name of the response file.
2. Check the messages in the log file when the installation finishes. The installation logs are located, by default, in the /tmp directory:
 - db2setup.log (db2setup_Username.log for non-root installations, where *Username* is the non-root user ID under which the installation was performed)

- `db2setup.err` (`db2setup_username.err` for non-root installations, where *username* is the non-root user ID under which the installation was performed)

You can specify the location of the log file. An installation history log `db2install.history` is located in `DB2DIR/install/logs` where *DB2DIR* specifies the path where the DB2 database product was installed. The default installation directory is:

- For AIX, HP-UX or Solaris `/opt/IBM/db2/V9.7`
- For Linux `/opt/ibm/db2/V9.7`

If more than one DB2 database product is installed in the same location, you will see `db2install.history.xxxx`, where *xxxx* are digits, starting at 0001 and increasing depending on how many DB2 copies you have installed.

What to do next

If you want your DB2 database product to have access to DB2 documentation either on your local computer or on another computer on your network, then you must install the *DB2 Information Center*. The *DB2 Information Center* contains documentation for the DB2 database and related products.

Installing DB2 Text Search using `db2_install` (Linux and UNIX)

When you issue the `db2_install` command, you also install DB2 Text Search.

About this task

To install DB2 Text Search, follow the steps outlined in "Install a DB2 product using `db2_install`" in *Installing DB2 Servers*. DB2 Text Search will automatically be installed as a part of the installation of your DB2 database product.

If this is a non-root installation, a DB2 instance is created and DB2 Text Search will be installed. If this a root installation, you must create a DB2 instance and configure DB2 Text Search using one of the available methods.

You do not have to configure DB2 Text Search immediately after you install it. For instructions on how to perform the configuration, see "Configuring DB2 Text Search" on page 41.

Installing a DB2 database product using the `db2_install` or `doce_install` command (Linux and UNIX)

To install DB2 database products and features or the DB2 Information Center, extract the product image if it is a compressed file and then issue the `db2_install` or `doce_install` commands.

Before you begin

Before you install DB2 database products and features or the *DB2 Information Center*:

- You should refer to the installation documentation for the particular DB2 database product that you want to install. For example, if you want to install DB2 Enterprise Server Edition, then refer to the *Installing DB2 Servers* documentation to review installation prerequisites and other important setup information.

- You can install a DB2 database product and the *DB2 Information Center* using either root or non-root user authority.
- The DB2 database product image must be available. You can obtain a DB2 installation image either by purchasing a physical DB2 database product DVD, or by downloading an installation image from Passport Advantage.

About this task

The **db2_install** command installs DB2 database products and features on supported Linux and UNIX operating systems.

The **doce_install** command installs the *DB2 Information Center* on supported Linux operating systems.

Restrictions

You *cannot* manually install a DB2 database product or feature using an operating system's native installation utility such as **rpm**, **SMIT**, **swinstall** or **pkgadd**. Any existing scripts containing a native installation utility that you use to interface and query with DB2 database installations will need to change.

The **db2_install** command is not supported on the National Language Package DVD.

Only one copy of the *DB2 Information Center* for the current release can be installed on your system. The Information Center cannot be installed in the same location where a DB2 database product is installed. If you install the *DB2 Information Center* on a system with a firewall, and you plan to allow other systems to access the *DB2 Information Center*, you must open the port in your firewall setting.

Procedure

To install a DB2 database product or feature using the **db2_install** command, or to install the *DB2 Information Center* using the **doce_install** command:

1. If you have a physical DB2 database product DVD, insert and mount the appropriate DVD or access the file system where the installation image was stored.
2. If you downloaded the DB2 database product image, you must extract and untar the product file.

- a. extract the product file:

```
gzip -d product.tar.gz
```

For example,

```
gzip -d ese.tar.gz
```

- b. Untar the product file:

On Linux operating systems

```
tar -xvf product.tar
```

For example,

```
tar -xvf ese.tar
```

On AIX, HP-UX, and Solaris operating systems

```
gntar -xvf product.tar
```

For example,
gnutar -xvf ese.tar

- c. Change directory into the product directory:

```
cd product
```

For example,
cd ese

3. Enter the `./db2_install` or `./doce_install` command:

```
./db2_install -b DB2DIR -p productShortName -c NLPackLocation -L language... -n
```

where:

- *DB2DIR* specifies the path where the DB2 database product will be installed. If the path is not specified, you are prompted to either select the default path or to provide the path. The default installation path is:
 - for AIX, HP-UX or Solaris operating systems: `/opt/IBM/db2/V9.7`
 - for Linux operating systems: `/opt/ibm/db2/V9.7`
 - for the *DB2 Information Center*: `/opt/ibm/db2ic/V9.7`. The Workstation version of the DB2 Information Center does not have a default installation path, you must specify the installation location. However, by default, the Workstation version of the DB2 Information Center is installed at port 51097.

If you provide your own path, you must specify the full path name.

DB2 installation paths have the following rules:

- Can include lowercase letters (a–z), uppercase letters (A–Z), and the underscore character (_)
- Cannot exceed 128 characters
- Cannot contain spaces
- Cannot contain non-English characters
- Cannot be symbolic links

Note: For DB2 database products and components to work together, they must be installed to a single path. This is not to be confused with the ability to install DB2 database products to multiple paths. But, for products and components to work together, they must be installed to the same path, and, must be at the same release level.

- *productShortName* specifies the DB2 database product to be installed. This parameter is case insensitive and is mandatory when the `-n` parameter is specified. The product short name (*productShortName*) can be found in the file `ComponentList.htm` (under the product full name) located in the `/db2/plat` directory on your media where *plat* is the platform name that you are installing on. You can only install one product at a time.
- *NLPackLocation* specifies the National Language Pack (NLPACK) location.
- *language* specifies national language support. You can install a non-English version of a DB2 database product. However, you must run this command from the product DVD, not the National Language pack DVD.

By default, English is always installed, therefore, English does not need to be specified. When more than one language is required this parameter is mandatory. To indicate multiple languages, specify this parameter multiple times. For example, to install both French and German specify `-L FR -L DE`.

- *n* parameter indicates noninteractive installation mode. When this parameter is specified, both **-b** and **-p** must also be specified. You only need to specify **-c** and **-L** if applicable.

When installing the *DB2 Information Center*, if you specify a port number other than the default, you might receive the error The service name specified is in use. You can correct this error by either choosing to use the default port number or by specifying a different service name.

What to do next

After installation, you must manually configure your DB2 database server. Tasks include user and instance creation and configuration.

Configuring DB2 Text Search

Your options for configuring DB2 Text Search depend on whether you are performing the initial configuration or a reconfiguration and which platform you are using.

Before you begin

Before reconfiguring DB2 Text Search, stop the text search instance services, as outlined in “Starting the DB2 Text Search instance service” on page 49.

Procedure

- Determine whether DB2 Text Search is configured.

Run the Configuration Tool by issuing the following command:

```
configTool printAll -configPath absolute-path-to-config-folder
```

In the output of the **printAll**, option, the authentication token is an empty string if DB2 Text Search is not configured.

- Configure DB2 Text Search for the first time.

On Linux and UNIX operating systems, use one of the following methods to configure DB2 Text Search:

- Rerun the silent installation as described in “Installing and configuring DB2 Text Search with a response file” on page 35.
- Rerun the GUI installation as described in “Installing and configuring DB2 Text Search with the DB2 Setup Wizard” on page 28.
- Use the Configuration Tool. Refer to “Configuring DB2 Text Search using the Configuration Tool” on page 42. Note that using the Configuration Tool to perform a manual installation requires you to manually configure all of the parameters, whereas using the installer requires you to configure only two parameters.
- Issue the **db2isetup** command. Note that on Linux and UNIX operating systems, if you create an instance using **db2icrt**, DB2 Text Search is not automatically configured.

On Windows operating systems, use one of the following methods to configure DB2 Text Search:

- Rerun the silent installation as described in “Installing and configuring DB2 Text Search with a response file” on page 35.
- Rerun the GUI installation as described in “Installing and configuring DB2 Text Search with the DB2 Setup Wizard” on page 28.
- Issue the **db2icrt**, **db2iupdt**, or **db2iupgrade** command.

- Determine whether the Java developer kit is from IBM.

The DB2 Text Search internally uses a Java developer kit whose location is pointed by **JDK_PATH** of `db2 get dbm cfg` command and this Java developer kit has to come from IBM. To verify if the Java developer kit is from IBM, run the following command:

```
JDK_PATH/jre/bin/java -version
```

This command will display the Java version information and IBM should display as part of string if the Java developer kit is from IBM.

- Re-configure DB2 Text Search.

After you have configured DB2 Text Search, you cannot use the GUI installer to re-configure it. You must make any updates to the configuration manually.

On Linux and UNIX operating systems, use one of the following methods to re-configure DB2 Text Search:

- Rerun the silent installation as described in “Installing and configuring DB2 Text Search with a response file” on page 35.
- Use the Configuration Tool. Refer to “Configuring DB2 Text Search using the Configuration Tool.”
- Issue the **db2isetup** command with the **-r** option.

On Windows operating systems, use one of the following methods to re-configure DB2 Text Search:

- Rerun the silent installation as described in “Installing and configuring DB2 Text Search with a response file” on page 35.
- Use the Configuration Tool. Refer to “Configuring DB2 Text Search using the Configuration Tool.”
- Run the **db2icrt**, **db2iupdt**, or **db2iupgrade** command, specifying the **-j** option as shown to meet your needs:
 - **-j "TEXT_SEARCH"** attempts to configure DB2 Text Search with the default service name and a generated port value.
 - **-j "TEXT_SEARCH, [servicename]"** reserves the service name with an automatically generated port number or with the same port number assigned to that service name if the service name is already reserved in the services file.
 - **-j "TEXT_SEARCH, [port number]"** reserves the port with the default service name.
 - **-j "TEXT_SEARCH, [servicename], [port#]"** reserves the specified service name and port number.

Note: On Windows operating systems, the **PATH** in the DB2 command window points to *current-default-copy-install-path\db2tss\bin*, so to configure an instance that is not in the current DB2 copy, first switch to the appropriate DB2 command window for that copy.

Configuring DB2 Text Search using the Configuration Tool

The Configuration Tool is a command-line tool that you can use to perform the initial configuration of DB2 Text Search or to change the current configuration.

Before you begin

To customize most of the configuration settings, you must stop the DB2 Text Search instance services.

About this task

For the initial configuration, the recommended method is to have it done automatically by the DB2 installer. However, if you want to manually re-configure DB2 Text Search, the Configuration Tool is an effective means of doing so.

Procedure

To perform the initial configuration of DB2 Text Search:

1. Run the **configTool** command with the **configureParams** option to set the configuration path, log path, temporary directory path, and installation path values.

- Specify the options as shown:

```
configTool configureParams
-configPath absolute-path-to-config-folder
-logPath path
-tempDirPath path
-installPath path
-maxHeapSize value
```

Where the **logPath** specifies the log path to be used by DB2 Text Search, the **tempDirPath** specifies the temporary directory path to be used by DB2 Text Search, the **installPath** specifies the path to DB2 Text Search install directory which is `DB2PATH\db2tss` on Windows and the `DB2DIR/db2tss` directory on Linux and UNIX, and the **maxHeapSize** specifies the size of the heap that will be used. `DB2DIR` is the location of the DB2 copy.

Note: Starting from DB2 Version 9.7 Fix Pack 4, **maxHeapSize** has been renamed to **startupHeapSize**.

- On Windows operating systems, specify the command as shown. You need to specify only **configPath**; all of the other parameters are assigned default paths and values.

```
configTool
-configPath absolute-path-to-config-folder
```

2. Generate the authentication token by issuing the **configTool** command with the **generateToken** parameter, as follows:

```
configTool generateToken
-configPath absolute-path-to-config-folder
-seed myInstanceName
```

DB2 Text Search authenticates text search index administration and text search requests by using an authentication token. You do not have to remember the authentication token or the authentication key because the text search instance services use the token internally.

3. Specify the HTTP port by issuing the **configTool** command with the **configureHTTPListener** parameter, as follows:

```
configTool configureHTTPListener
-configPath absolute-path-to-config-folder
-adminHTTPPort port-number
```

Note: The value of the port should be between 1024 and 65535.

The administrative HTTP port allows communication between text search processes using TCP/IP. During the installation of a DB2 database product or during instance creation, you can specify a service name and port if you have root authority. These are used for updating the services file.

4. Update the services file.

Refer to “Updating the services file on the server for TCP/IP communications.” When you use the Configuration Tool for configuration, the tool does not update the services file. Therefore, you must update the services file manually,

Note: Only root users can update the services file. Non-root users must have the system administrator run the **db2rfe** command first.

Updating the services file on the server for TCP/IP communications

This task is part of the main task of *Configuring TCP/IP communications for a DB2 instance*.

About this task

The TCP/IP services file specifies the ports that server applications can listen on for client requests. If you specified a service name in the *svcename* field of the DBM configuration file, the services file must be updated with the service name to port number/protocol mapping. If you specified a port number in the *svcename* field of the DBM configuration file, the services file does *not* need to be updated.

Update the services file and specify the ports that you want the server to listen on for incoming client requests. The default location of the services file depends on the operating system:

Linux and UNIX operating systems

/etc/services

Windows operating systems

%SystemRoot%\system32\drivers\etc\services

Procedure

Using a text editor, add the Connection entry to the services file. For example:

```
db2c_db2inst1 3700/tcp # DB2 connection service port
```

where:

db2c_db2inst1

represents the connection service name

3700 represents the connection port number

tcp represents the communication protocol that you are using

Results

Chapter 4. Upgrading text search

Upgrading DB2 Text Search for administrator or root install

To obtain the latest functionality upgrade your DB2 Text Search instance. You need to upgrade the DB2 server, instance, and all databases when upgrading the text search instance.

About this task

The following steps describe the process to upgrade DB2 Text Search Version 9.5 or Version 9.7 root installations on Linux or UNIX platforms, or for administrators on the Windows platform.

Procedure

1. Back up values for all configurable properties of DB2 Text Search and the database servers. Use the following commands:

```
configTool printAll -configPath configuration-directory > db2tss_config.out
adminTool version -configPath configuration-directory >> db2tss_config.out
```
2. If you enabled DB2 Text Search for rich text document support and are upgrading from DB2 Version 9.5 Fix Pack 3 or later, disable rich text document support. Issue the following command:

```
richtextTool disable <DB2-install-directory>
```
3. Log in as the instance owner or a user with SYSADM authority.
4. Stop the DB2 instance and the DB2 Text Search instance service.
5. Back up the DB2 Text Search configuration directory, located under *INSTHOME/db2tss/config*.
6. Log on to the DB2 server as root on Linux and UNIX operating systems or user with Local Administrator authority on Windows operating systems. If you are upgrading a multi-partitioned instance, you must perform instance upgrade from the instance-owning partition.
7. Install a new copy of the DB2 version you are upgrading to and perform a custom installation. Ensure the text search component is selected. DB2 Text Search is an optional component that is only available when you select a custom installation.
8. Upgrade your instances by issuing the **db2iupgrade** command:

```
db2 iupgrade /j "text_search [[,service-name]][[,port-number]]"
```
9. Upgrade the existing databases by issuing **DB2 UPGRADE DATABASE** command.
10. Upgrade DB2 Text Search server for your instances by issuing the **configTool upgradeInstance** command.
 - For Linux and UNIX operating systems:

```
configTool upgradeInstance
  -installedConfigPath $DB2DIR/cfg/db2tss/config
  -configPath $INSTHOME/db2tss/config
```

where, *INSTHOME* is the instance home directory and *DB2DIR* is the location of the DB2 Version 9.7 copy.

- For Windows operating systems:

```
configTool upgradeInstance
  -installedConfigPath "<DB2PATH>\CFG\DB2TSS\CONFIG"
  -configPath "<INSTPROFDIR>\instance-name\DB2TSS\CONFIG"
```

where, *DB2PATH* is the location of the latest DB2 copy and *INSTPROFDIR* is the instance profile directory.

11. Review the values for all DB2 Text Search configurable properties. Compare with the values that you backed up to ensure that they have correct values. issue the following command to check the configuration values:

```
configTool printAll -configPath <configuration-directory>
```
12. Starting with DB2 Version 9.7 Fix Pack 4, you must set up the DB2 Accessories Suite package for rich text or proprietary format support.
 - a. Download and install the DB2 Accessories Suite package (Oracle's Outside In Technology (OIT)). For more information, see the topic about installing the DB2 Accessories Suite.
 - b. Run the **richtextTool** utility to setup rich text document support as a system administrator or user with administrative permissions.
13. If you disabled DB2 Text Search for rich text document support earlier, set up and enable rich text document support now. Use the following commands to set up and enable rich text document support:

```
richtextTool setup <DB2DIR> <OIT-zipfile-directory>
richtextTool enable <DB2DIR>
```

where, *DB2DIR* is the location of the DB2 Version 9.7 copy, and *OIT-zipfile-directory* is the directory that contains the OIT (outside in filter) library zip files.

14. Verify that the upgrade was successful by starting the DB2 Text Search instance service. If you disabled rich text document support, verify that rich text document support is enabled by issuing text search queries and compare with pre-upgrade results.

Upgrading DB2 Text Search for non-root install (Linux and UNIX)

To obtain the latest functionality upgrade your DB2 Text Search instance. You need to upgrade the DB2 server, instance, and all databases when upgrading the text search instance.

About this task

The following steps describe the process to upgrade DB2 Text Search Version 9.5 or Version 9.7 non-root installations on Linux or UNIX platforms.

Procedure

1. Back up values for all configurable properties of DB2 Text Search and the database servers. You can use the following two commands:

```
configTool printAll -configPath configuration-directory > db2tss_config.out
adminTool version -configPath configuration-directory >> db2tss_config.out
```
2. If you enabled DB2 Text Search for rich text document support and are upgrading from DB2 Version 9.5 Fix Pack 3 or later, disable rich text document support. Issue the following command:

```
richtextTool disable <DB2-install-directory>
```
3. Log in as the instance owner or a user with SYSADM authority. Then stop the DB2 instance and the DB2 Text Search instance service.

4. Back up the DB2 Text Search configuration directory under `sqllib/db2tss/config`.
5. Log on to the DB2 server as a non-root user. Review the database instance type to ensure it can be upgraded as a non-root installation.
6. Install a new copy of DB2 and select the upgrade option. If you specified the **-f nobackup** parameter and the DB2 database product installation fails, manually install the DB2 database product and run the **db2nrupgrade** command to upgrade the non-root instance. You must select the DB2 Text Search component from the feature tree. Issue the following command:
`db2nrupg -b <BackupDir>`

of the non-root installation before the upgrade. *BackupDir* is the backup directory for the configuration file.

7. Upgrade the existing databases by issuing **upgrade database** command.
8. Enable root-based features. You might ask a system administrator with root access to issue the **db2rfe** command
9. Upgrade DB2 Text Search server for your instances by issuing the **configTool upgradeInstance** command.

- For Linux and UNIX operating systems:

```
configTool upgradeInstance
  -installedConfigPath $DB2DIR/cfg/db2tss/config
  -configPath $INSTHOME/sqllib/db2tss/config
```

where, *INSTHOME* is the instance home directory and *DB2DIR* is the location of the DB2 Version 9.7 copy.

10. Review the values for all DB2 Text Search configurable properties. Compare with the values that you backed up to ensure that they have correct values. issue the following command to check the configuration values:
`configTool printAll -configPath <configuration-directory>`
11. Starting with DB2 Version 9.7 Fix Pack 4, you must set up the DB2 Accessories Suite package for rich text or proprietary format support.
 - a. Download and install the DB2 Accessories Suite package (Oracle's Outside In Technology (OIT)). For more information, see the topic about installing the DB2 Accessories Suite.
 - b. Run the **richtextTool** utility to setup rich text document support as a system administrator or user with administrative permissions.
12. If you disabled DB2 Text Search for rich text document support earlier, set up and enable rich text document support now. Use the following command to set up and enable rich text document support:
`richtextTool setup <DB2DIR> <OIT-zipfile-directory>`
`richtextTool enable <DB2DIR>`

where, *DB2DIR* is the location of the DB2 Version 9.7 copy, and *OIT-zipfile-directory* is the directory that contains the OIT (outside in filter) library zip files.

13. Verify that the upgrade was successful by starting the DB2 Text Search instance service. If you disabled rich text document support, verify that rich text document support is enabled by issuing text search queries and compare with pre-upgrade results.

Chapter 5. Administering and maintaining text search indexes

Starting the DB2 Text Search instance service

Before you can enable your database for text search, you must start the DB2 Text Search instance service.

About this task

To start the DB2 Text Search instance service, enter the following command:

```
db2ts "START FOR TEXT"
```

The index update service is started when you issue **db2ts START FOR TEXT**.

Enabling a database for DB2 Text Search

You must enable each database that contains columns of text to be searched. You can enable a database for DB2 Text Search by using the **db2ts ENABLE DATABASE FOR TEXT** command or the `SYSPROC.SYSTS_ENABLE` stored procedure.

Before you begin

You must start the DB2 instance service.

The authorization ID of the statement must hold DBADM and DATAACCESS authority.

About this task

When you enable a database, you can use the following views to get information about the text search indexes in the database and their properties:

SYSIBMTS.TSDEFAULTS

Shows the database default values for index, text, and processing characteristics

SYSIBMTS.TSLOCKS

Shows information about command locks set at the database and index level

SYSIBMTS.TSINDEXES

Shows all text search indexes and their settings

SYSIBMTS.TSCONFIGURATION

Shows the index configuration parameters

SYSIBMTS.TSCOLLECTIONNAMES

Shows the collection names for each index

After you enable a database for text search, it remains enabled until you explicitly disable it.

To prepare the database for use with DB2 Text Search, use one of the following methods:

- Enter the following command:

```
db2ts "ENABLE DATABASE FOR TEXT CONNECT TO databaseName"
```

You must do this step only once for each database. You do not have to enable a database each time that you stop and restart the instance services.

For example, to enable a database named SAMPLE, enter the following command:

```
db2ts "ENABLE DATABASE FOR TEXT CONNECT TO SAMPLE"
```

- Call one of the administrative SQL routines, as follows:
 - CALL SYSPROC.SYSTS_ADMIN_CMD ('ENABLE DATABASE FOR TEXT', 'en_US', ?)
 - CALL SYSPROC.SYSTS_ENABLE('en_US', ?)

Installing DB2 Accessories Suite for DB2 Text Search Version 9.7 FP3 or earlier

DB2 Accessories Suite provides the filter libraries required for the DB2 Text Search rich text document support functionality.

Before you begin

To install DB2 Accessories Suite you must:

- Download DB2 Accessories Suite (V8.2.2 of Oracle's Outside In Technology Search Export SDK and HTML Export SDK). For the download link, see <https://www.ibm.com/services/forms/preLogin.do?source=swg-dm-db2accsuite>

About this task

It is imperative that the filter libraries provided within DB2 Accessories Suite are installed according to the instructions within this task. The rich text document support feature will only function with the proper filter libraries in place.

Restrictions

In order to install DB2 Accessories Suite on Linux and UNIX, you must be logged on to the DB2 server as a system administrator. On Windows, you must be logged on as a user with Local Administrator authority.

Procedure

To install DB2 Accessories Suite:

1. Stop the DB2 Text Search instance service. To stop the service, run **db2ts "STOP FOR TEXT"**. For more information about this command, see "Stopping the DB2 Text Search instance service" on page 56.
2. Log on to the DB2 server with a user having permissions as described in the Restrictions section.
3. Decompress the package. The following files are provided:
 - For Linux and UNIX operating systems:

```
db2_accsuite_license
db2-install_directory
db2-install_directory/db2tss
db2-install_directory/db2tss/lib
db2-install_directory/db2tss/lib/ciemsg
db2-install_directory/db2tss/bin
db2-install_directory/db2tss/bin/richtextTool.sh
oit_sdk
```

```

oit_sdk/<hx_sdk_file>
oit_sdk/<sx_sdk_file>
readme.txt
setup.sh

```

The file in the oit_sdk directory are platform dependent.

| Platform | <hx_sdk_file> | <sx_sdk_file> |
|-----------------|---------------|---------------|
| AIX 64bit | V13087-01.zip | V13119-01.zip |
| Linux 32bit | V13125-01.zip | V13132-01.zip |
| Linux 64bit | V13094-01.zip | V13126-01.zip |
| HP-UX IA64 | V13088-01.zip | V13121-01.zip |
| SunOS x86 64bit | V13096-01.zip | V13128-01.zip |
| SunOS SPARC | V13095-01.zip | V13127-01.zip |

- For Windows operating systems:

```

db2_accsuite_license
db2-install_directory
db2-install_directory\db2tss
db2-install_directory\db2tss\lib
db2-install_directory\db2tss\lib\ciemsg.exe
db2-install_directory\db2tss\bin
db2-install_directory\db2tss\bin\richtextTool.bat
oit_sdk/<hx_sdk_file>
oit_sdk/<sx_sdk_file>
readme.txt
setup.bat

```

The file in the oit_sdk directory are platform dependent:

| Platform | <hx_sdk_file> | <sx_sdk_file> |
|----------------|---------------|---------------|
| Windows 32-bit | V13098-01.zip | V13130-01.zip |
| Windows 64-bit | V13099-01.zip | V13131-01.zip |

4. Check for required files and packages.

- For Linux and UNIX operating systems, check that the following exist:

```

DB2DIR/db2tss/lib/ciemsg
DB2DIR/db2tss/bin/richtextTool.sh

```

where *DB2DIR* is the location of the DB2 Version 9.7 copy.

If the db2tss directory is missing, DB2 Text Search has not been properly installed and configured.

If the db2tss directory exists, but the required files and packages are missing, copy ciemsg from the package to *DB2DIR/db2tss/lib*, and copy richtextTool.sh to *DB2DIR/db2tss/bin* where *DB2DIR* is the location of the DB2 Version 9.7 copy. After copying ciemsg, the **db2chgpath** command must be issued.

Example:

```

cp ./db2-install_directory/db2tss/lib/ciemsg DB2DIR/db2tss/lib/
cp ./db2-install_directory/db2tss/bin/richtextTool.sh DB2DIR/db2tss/bin/
DB2DIR/install/db2chgpath -f db2tss/lib/ciemsg

```

- For Windows operating systems, check that the following exist:

```

DB2PATH\db2tss\lib\ciemsg.exe
DB2PATH\db2tss\bin\richtextTool.bat

```

where *DB2PATH* is the location of the DB2 Version 9.7 copy.

If the db2tss directory is missing, DB2 Text Search has not been properly installed and configured.

If the `db2tss` directory exists, but the required files and packages are missing, copy `ciemsg.exe` from the package to `DB2PATH\db2tss\lib`, and copy `richtextTool.bat` to `DB2PATH\db2tss\bin` where `DB2PATH` is the location of the DB2 Version 9.7 copy.

Example:

```
copy .\db2-install_directory\db2tss\lib\ciemsg.exe DB2PATH\db2tss\lib\  
copy .\db2-install_directory\db2tss\bin\richtextTool.bat DB2PATH\db2tss\bin\  

```

5. For Linux and UNIX operating systems, ensure execute permission has been set on `ciemsg` and `richtextTool.sh`, so that the instance owner can run them.

6. Run the setup.

- For Linux and UNIX operating systems:

```
OIT_extract_directory/setup.sh <DB2DIR>
```

where `OIT_extract_directory` is where you unpacked the DB2 Accessories Suite, and `<DB2DIR>` is the location of the DB2 Version 9.7 copy.

- For Windows operating systems:

```
OIT_extract_directory\setup.bat "<DB2PATH>"
```

where `OIT_extract_directory` is where you unpacked the DB2 Accessories Suite, and `<DB2PATH>` is the location of the DB2 Version 9.7 copy.

Results

You have successfully installed DB2 Accessories Suite.

What to do next

You can now setup rich text document support for DB2 Text Search. See, "Setting-up DB2 Text Search for rich text document support" on page 54 for more details.

Installing DB2 Accessories Suite for DB2 Text Search Version 9.7 FP4 or later

DB2 Accessories Suite enables indexing and search for documents with rich text and proprietary formats with DB2 Text Search.

Before you begin

To install DB2 Accessories Suite you must:

- Download DB2 Accessories Suite for Version 9.7 FP4. For the download link, see: <https://www.ibm.com/services/forms/preLogin.do?source=swg-dm-db2accsuite>

Ensure you download the most up-to-date version of the suite.

About this task

It is imperative that the matching DB2 Accessories Suite fix pack is installed to ensure proper functioning of the feature.

Restrictions

In order to install DB2 Accessories Suite on Linux and UNIX, you must be logged on to the DB2 server as a system administrator. On Windows, you must be logged on as a user with Local Administrator authority.

Procedure

To install DB2 Accessories Suite:

1. Stop the DB2 Text Search instance service. To stop the service, run **db2ts "STOP FOR TEXT"**. For more information about this command, see “Stopping the DB2 Text Search instance service” on page 56.
2. Log on to the DB2 server with a user having permissions as described in the Restrictions section.
3. Uncompress the package.
4. There are two installation modes. One is console installation, while the other is silent installation.
 - To complete a console install:
 - a. Run the InstallDB2AccSuite97 program.
 - b. After accepting the license, enter the location of the /db2tss subdirectory in the DB2 Version 9.7 copy when prompted for the install path.
 - c. The db2tss directory must already exist. If it is missing, DB2 Text Search has not been properly installed and configured.
 - d. Review the summary and confirm the installation.
 - To complete a silent install:
 - a. You can silently install DB2 Accessories Suite by running the install program with the option **-i** and using **-f** to specify the response file that contains the settings.
 - b. Set the acceptance to YES to accept the license in the response file.

Results

You have successfully installed DB2 Accessories Suite.

What to do next

You can now enable rich text document support for DB2 Text Search. See, “Enabling DB2 Text Search for rich text document support” on page 55 for more details.

Uninstalling the DB2 Accessories Suite for DB2 Text Search

You can uninstall a stand-alone DB2[®] Text Search by using the **Uninstall_DB2AS** command.

Before you begin

In order to uninstall DB2 Accessories Suite on Linux and UNIX platforms, you must be logged on to the DB2 database server as a system administrator. On Windows platforms you must be logged on as a user with Local Administrator authority.

Procedure

To uninstall DB2 Accessories Suite:

1. Stop the DB2 Text Search instance service. To stop the service, run db2ts "STOP FOR TEXT".
2. Log on to the DB2 database server with as a user who has the necessary privileges for the operating system.
3. Disable rich text document support for all text search instances which were enabled with rich text feature before. For details, see the topic about disabling DB2 Text Search for rich text document support.
4. Uninstall DB2 Accessories Suite installer. To uninstall the installer:
 - On Linux and UNIX operating systems:
`<DB2DIR>/db2tss/Uninstall_DB2AS/Uninstall_DB2AS.bin`

where `<DB2DIR>` is the location of the latest DB2 copy.
 - On the Windows operating system:
`<DB2PATH>\db2tss\Uninstall_DB2AS\Uninstall_DB2AS.exe`

where `<DB2PATH>` is the location where you installed the latest DB2 copy.

Results

You have uninstalled the DB2 Accessories Suite.

Setting-up DB2 Text Search for rich text document support

DB2 Text Search servers must be properly configured before rich text documents are supported.

Before you begin

To setup rich text document support for DB2 Text Search servers you must:

1. Download and install the DB2 Accessories Suite package (Oracle's Outside In Technology (OIT)). For more information, see "Installing DB2 Accessories Suite for DB2 Text Search Version 9.7 FP3 or earlier" on page 50.
2. As a system administrator or user with administrative permissions, run the `richtextTool` utility to setup rich text document support.

About this task

The `richtextTool` utility sets-up, enables, and disables rich text document support for DB2 Text Search. The tool must be run before full-text indexes can be created for rich text document processing.

Restrictions

In order to run **richtextTool setup** on Linux and UNIX, you must be logged on to the DB2 server as a system administrator or a user with write and execute permissions on the DB2 install directory. In order to run **richtextTool setup** on Windows, you must be logged on as a user with Local Administrator authority. In order to run **richtextTool setup**, you must have the decompression command installed. To obtain the unzip package for AIX, see <http://www-03.ibm.com/systems/power/software/aix/linux/toolbox/alpha.html>

You must be able to write to the DB2 installation directory.

Procedure

1. Stop the DB2 Text Search instance service. To stop the service, run **db2ts "STOP FOR TEXT"**. For more information about this command, see "Stopping the DB2 Text Search instance service" on page 56.
2. Log on to the DB2 server with a user having permissions as described in the Restrictions section.
3. Run the `richtextTool` utility to setup support. To setup support:

- For Linux and UNIX operating systems:

```
$INSTHOME/sql1lib/db2tss/bin/richtextTool setup <DB2DIR> <OIT zipfile-directory>
```

where *INSTHOME* is the instance home directory, *<DB2DIR>* is the location of the DB2 Version 9.7 installation copy, and *<OIT zipfile-directory>* contains the OIT (search and HTML export) library compressed files.

Note: If DB2 is installed by a root user, you need to run the `richtextTool` utility setup as a root user. Otherwise, you need to run the `richtextTool` utility setup using an instance owner user ID.

- For Windows operating systems:

```
DB2PATH\db2tss\bin\richtextTool.bat setup <DB2PATH> <OIT zipfile-directory>
```

where *<DB2PATH>* is the location of the DB2 Version 9.7 copy and *<OIT zipfile-directory>* contains the OIT(search and html export) library zip files.

4. Enable support for rich text. For instructions on enabling support for rich text, see "Enabling DB2 Text Search for rich text document support."

Results

You have setup rich text support for a DB2 Text Search server.

Enabling DB2 Text Search for rich text document support

Rich text support can be enabled on properly configured DB2 Text Search servers.

Before you begin

The instance owner must hold DBADM with DATAACCESS authority. To enable rich text document support for DB2 Text Search servers you must, as the instance owner, run the **richtextTool** utility with the enable option.

Starting from DB2 Version 9.7 FP3 and earlier, before enabling rich text document support, each DB2 Text Search server must be setup for rich text document support. For more details see, "Setting-up DB2 Text Search for rich text document support" on page 54.

Starting from DB2 Version 9.7 FP4 and later, before enabling rich text document support, each DB2 Text Search server must be prepared for rich text document support. For more information, see "Installing DB2 Accessories Suite for DB2 Text Search Version 9.7 FP4 or later" on page 52

About this task

Restrictions

In order to run **richtextTool enable**, you must be logged on as the instance owner.

Procedure

1. Log on as the instance owner.
2. Stop the DB2 Text Search instance service. To stop the service, run **db2ts "STOP FOR TEXT"**. For more information about this command, see "Stopping the DB2 Text Search instance service."
3. Run the richtextTool utility to enable support. To enable support:
 - For Linux and UNIX operating systems:

```
$INSTHOME/sqllib/db2tss/bin/richtextTool enable <DB2DIR>
```

where *INSTHOME* is the instance home directory and *<DB2DIR>* is the location of the DB2 Version 9.7 copy.
 - For Windows operating systems:

```
DB2PATH\db2tss\bin\richtextTool.bat enable <DB2PATH>
```

where *DB2PATH* is the location where you installed your DB2 Version 9.7 copy.
4. Start the DB2 Text Search instance service. To start the service, run **db2ts "START FOR TEXT"**. For more information about this command, see "Starting the DB2 Text Search instance service" on page 49.

Results

You have enabled rich text support for a DB2 Text Search server.

Stopping the DB2 Text Search instance service

When you stop the DB2 Text Search instance service, any currently running **db2ts** commands will be completed, but you cannot issue any new ones.

About this task

To stop the DB2 Text Search instance service, enter the following command:

```
db2ts "STOP FOR TEXT"
```

Disabling a database for DB2 Text Search

Disable a database when you no longer intend to perform text searches in that database.

About this task

When you disable a database for text search, you also delete the following objects from the SYSIBMTS schema:

- SYSIBMTS.TSDEFAULTS
- SYSIBMTS.TSLOCKS
- SYSIBMTS.TSINDEXES
- SYSIBMTS.TSCONFIGURATION
- SYSIBMTS.TSCOLLECTIONNAMES

Procedure

To disable a database for text search, use one of the following methods:

1. Drop any text search indexes defined in the database, using the **DROP INDEX** command.
2. To disable a database for text search, use one of the following methods:
 - Issue the **DISABLE DATABASE FOR TEXT** command:
`db2ts "DISABLE DATABASE FOR TEXT CONNECT TO databaseName"`
 - Call the SYSPROC.SYSTS_DISABLE procedure:
 - `CALL SYSPROC.SYSTS_DISABLE('en_US', ?)`

Note: Text search indexes can also be dropped using the **FORCE** option. However, it is possible that some data, specifically a text search collection, will remain after you disable the database. This can occur because the **FORCE** option allows you to drop text search indexes even if the DB2 Text Search server cannot be reached.

Deleting orphaned text search collections

One of the administrative tasks that you might have to do when using DB2 Text Search is deleting orphaned collections. You can delete orphaned collections using the Administration Tool.

About this task

A text search collection refers to the underlying representation of a text search index. There is a one-to-one relationship between a text search collection and a text search index. Although the **db2ts** commands operate on text search indexes, the text search tools operate on the text search collections. When a text search index no longer exists but its corresponding text search collection does, it is called an orphaned collection.

You also create an orphaned collection by performing any of the following operations when the text search server is not reachable:

- Dropping a table or dropping a database
The DROP TABLE statement and the **DROP DATABASE** command do not automatically drop text search collections.
- Using the **FORCE** option with the **DISABLE** operation, and the **DROP** index operation.

All of these commands or stored procedures succeed even if the text search server is not reachable.

To determine whether any orphaned collections exist:

1. Use the Administration Tool to report all text search collections, as follows:
`adminTool status -configPath absolute-path-to-config-folder`
2. Query the SYSIBMTS.TSCOLLECTIONNAMES administrative view to report all text search indexes on the current database, as follows:
`SELECT collectionname FROM SYSIBMTS.TSCOLLECTIONNAMES`

Perform this query on all the databases enabled for DB2 Text Search, and combine the results into a list.

Note: The Administration Tool lists all text search collections, whereas the query on SYSIBMTS.TSCOLLECTIONNAMES lists only text search indexes on the current database.

3. Compare the lists returned by the Administration Tool and by the SELECT statement. Any text search collection returned by the Administration Tool but not by the SELECT statement is an orphaned collection. The only exception to this is the "Default" collection that is created when DB2 Text Search is started.

To remove the orphaned text search collections, use the following command:

```
adminTool delete -configPath absolute-path-to-config-folder
                -collectionName collection-name
```

Example

A user currently has DB2 Text Search enabled for a database called DBCP1208, which is running on a UNIX system. To determine whether any orphaned text search collections exist, the user employs the Administration Tool and a SELECT statement:

```
adminTool.sh status -configPath $HOME/sql/lib/db2tss/config
```

| CollectionName | IndexSize | NumOfDocuments | |
|----------------------------------|-----------|----------------|----|
| Default | 13,159B | 0 | |
| tigertail_DBCP1208_TS542717_0000 | | 13,159B | 11 |
| tigertail_DBCP1208_TS012817_0000 | | 13,159B | 17 |
| tigertail_DBCP1208_TS082817_0000 | | 13,159B | 16 |
| tigertail_DBCP1208_TS152817_0000 | | 13,159B | 18 |
| tigertail_DBCP1208_TS212817_0000 | | 13,159B | 16 |
| tigertail_DBCP1208_TS302817_0000 | | 13,159B | 17 |
| tigertail_DBCP1208_TS392817_0000 | | 13,159B | 10 |
| tigertail_DBCP1208_TS462817_0000 | | 13,159B | 10 |
| tigertail_DBCP1208_TS542817_0000 | | 13,159B | 12 |
| tigertail_DBCP1208_TS022917_0000 | | 13,159B | 10 |
| tigertail_DBCP1208_TS112917_0000 | | 13,159B | 16 |
| tigertail_DBCP1208_TS192917_0000 | | 13,159B | 11 |
| tigertail_DBCP1208_TS262917_0000 | | 13,159B | 12 |
| tigertail_DBCP1208_TS867530_0000 | | 13,159B | 16 |

```
db2 select collectionname from sysibmts.tscollectionnames
```

```
COLLECTIONNAME
-----
tigertail_DBCP1208_TS542717_0000
tigertail_DBCP1208_TS012817_0000
tigertail_DBCP1208_TS082817_0000
tigertail_DBCP1208_TS152817_0000
tigertail_DBCP1208_TS212817_0000
tigertail_DBCP1208_TS302817_0000
tigertail_DBCP1208_TS392817_0000
tigertail_DBCP1208_TS462817_0000
tigertail_DBCP1208_TS542817_0000
tigertail_DBCP1208_TS022917_0000
tigertail_DBCP1208_TS112917_0000
tigertail_DBCP1208_TS192917_0000
tigertail_DBCP1208_TS262917_0000
```

13 record(s) selected.

Comparing the two outputs, the user sees that the text search collection tigertail_DBCP1208_TS867530_0000 doesn't have a corresponding text search index. The user then uses the Administration Tool to delete that orphaned collection:

```
adminTool.sh delete -configPath $HOME/sql1lib/db2tss/config
-collectionName tigertail_DBCP1208_TS867530_0000
```

Disabling DB2 Text Search for rich text document support

Rich text support can be disabled at any time on DB2 Text Search servers.

Before you begin

In Version 9.7, the instance owner must hold DBADM with DATAACCESS authority. To disable rich text document support for DB2 Text Search servers you must, as the instance owner, run the **richtextTool** utility with the **disable** option.

About this task

Restrictions

In order to run **richtextTool disable**, you must be logged on as the instance owner.

Procedure

1. Log on as the instance owner.
2. Stop the DB2 Text Search instance service. To stop the service, run **db2ts "STOP FOR TEXT"**. For more information about this command, see “Stopping the DB2 Text Search instance service” on page 56.
3. Run the **richtextTool** utility to disable support. To disable support:

- For Linux and UNIX operating systems:

```
$INSTHOME/sql1lib/db2tss/bin/richtextTool disable <DB2DIR>
```

where *INSTHOME* is the instance home directory and <*DB2DIR*> is the location of the DB2 Version 9.7 copy.

- For Windows operating systems:

```
DB2PATH\db2tss\bin\richtextTool.bat disable
```

```
DB2PATH\db2tss\bin\richtextTool.bat disable DB2PATH
```

where *DB2PATH* is the location where you installed your DB2 Version 9.7 copy.

4. Start the DB2 Text Search instance service. To start the service, run **db2ts "START FOR TEXT"**. For more information about this command, see “Starting the DB2 Text Search instance service” on page 49.

Results

You have disabled rich text support for a DB2 Text Search server.

Text search index creation

A text search index is a compilation of significant terms extracted from text documents. Each term is associated with the document from which it was extracted.

You create a text search index once for each column that contains text to be searched. When you create a text search index, you also create the following objects:

A staging table

This keeps track of all changed rows in the user table.

An event table

This collects information about the status of an update index command or any errors encountered during its processing. If errors occur during indexing, *index update events* are added to the event table.

Triggers on the user table

These add information to the staging table whenever a document in the column is added, deleted, or changed. The information is necessary for index synchronization when indexing time next occurs.

Note: If you use the **LOAD** command to populate your documents, triggers are not activated, and incremental indexing of the loaded documents will not work. Instead, use the **IMPORT** command, which does activate triggers.

After you create a text search index, it is empty (and, therefore, not searchable) until you update it. When creating the text search index, you can specify that the text search instance service is to check periodically whether an update of the text search index is required and that the update command is to be run if necessary.

Creating a text search index

After you enable a database for DB2 Text Search, you can create text search indexes on columns that contain the text that you want to search.

Before you begin

Creating a text search index requires one of following authorization levels:

- CONTROL privilege on the index table
- INDEX privilege on the index table with either the IMPLICIT_SCHEMA authority on the database or the CREATEIN privilege on the index table schema
- DBADM with DATAACCESS authority

Restrictions

- A primary key must exist for this table. If a primary key does not exist, you must create one before creating the index.
- A text column in an index must be one of the following supported types:
 - CHAR
 - VARCHAR
 - LONG VARCHAR
 - CLOB
 - GRAPHIC
 - VARGRAPHIC
 - LONG VARGRAPHIC
 - DBCLOB
 - BLOB
 - XML
- Text search related objects must follow not only DB2 naming conventions, their identifiers must also contain these characters only:
 - [A-Za-z][A-Za-z0-9@#\$_]* or

– "[A-Za-z][A-Za-z0-9@#\$_]*"

This limitation applies to the following scenarios:

- the name of the schema containing the text search index
 - the name of the table the text search index is associated with
 - the name of the text column
 - the name of the text search index
- For composite primary key, the sequence of columns must be in the same sequence as it appears in the base table.

About this task

Procedure

Create a text search index using one of the following methods:

- Issue the **CREATE INDEX** command:

```
db2ts "CREATE INDEX index-name FOR TEXT ON table-name (column-name)"
```
- Call the SYSPROC.SYSTS_CREATE stored procedure:

```
CALL SYSPROC.SYSTS_CREATE('index-schema', 'index-name', 'table-name'  
(column-name), 'options', 'locale', ?)
```

Example

For example, the PRODUCT table in the SAMPLE database includes columns for the product ID, name, price, description, and so on. To create a text search index called MYSCHEMA.MYTEXTINDEX for the NAME column, issue the command or call the stored procedure, as follows:

```
db2ts "CREATE INDEX MYSCHEMA.MYTEXTINDEX FOR TEXT ON PRODUCT(NAME)"  
CALL SYSPROC.SYSTS_CREATE('myschema', 'myTextIndex',  
'product (name)', '', 'en_US', ?)
```

Similarly, to create a text search index called MYSCHEMA.MYXMLINDEX for the XML column DESCRIPTION, enter the following command:

```
db2ts "CREATE INDEX MYSCHEMA.MYXMLINDEX FOR TEXT ON PRODUCT(DESCRIPTION)"
```

or

```
CALL SYSPROC.SYSTS_CREATE('myschema', 'myXMLIndex',  
'product (description)', '', 'en_US', ?)
```

Specifying update frequency

If you don't want to manually manage updating a text search index as the table with which it is associated with changes, you can add update parameters to your **db2ts CREATE INDEX** command. **UPDATE MINIMUM** specifies the minimum number of changes that must be made to the underlying table before an update can be triggered, and **UPDATE FREQUENCY** specifies how often the update services should check to see whether an update is required. For example, to specify that MYSCHEMA.MYTEXTINDEX is to be updated after at least five changes have occurred and that the update services are to check every Monday and Wednesday at 12 midnight and 12 noon, issue the following command:

```
db2ts "CREATE INDEX MYSCHEMA.MYTEXTINDEX FOR TEXT ON PRODUCT(NAME)  
UPDATE FREQUENCY d(1,3) h(0,12) m(0) UPDATE MINIMUM 5"  
CALL SYSPROC.SYSTS_CREATE('myschema', 'myTextIndex', 'product (name)',  
'UPDATE FREQUENCY D(1,3) H(0,12) M(0)' 'UPDATE MINIMUM 5', 'en_US', ?)
```

Language specification

When you create an index, you can specify its locale (language and territory), using the **LANGUAGE** option. To have your documents automatically scanned to determine the locale, set the **LANGUAGE** to **AUTO**. If you do not specify **LANGUAGE** is not specified then a default **LANGUAGE** is used. This default is derived using the **DEFAULTVALUE** from **SYSIBMTS.TSDEFAULTS** where **DEFAULTNAME='LANGUAGE'**. (In this case, **DEFAULTVALUE** is set at the time the database is enabled for text search. This value is derived from the database territory if the database territory can be mapped to one of the document locales supported. If the database territory cannot be used to determine a supported document locale, **DEFAULTVALUE** is set to **AUTO**.)

Creating a text search index on binary data types

When creating a text search index, you have the option of specifying a code page for a binary column. Doing so helps the DB2 Text Search engine identify the character encoding.

About this task

To specify the code page when creating the text search index, use the following command:

```
db2ts "CREATE INDEX index-name FOR TEXT ON table-name
CODEPAGE code-page"
```

When you store data in a column having a binary data type, such as **BLOB** or **FOR BIT DATA**, the data is not converted. This means that the documents retain their original code pages, which can cause problems when you create a text search index because you might have two different code pages. Therefore, you need to determine whether you are using the code page of the database or the code page specified for the db2ts **CREATE INDEX** command. If you do not know which code page was used to create the text search index, you can find out by issuing the following statement:

```
db2 "SELECT CODEPAGE FROM SYSIBMTS.TSINDEXES where INDSHEMA='schema-name'
and INDNAME='index-name'"
```

Creating a text search index on unsupported data types

If documents are in a column of an unsupported data type, such as a user-defined type (UDT), you must provide a function that takes the user type as input and provides an output type that is one of the supported types.

About this task

A text column in an index must be one of the following supported types:

- CHAR
- VARCHAR
- LONG VARCHAR
- CLOB
- GRAPHIC
- VARGRAPHIC
- LONG VARGRAPHIC
- DBCLOB
- BLOB
- XML

To convert the data type of the column to one of valid types, use one of the following methods:

- Run the **db2ts CREATE INDEX** command with the name of a transformation function.

```
db2ts "CREATE INDEX index-name FOR TEXT ON
      table-name function-name(text-column-name)"
```
- Use a user-defined external function (UDF), which is specified by *function-name*, that accesses text documents in a column that is not of a supported type for text searching, performs a data-type conversion of that value, and returns the value as one of the supported data types.

Example

In the following example, there is a table UDTTABLE that contains a column of a user-defined type (UDT) named "COMPRESSED_TEXT", which is defined as CLOB(1M). To create an index on that data type, first create a UDF called UNCOMPRESS, which receives a value of type COMPRESSED_TEXT. Next, create your text search index in the following way:

```
db2ts "CREATE INDEX UDTINDEX FOR TEXT ON
      UDTTABLE UNCOMPRESS(text) ..."
```

Improving the performance of text search indexes

There are a number of ways that you can improve the performance of your text search indexes.

About this task

To enhance performance during indexing, use the following techniques:

- To store a text search index and the database files, use a VARCHAR data type.
- For primary key columns, use small data types, such as TIMESTAMP or INTEGER, instead of a VARCHAR type.
- Ensure that your system has enough real memory available for the index update operation. Index updates require memory that is in addition to that required for any database buffer pools. If there is insufficient memory, the operating system uses paging space instead, which decreases search performance considerably.
- Avoid working with a large number of small documents, as it increases your memory usage for each document.

Also, note that performance might decrease during an index update if error and warning messages are written to the event table.

Text search index maintenance

After you create text search indexes, there are several maintenance tasks that you need to perform. There are several ways to perform these tasks, including using various administration commands, stored procedures, and the Administration Tool.

The routine text search index maintenance tasks include the following ones:

- Running periodic updates
 Unless you specified that automatic updates are to be performed, you must update the text search indexes to reflect changes in the indexed text columns that they are associated with.
- Monitoring the event table
 You can use the event table to determine whether there are document errors or whether the index update frequency needs to change.

Less frequent maintenance tasks include altering and dropping text search indexes.

Updating a text search index

You can update a text search index automatically or manually. Automatic updates occur based on how you defined the text search index. You can update indexes manually by issuing a command or by calling a stored procedure.

Before you begin

Updating a text search index requires either the CONTROL privilege or DATAACCESS authority on the target table.

About this task

After creating and updating (filling) the text search index for the first time, you must keep it up to date. For example, when you add a text document to a database or change an existing document in a database, you must index the document to keep the content of the text search index synchronized with the content of the database. Also, when you delete a text document from a database, you must remove its terms from the text search index.

You should plan periodic indexing carefully because indexing text documents is a time- and resource-consuming task. The time taken depends on many factors, including how big the documents are, how many documents you added or changed since the previous text search index update, and how powerful your processor is.

The Administration Tool's status option can be used to retrieve information about the progress of document updates while the **db2ts UPDATE INDEX** command is running. If an index update is still in progress when a new update starts, the new update fails.

Automatic updates

To have text search index updates performed automatically, use one of the following commands:

- **db2ts CREATE INDEX**
- **db2ts ALTER INDEX**

The **UPDATE FREQUENCY** parameter, which has a minimum setting of five minutes, specifies the update frequency. The **UPDATE MINIMUM** parameter specifies the minimum number of text changes that must be queued.

If there are not enough changes in the staging table for the specified day and time, the text search index is not updated.

Manual updates

There are also times when you want to update a text search index immediately: for example, after you create a text search index, when the index is still empty, or after you have added several text documents to a database and want to search.

To fill or synchronize (update) a text search index with the table data, use one of the following methods:

- Issue the **UPDATE INDEX** command:
`db2ts "UPDATE INDEX index-name FOR TEXT"`

- Call the SYSPROC.SYSTS_UPDATE administrative SQL routine:

Example

For example, suppose that there are two text search indexes on the PRODUCT table: MYSCHEMA.MYTEXTINDEX on the NAME column and MYSCHEMA.MYXMLINDEX on the DESCRIPTION column. A new entry is added to PRODUCT as follows:

```
INSERT INTO PRODUCT VALUES ('100-104-01', 'Wheeled Snow Shovel', 99.99, NULL,
NULL, NULL, XMLPARSE(DOCUMENT '<product xmlns="http://posample.org/wheelshovel"
pid="100-104-01"><description><name>Wheeled Snow Shovel</name>
<details>Wheeled Snow Shovel, lever assisted, ergonomic foam grips, gravel wheel,
clears away snow 3 times faster</details><price>99.99</price>
</description></product>'))
```

To make the information in the new entry searchable, issue the following commands:

```
db2ts "UPDATE INDEX MYSCHEMA.MYTEXTINDEX FOR TEXT"
db2ts "UPDATE INDEX MYSCHEMA.MYXMLINDEX FOR TEXT"
```

Clearing text search index events

If you no longer need the messages in the event view of an index, you can clear (delete) them.

Before you begin

Clearing the event view requires either the CONTROL privilege or DBADM with DATAACCESS AUTHORITY.

About this task

Information about indexing events, such as the update start and end times, the number of indexed documents, or document errors that occurred during the update, are stored in the event view of a text search index. This information can help you determine the cause of a problem.

Procedure

To clear the event view of a text search index, use one of the following methods:

- Run the **db2ts CLEAR EVENTS FOR INDEX** command, as follows:

```
db2ts "CLEAR EVENTS FOR INDEX index-name FOR TEXT"
```
- Use the SYSPROC.SYSTS_CLEAR_EVENTS administrative SQL routine, as follows:

```
CALL SYSPROC.SYSTS_CLEAR_EVENTS('index-schema', 'index-name', 'locale')
```

Altering a text search index

You can alter the update properties of a text search index.

Before you begin

The **db2ts ALTER INDEX** command requires either the CONTROL privilege or DBADM with DATAACCESS authority on the target table.

Procedure

To alter an index, use one of the following methods:

- Run the following command:

```
db2ts "ALTER INDEX index-name FOR TEXT update-characteristics"
```

Where *update-characteristics* is a characteristic such as the update frequency of the text search index.

- Call the SYSPROC.SYSTS_ALTER administrative SQL routine:

```
CALL SYSPROC.SYSTS_ALTER('db2ts', 'myTextIndex', 'alter-option',  
  'en_US', ?)
```

Where *alter-option* is a characteristic such as the update frequency of the text search index.

Results

If an index update is already running, an error message is displayed, informing you that the text search index is currently locked and that no changes can be made.

Example

You can use either method to change both the update frequency of a text search index and the minimum number of changes required to trigger an update. (If you do not specify any parameters, the current settings are left unchanged.) For example, to change the update frequency for the text search index MYTEXTINDEX so that it is updated from Monday to Friday at 12 noon and 3 p.m., provided that at least 100 changes have occurred to the indexed column, issue the following command:

```
db2ts "ALTER INDEX MYTEXTINDEX FOR TEXT  
UPDATE FREQUENCY d(1,2,3,4,5) h(12,15) m(00) UPDATE MINIMUM 100"
```

To stop the periodic updating of MYTEXTINDEX, issue the following command:

```
db2ts "ALTER INDEX MYTEXTINDEX FOR TEXT UPDATE FREQUENCY NONE"
```

Viewing text search index status

To get information about the current text search indexes within a database, you can query the administrative views or use the Administration Tool.

About this task

To obtain information about the currently available text search indexes, use the following command:

```
db2 "select column-name from SYSIBMTS.TSINDEXES"
```

To check the status of all text search collections and their properties using the Administration Tool, use the following command:

```
adminTool status  
-configPath absolute-path-to-config-folder
```

Dropping a text search index

When you no longer intend to perform text searches in a text column, you can drop the text search index.

Before you begin

When you issue the **DROP INDEX** command, at least one of the following privileges must be held by the authorization ID of the statement:

- CONTROL privilege on the table on which the index is defined
- DBADM authority

About this task

When you drop a text search index, the following other objects are also dropped:

- Index staging and event tables
- Triggers on the user table

Always drop the text search indexes on a table before dropping the table. If you drop the table first, you may create what is called an *orphaned collection*. When you create a text search index, a collection (the file system representation of the index) is created with an automatically generated name. If the collection remains after the index has been dropped, it can lead to problems with future queries if the following conditions also true:

- the same database connection is being used,
- a table is created with the same table name,
- a text index with the same name as before is created on this table, and
- the same query is reissued as before.

In this case, a cached query plan might be reused, which could result in a wrong query result.

Procedure

To drop a text search index, use one of the following methods:

- Issue the **DROP INDEX** command:

```
db2ts "DROP INDEX index-name FOR TEXT"
```
- Call the SYSPROC.SYSTS_DROP stored procedure:

```
CALL SYSPROC.SYSTS_DROP('index-schema', 'index-name', 'locale', ?)
```

Where *locale* is the five-character locale code, such as en_US, that specifies the language in which messages will be written to the log file.

What to do next

Note: If any orphaned collections exist after you drop a text search index, you can remove them using the Administration Tool.

If, after dropping a text search index, you plan to create a new one on the same text column, you must first disconnect from and then reconnect to the database.

Backing up and restoring text search indexes

About this task

Procedure

- Use the following steps to back up a database with DB2 Text Search indexes:
 1. Get a current list of text index locations for DB2 Text Search indexes

- ```
db2 "select indschema, indname, collectiondirectory, collectionnameprefix
from sysibmts.tsindexes"
```
2. Ensure that no DB2 Text Search administrative command is running
  3. Stop the DB2 Text Search services

```
db2ts stop for text
```
  4. Back up the database by issuing the following command:

```
db2 backup database <db_name>
```
  5. Back up the text search index directories and subdirectories.
  6. Restart DB2 Text Search services
- Use the following steps to restore a database with DB2 Text Search indexes:
    1. Make sure no DB2 Text Search administrative command is running
    2. Stop the DB2 Text Search services

```
db2ts stop for text
```
    3. Restore database by issuing the following command:

```
db2 restore database <db_name>
```
    4. Restore the backup of text search index locations to the same path as before.
    5. Restart DB2 Text Search services

```
db2ts start for text
```

---

## Synonym dictionaries for DB2 Text Search

A synonym dictionary contains words that are synonyms of each other. You can use a synonym dictionary to search for synonyms of your query terms in a text search index, thus improving the results of your search queries.

Using a synonym dictionary, you can search for words specific to your organization, such as acronyms and technical jargon.

By default, a synonym dictionary is not used for a search. To use a synonym dictionary, you must explicitly add it to a specific text search index. The text search index needs to be updated at least once before you can add a synonym dictionary. After the synonym dictionary has been added, you can modify it as frequently as you want.

A synonym dictionary consists of synonym groups that you define in an XML file, as shown in the following example:

```
<?xml version="1.0" encoding="UTF-8"?>
<synonymgroups version="1.0">
 <synonymgroup>
 <synonym>ball</synonym>
 <synonym>globe</synonym>
 <synonym>sphere</synonym>
 <synonym>orb</synonym>
 </synonymgroup>
 <synonymgroup>
 <synonym>worldwide patent tracking system</synonym>
 <synonym>wpts</synonym>
 </synonymgroup>
</synonymgroups>
```

## Adding a synonym dictionary for DB2 Text Search

You can easily add a synonym dictionary to a text search index by using the Synonym Tool.

## Before you begin

- You must activate the DB2 Text Search instance service before you can add a synonym dictionary to a text search index.
- You must have updated the text search index at least once.
- You must also have a synonym XML file that specifies synonym groups.

## Procedure

To add a synonym dictionary:

1. Copy the XML file to any directory on the DB2 Text Search server.
2. Determine the name of the text search collection associated with the text search index to which you want to add the synonym dictionary. You can use the Administration Tool to report all text search collections, as follows:  

```
adminTool status -configPath absolute-path-to-config-folder
```
3. Use the Synonym Tool to add the synonym dictionary to the specific text search index. You can add the synonyms in **append** or **replace** mode, meaning that you either add them to or replace the existing synonyms defined for that text search index.

```
synonymTool importSynonym -synonymFile absolute-path-to-syn-file
-collectionName collection-name -replace true or false
-configPath absolute-path-to-config-folder
```

**Note:** If the XML format is not valid or if the XML file is empty, an error is returned.

## Example

For example, to add the synonym file `synfile.xml` in append mode, use the following command:

```
synonymTool importSynonym
-synonymFile $HOME/sql1lib/misx/xmlsynfile.xml
-collectionName tigertail_DBCP1208_TS867530_0000
-replace false
-configPath $HOME/sql1lib/db2tss/config
```

## Removing a synonym dictionary for DB2 Text Search

You need to remove synonym dictionaries on a collection-by-collection basis, so you must use the Synonym Tool on each text search index.

### About this task

To remove a synonym dictionary, use the following command:

```
synonymTool removeSynonym -collectionName collection-name
-configPath absolute-path-to-config-folder
```

Where *collection-name* specifies the text search collection and *absolute-path-to-config-folder* specifies the absolute path to the text search configuration folder.



---

## Chapter 6. Searching with text search indexes

After you populate a text search index with data, you can search that index. DB2 Text Search supports searches in SQL, XQuery, and SQL/XML.

You can use the following search functions:

- The SQL function CONTAINS and the XML function xmlcolumn-contains, to create queries for specific words or phrases
- The SQL function SCORE, to obtain the relevancy of a found text document

Searches on text search indexes can range from the simple, such as queries for the occurrence of a single word in a title, to the complex, such as queries that use Boolean operators or term boosting. In addition to the operators that allow you to refine the complexity of your search, features such as synonym dictionaries and linguistic support can enhance searches on text search indexes.

---

### Search functions for DB2 Text Search

After you update a text search index, you can search using the CONTAINS or SCORE SQL scalar search function or using the xmlcolumn-contains function.

Searches on text search indexes can range from the simple, such as queries for the occurrence of a single word in a title, to the complex, such as queries that use Boolean operators or term boosting. In addition to the operators that allow you to refine the complexity of your search, features such as synonym dictionaries and linguistic support can enhance searches on text search indexes.

You can use the following search functions:

- The SQL function CONTAINS and the XML function xmlcolumn-contains, to create queries for specific words or phrases
- The SQL function SCORE, to obtain the relevancy of a found text document

The scalar text search functions, CONTAINS and SCORE, are seamlessly integrated within SQL. You can use the search functions in the same places that you would use standard SQL expressions within SQL queries. The SQL SCORE scalar function returns an indicator of how well the text documents matched a given text search condition. The SELECT phrase of the SQL query determines which information is returned to you.

The CONTAINS function searches for matches of a word or phrase and can be used with wildcard characters to search for substring matches in a manner similar to the SQL LIKE predicate and can search for exact string matches in a manner similar to the SQL = operator. However, there are key distinctions between using the CONTAINS function and using the SQL LIKE predicate or the = operator. The LIKE predicate and the = operator search for patterns in a document, while CONTAINS uses linguistic processing: that is, it searches for different forms of the search term. For example, even without using wildcard characters, searches for the term work also return documents containing working and worked. Moreover, you can add a synonym dictionary to the text search index, increasing the scope of a search. For example, you can group laptop and ThinkPad together so they are

returned from searches for notebook computers. For XML documents, the XML search argument syntax allows you to search for text inside tags and attributes. As well, XQuery searches are case sensitive.

Note that the DB2 optimizer estimates how many text documents can be expected to match a CONTAINS predicate and how costly different access plan alternatives will be. The optimizer chooses the cheapest access plan.

The function `xmlcolumn-contains` is a built-in DB2 function that returns XML documents from a DB2 XML data column based on a text search performed by the DB2 Text Search engine. You can use `xmlcolumn-contains` in XQuery expressions to retrieve documents based on a search of specific document elements. For example, if your XML documents contain product descriptions and prices for toys that you sell, you can use `xmlcolumn-contains` in an XQuery expression to search the description and price elements and return only the documents that have the term `outdoors` but not `pool` and cost less than \$25.00.

There are key distinctions between using the `xmlcolumn-contains` function and the XQuery `contains` function. The XQuery `contains` function searches for a substring inside a string; it looks for an exact match of the search term or phrase. The XQuery `xmlcolumn-contains` function, however, has similar functionality to the CONTAINS function, except that it operates on XML columns only. As well, it returns XML documents containing the search term or phrase, whereas `contains` returns only a value such as 1, 0, or NULL to indicate whether the search term was found.

#### Locale specification

When you perform a search on a text search index, it is recommended that you always use the **QUERYLANGUAGE** option with your search query to specify which locale (a combination of language and territory information) to use to interpret a search term. For example, if you have a search term such as "bald", you can specify to treat it as an English word by setting the **QUERYLANGUAGE=en\_US** in the search query. Similarly, if you want it to be treated as a German word, **QUERYLANGUAGE** can be set to `de_DE`. However, it should be noted that the results returned are highly dependent on the LANGUAGE used for indexing, regardless of the **QUERYLANGUAGE** specified in a query.

If the **QUERYLANGUAGE** is not specified in the search query, then the following logic is used:

- The search term is interpreted to be of the locale that was set for the underlying text index during index creation.
- If the locale set for the index during index creation is AUTO, then this defaults to English (`en_US`), and the search term will be treated as an English word.

Restrictions:

- If the locale specified in the search queries is invalid (for example, **QUERYLANGUAGE=Mongolian**), then the query will be considered invalid and an exception will be thrown.
- Setting **QUERYLANGUAGE=AUTO** in the search query is an unsupported option and the results of the query are undefined.

Note that the locale specified by **QUERYLANGUAGE** has no effect on the locale of error messages resulting from search queries. The error-message locale that is used depends on whether you started the text search instance services. If you did not

start them, messages are written using en\_US; if you did start them, messages are written in the same locale of the environment in which you issued the **db2ts START FOR TEXT** command.

### **Limitation on applying text search on top of LEFT OUTER JOIN and FULL OUTER JOIN**

Text search predicates use a join on the primary key to integrate results from the text search index with the base table. The CONTAINS and SCORE functions cannot be used with a LEFT OUTER JOIN or FULL OUTER JOIN because primary keys may not be available for integration during the execution of a query plan. Text Search indexes are collocated with base tables in MPP environments. Ensure that rows from a text-indexed table have not been moved to other nodes when applying these functions.

1. The primary key may have been projected out, therefore the join may not be possible.
2. More importantly, in MPP environment, text index is collocated with base table. For search function to return correct result, ensure the rows from a text-indexed table have not been moved to other nodes when applying those functions. The most reliable approach is to require the text search function occurs right next base table access.

---

## **DB2 Text Search query planning**

There are a number of things to consider when planning your text search query.

### **DB2 Text Search arguments**

Usage of wildcard characters and their expansion limit, the case sensitivity of arguments or the properties of argument options impact query behavior and often performance.

#### **Wildcard character**

Using a wildcard character at the beginning of a search term slows down query processing. Where possible, avoid performing searches such as *\*search-term* or *?search-term*.

**Note:** If the character ? is used and the term is not enclosed in double quotation marks, the term is interpreted as optional. An optional marker does not impact performance.

#### **Wildcard expansion limit**

When a query term includes a wildcard, the query term is expanded to retrieve matching documents. If a text index collection includes more distinct matching terms than the defined wildcard expansion limit, only the subset of documents that match the already expanded terms will be returned. This limitation is mostly relevant for the wildcard character \*. Include more characters into the search term to reduce the number of potentially matching distinct terms.

By default, 1024 terms can be returned. To change this limit, specify the **queryExpansionLimit** parameter and a value for the parameter in the `<ECMTS_HOME>\config\config.xml` file. For example, to set the limit to 4096, add the following line to the file:

```
<queryExpansionLimit>4096</queryExpansionLimit>
```

## Case sensitivity

Text search arguments are not case sensitive. This is true even when you specify an exact term or phrase using double quotation marks. For example, a search for the term "Hamlet" can return both the Shakespeare play Hamlet and hamlet, the word for a small village.

## Search argument options

Search argument options are properties of the search argument. For example, in the following search query for the word bank in English and German, the search arguments in the two calls are distinct because the search arguments belong to different query languages in these two calls:

```
...CONTAINS(column, 'bank', 'QUERYLANGUAGE=en_US')
and CONTAINS(column, 'bank', 'QUERYLANGUAGE=de_DE')...
```

## DB2 Text Search multiple predicates

If a query contains multiple predicates, consider the following limitations depending on how the predicates are organized.

### UNION versus OR operators

Query performance might improve by using UNION instead of OR to combine multiple predicates.

### Limitation of using a JOIN

The CONTAINS and SCORE functions are not supported as a predicate in a LEFT OUTER JOIN or FULL OUTER JOIN. Using the search functions on the non-tuple preserving side of an outer join results in an error.

## DB2 Text Search SCORE function

The score of a document is dynamic.

Updates to a document as well as adding or removing documents from a text index can cause a change of the score of a document for a query term.

Assume there is a set of documents discussing transportation and pollution. If you want to locate documents containing references to both terms, while treating occurrences of the term pollution more prominently than transportation, you can use the following command:

```
SELECT document_id
FROM document_library
WHERE SCORE(document_content, 'pollution') >
SCORE(document_content, 'transportation')
and CONTAINS(document_content, 'transportation pollution') = 1
```

To enhance performance, you can format your query to use the boost (^) modifier so that the search function is run only once, as follows:

```
SELECT document_id
FROM document_library
WHERE SCORE(document_content, 'pollution^10 transportation') > 0
```

The first query does not return any results if pollution scores low. The second query gives higher importance to pollution but still returns documents if pollution scores low in all documents.

## DB2 Text Search RESULTLIMIT function

Multiple instances of **RESULTLIMIT** within a query require the same search argument to produce predictable results.

### Description

If you use multiple text searches that specify **RESULTLIMIT** in the same query, use the same search argument. Using different text search arguments might not return the expected results.

For example, in the following query, it is unpredictable whether the 10 documents specified by **RESULTLIMIT** will be returned:

```
SELECT EMPNO
FROM EMP_RESUME WHERE RESUME_FORMAT = 'ascii'
AND CONTAINS(RESUME, '"ruby on rails"', 'RESULTLIMIT=10') = 1
AND CONTAINS(RESUME, '"java script"', 'RESULTLIMIT=10') = 1
```

Instead, use **RESULTLIMIT** as follows:

```
SELECT EMPNO
FROM EMP_RESUME WHERE RESUME_FORMAT = 'ascii'
AND CONTAINS(RESUME, '"java script" "ruby on rails"', 'RESULTLIMIT=10') = 1
```

Note that this method works only when both **CONTAINS** functions are operating on the same table column. If they are not operating on the same column, try using **FETCH FIRST *n* ROWS** to improve query performance.

---

## Searching in text search indexes

You can use an SQL statement or XQuery to search through text search indexes.

### Procedure

To search a text search index for a specific term or phrase, use one of the following methods:

- Search with SQL.

To search a text search index for a specific term or phrase with an SQL statement, use the **CONTAINS** function as follows:

```
db2 "SELECT column-name FROM table-name WHERE CONTAINS (...)"
```

For example, the following query searches the **PRODUCT** table for the names and prices of various snow shovels:

```
db2 "SELECT NAME, PRICE FROM PRODUCT WHERE CONTAINS
(NAME, '"snow shovel"') = 1"
```

- Search with XQuery.

To search a text search index for a specific term or phrase using XQuery, use the **db2-fn:xmlcolumn-contains()** function.

For example, the following query searches the **PRODUCT** table for the names and prices of various snow shovels:

```
db2 "xquery for \${info} in db2-fn:xmlcolumn-contains
('PRODUCT.DESCRPTION', '"snow shovel"')
return <result> {\${info}/description/name, \${info}/description/price} </result>"
```

**Note:** Depending on the operating system shell that you are using, you might need a different escape character in front of the dollar sign of the variable information. The previous example uses the backward slash ( \ ) as an escape character for UNIX operating systems.

---

## Searching text search indexes using SCORE

You can use the SCORE function to find out the extent to which a document matches a search argument.

### About this task

SCORE returns a double-precision floating-point number between 0 and 1 that indicates how well a document meets the search criteria. The better a document matches the query, the more relevant the score and the larger the result value.

The score is calculated dynamically based on the content of a text index collection at the time of the query and is therefore only meaningful for text search index which is not partitioned.

### Example

For example, assume there is a set of documents discussing transportation and pollution. If you want to locate documents containing references to both terms, while treating occurrences of the term pollution more prominently than transportation, you can use the following command:

```
SELECT document_id
FROM document_library
WHERE SCORE(document_content, 'pollution') >
SCORE(document_content, 'transportation')
and CONTAINS(document_content, 'transportation pollution') = 1
```

However, to enhance performance, you can format your query to use the boost (^) modifier so that the search function is run only once, as follows:

```
SELECT document_id
FROM document_library
WHERE SCORE(document_content, 'pollution^10 transportation') > 0
ORDER BY SCORE(document_content, 'pollution^10 transportation') DESC
```

The first query does not return any results if pollution scores low. The second query gives higher importance to pollution but still returns documents if pollution scores low in all documents.

Similarly, to search in the SAMPLE database for the number of employees who indicated on their resumes that they know how to program in Java or COBOL, you can issue the following query:

```
SELECT EMPNO, INTEGER(SCORE(RESUME, 'programmer AND (java OR cobol)') * 100)
AS RELEVANCE FROM EMP_RESUME WHERE RESUME_FORMAT = 'ascii'
ORDER BY RELEVANCE DESC
```

However, the following query using CONTAINS is superior. The DB2 optimizer evaluates the CONTAINS predicate in the WHERE clause first and thereby avoids evaluating the SCORE function in the SELECT list for every row of the table. Note that this is possible only if the SCORE and CONTAINS arguments in the query are identical.

```
SELECT EMPNO, INTEGER(SCORE(RESUME, 'programmer AND (java OR cobol)') * 100)
AS RELEVANCE FROM EMP_RESUME WHERE RESUME_FORMAT = 'ascii'
AND CONTAINS(RESUME, 'programmer AND (java OR cobol)') = 1
ORDER BY RELEVANCE DESC
```

---

## Enhancing performance for full-text queries

To enhance performance during searches, use one or more of the following approaches:

### Procedure

- Avoid using the SCORE function without the CONTAINS function. Also, to avoid duplicate processing, ensure that the string (that is, the search argument and any search options) that you specify for the CONTAINS function exactly matches the string (including white spaces) that you use for the SCORE function.
- Ensure that the DB2 compiler has the correct table statistics. Use the **RUNSTATS** command to update the statistics.

---

## Text search argument syntax

A search argument comprises one or more terms and optional search parameters, separated by white space, that you specify to search in text documents.

To perform a simple search, you can enter one or more terms. The search engine returns documents that contain all of those terms and, by default, variations on those terms. For example, if you perform a search on the term king, documents containing king and kings are returned. If you search on two terms, the search engine returns only documents containing both terms. If you want the terms to be searched as an exact phrase, add quotation marks.

### Note:

- Searches are not case sensitive, so a search in Spanish for the exact term "DOS" might return documents containing DOS or dos.
- Text search queries must not exceed DB2 SQL query limits.

The more specific the search term that you use, the more precise the results. However, you can also refine your searches by using options such as the following ones:

- Boolean operators. Use AND or OR to search on both terms or at least one term.
  - The AND operator is the default conjunction operator, so if there is no logical operator between the two terms, AND is used.
  - The OR operator links the two terms and finds a matching document if either of the terms exists in a document.
- Occurrence modifiers. Use +, NOT, or - to specify that terms are required or prohibited.
  - The + modifier is distinct from the AND operator because the + modifier indicates that the second term must be an exact match—that is, no synonym is used.
  -
- Boost. Use ^ to give higher importance to occurrences of a given term (relevant for the SCORE function).

- ^ provides a *boost* to the term or phrase that precedes it when the specified number is greater than 1. If you want to reduce the ranking of the term or phrase in the returned list, specify a number greater than 0 but less than 1.
- For the boost modifier to work correctly, use it with the ORDER BY clause and the SCORE function.
- Wildcards. Use ? to enable one character or \* to enable any number of characters to be added to your search terms, which allows spelling variations and increases search scope.
  - The \* character specifies that any number of characters can be added to the term.
  - Using the wildcard \* at the beginning of a search term has a negative performance impact on the search query.
  - The ? character specifies that a single character can be added to the term.
  - You must use double quotation marks for an initial ? character to be treated as a wildcard character. Otherwise, it is treated as an *optional-term* modifier, as outlined in the following bullet.
- Optional terms. Use an initial ? to specify that a term or phrase is optional
- The escape character: Use \ so that special characters, namely:

|   |   |    |   |   |
|---|---|----|---|---|
| < | > | && |   | ! |
| ( | ) | !  | = | " |
| { | } | ~  | * | ? |
| [ | ] | :  | \ | - |

are analyzed as part of the query.

- Exact search. Use " " around your search term or phrase to have only exact matches returned.
- Parentheses. Use ( ) to have search terms (and the relationship) treated as a single item.

Write XML search queries that are sent to the XML parser in a subset of the XPath language using opaque terms. The query parser recognizes an opaque term by the syntax that is used in the query.

For any language-specific processing during a search, a locale is assumed for the search-argument parameter. This query language is the locale of the text search index that is used when the search function is run.

The search argument syntax is as follows:

*Search argument*

**QualifiedClause** ((Operator) (QualifiedClause))

*Operator*

AND | OR

*QualifiedClause*

(Modifier) **Clause** (^number)

*Modifier*

+ | - | NOT

*Clause* unqualified term | opaque term.

**Note:**

- An unqualified term is a term or a phrase. A term can be a word, such as king; an exact word, such as "king"; or a word that includes a wildcard, such as king\* or king?. Similarly, a phrase can be a group of words, such as cabbages and kings; an exact phrase, such as "The King and I"; or a phrase that includes a wildcard, such as "all the king's ho\*ses" or "all the king's ?".
- An opaque query term is not parsed by the linguistic query parser; opaque terms are identified by their syntax. The opaque term used for text search queries is @xpath: for example, @xpath: '/TagA/TagB[. contains("king")]'

## Examples

Table 2. Boolean operators for search queries

| Operator | Example                    | Query results                                                                                                                                 |
|----------|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| AND      | King AND Lear<br>King Lear | Returns documents that contain the terms King and Lear. If you have a synonym dictionary enabled, words such as monarch can also be returned. |
| OR       | Hamlet OR Othello          | Returns documents that contain either Hamlet or Othello.                                                                                      |

Table 3. Occurrence modifiers

| Modifier | Example                               | Query result                                                                                                   |
|----------|---------------------------------------|----------------------------------------------------------------------------------------------------------------|
| NOT<br>- | Hamlet NOT Othello<br>Hamlet -Othello | Returns documents that contain Hamlet but not Othello.                                                         |
| +        | Lear + King                           | Returns documents that contain the terms Lear and King. Documents containing Lear and monarch aren't returned. |

Table 4. Other modifiers

| Modifier                                                                         | Example                               | Query result                                                                                                                                                                                                                                                                 |
|----------------------------------------------------------------------------------|---------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>term1</i> or <i>phrase1</i> ^ <i>number</i><br><i>term2</i> or <i>phrase2</i> | Hamlet^2 Othello<br>Hamlet Othello^.5 | Returns documents containing Hamlet and Othello but gives more importance to the term Hamlet. In both example queries, each occurrence of the term Hamlet is given twice as much importance as each occurrence of Othello is given.                                          |
| *                                                                                | king*<br>k*ng<br>*ing                 | Returns documents that contain possible combinations of the search term with the wildcard character. The example query could return results such as king and kingdom in the first example, king and kissing in the second example, and king and skiing in the third example. |

Table 4. Other modifiers (continued)

| Modifier                                           | Example                                                       | Query result                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------------------------------------------|---------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ?                                                  | mea?<br>be?n<br>"?ean"                                        | Returns documents that contain possible combinations of the search term with the wildcard character. returning results such as meal and mean in the first example, bean and been in the second example, and mean and bean in the third example.                                                                                                                                                     |
| initial ?                                          | King James ?Edition                                           | Returns documents that contain both king and james, but edition is an optional term.                                                                                                                                                                                                                                                                                                                |
| "phrase"<br>"exact term"<br>"phrase with wildcard" | "King Lear"<br>"king"<br>"John * Kennedy"<br>"John ? Kennedy" | Returns documents that contain the exact word or phrase: King Lear in the first example and the word king and no other forms, such as kings or kingly, in the second example.<br><br>You can use quotation marks with wildcards, returning occurrences of John Kennedy with or without various middle names or initials in the third example and John <i>initial</i> Kennedy in the fourth example. |
| ( )                                                | (Hamlet OR Othello) AND plays                                 | Returns documents that contain the following terms: <ul style="list-style-type: none"> <li>• The term Hamlet or Othello</li> <li>• The term plays</li> </ul>                                                                                                                                                                                                                                        |
| \                                                  | \(1\+1\)\:2                                                   | Returns documents that contain (1+1):2. Use the \ character to escape special characters that are part of the query syntax.                                                                                                                                                                                                                                                                         |

## Searching XML documents using DB2 Text Search

DB2 Text Search supports using XML search for searching XML documents.

By using a subset of the XPath language with extensions for text search, XML search allows you to index and search XML documents. You can use structural elements (tag names, attribute names, and attribute values) separately or combine them with free text in queries.

The following list highlights the key features of XML search:

### XML structural search

By using XML search syntax in text search queries, you can search XML documents for structural elements (tag names, attribute names, and attribute values) and text scoped by those elements. Note that plain searches do not search the attribute field in an XML document.

### XML query tokenization

The text used in the XML search predicate expression as XML query terms is tokenized the same way that text in non-XML query terms is tokenized, except that spelling corrections, fielded terms, and nested XML search terms are unsupported. Synonyms, wildcard characters, phrases, and lemmatization are supported.

### Disregarding of XML namespaces

Namespace prefixes are not retained in the indexing of XML tag and attribute names. You can index and search XML documents declaring and using namespaces, but namespace prefixes are discarded during indexing and removed from XML search queries.

### Numeric values

Predicates comparing attribute values to numbers are supported.

### Complete match

The operator = (equal sign) with a string argument in a predicate means that a complete match of all tokens in the string with all tokens in the identified text span is required, with the order being significant.

### No UIMA access

Unstructured Information Management Architecture (UIMA) is used for tokenization in XML search, but user-written annotators are unsupported.

The subset of XPath that is implemented in XML search differs from standard XPath in the following ways:

- It does not support iteration and ranges in path expressions.
- It eliminates filter expressions: that is, it allows filtering only in the predicate expression, not in the path expression.
- It disallows absolute path names in predicate expressions.
- It implements only one axis (tag) and allows propagation only in the forward direction.

The following table lists some valid XML search queries.

Table 5. Valid XML search queries

| Query                                                          | Description                                                                                                                       |
|----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| /                                                              | The root node; any document                                                                                                       |
| /sentences                                                     | Any document with a top-level tag of sentences                                                                                    |
| //sentences                                                    | Any document with a tag at any level of sentences                                                                                 |
| sentences                                                      | Any document with a tag at any level of sentences                                                                                 |
| /sentence/paragraph                                            | Any document with a top-level tag of sentences having a direct child tag of paragraph                                             |
| /sentence/paragraph/                                           | Any document with a top-level tag of sentences having a direct child tag of paragraph                                             |
| /book/@author                                                  | Any document with a top-level book tag having an attribute author                                                                 |
| /book//@author                                                 | Any document with a top-level book tag having a descendant tag at any level with attribute author                                 |
| /book[@author contains("barnes") and @title contains("lemon")] | Any document with a top-level book tag with the attributes author and title with values that contain the normalized strings shown |

Table 5. Valid XML search queries (continued)

| Query                                                                                           | Description                                                                                                                                                                         |
|-------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /book[@author contains("barnes") and (@title contains("lemon") or @title contains("flaubert"))] | Any document with a top-level book tag with the specified author attribute and either of the two specified title attributes                                                         |
| /program[. contains("hello, world.")]                                                           | Any document with a top-level program tag containing at least the tokens hello and world                                                                                            |
| /book[paragraph contains("flaubert")]/sentence                                                  | Any document with a top-level tag book tag with a direct child tag of paragraph containing "flaubert" and, referring to the book tag, having a descendant tag sentence at any level |
| /auto[@price <30000]                                                                            | Any document with a top-level auto tag having an attribute price with a numeric value that is less than 30000                                                                       |
| //microbe[@size <3.0e-06]                                                                       | Any document containing a microbe tag at any level with a size attribute with a value that is less than 3.0e-06                                                                     |

**Note:** The following are unsupported in the XML search syntax:

- /\*
- //\*
- /@\*
- //@\*

A *plain* search does not search the attribute field in the XML document.

## XML search configuration for DB2 Text Search

You can configure some of the settings that are used for XML search.

All parser configuration parameters are located in the `parser_config.xml` file, in the XML element defining the parser, `com.ibm.es.nuvo.parser.xml.XMLParser`. Each parameter is specified by a `Parameter` element of this form:

```
<Parameter Name="parameter">setting</Parameter>
```

If no parameter elements are present, the default settings are used. The following list provides the valid parameter names and settings:

### UnresolvableEntityResponse

This element allows DB2 Text Search to handle unresolvable external entities. Even though the DB2 Text Search XML parser does not use a DTD for validation, it still might need one for entity resolution. Use the following settings to specify what the parser should do when it detects a missing DTD or unresolvable entity:

**ignore** The document is accepted for parsing, but any externally defined items that cannot be resolved are silently discarded during parsing, rather than being indexed in unresolved form. This is the default setting.

**reject** The document is rejected with a 404 status code.

### handleExternalFiles

XML documents can refer to other files containing a DTD, schema,

namespace definition, or other content. Such files are typically located on the Web, and downloading them adds to the parsing time for XML documents and the I/O load on the computer running the search system. Most of the information in such external files is metadata specifying the grammatical properties of markup elements and adds nothing to the search index. The parameter **handleExternalFiles** specifies whether to download such files or to treat them as if they are empty, without reading them. You can specify the following settings:

- ignore** External files are not be downloaded; they are resolved as if they were empty. Parsing is faster, but externally defined entities cannot be indexed. This is the default setting.
- read** External files are downloaded if they are accessible, with a consequent increase in parsing time. If a file is inaccessible, an I/O exception is thrown by the parser, and the URI of the affected document is indexed with an error code.

#### **handleSkippedEntities**

XML documents might refer to entities defined elsewhere. If the parser cannot resolve any of these entities, the configuration parameter **handleSkippedEntities** specifies whether to reject the document (not index it) or to ignore the entity and index the rest of the document.

- ignore** Unresolved references are silently ignored during parsing. This is the default setting.
- reject** A document containing an unresolved entity is rejected, and an error entry is stored in the index for its URI.

#### **titleTagNameList**

If an XML document contains a tag named title (case insensitive), the text content of that tag, if any, is indexed as the content of the title field for that document and is field-searchable as such. If an XML document does not contain a title tag, its URL is stored as its field-searchable title. To use a different tag, any of several tags, or no tag to identify document titles for fielded search, you can use the **titleTagNameList** parameter with the following options. In the event of multiple matches, only the content from the first match is indexed as the title of a document.

- absent** If the parameter is not in the configuration file, the only tag used to identify title content is <title>. This is the default setting.
- empty** If the parameter is in the configuration file and does not specify one or more tag names. No tag name is used to identify title content.

*tag-name*

The tag name matching the specified tag name (case insensitive) is used to identify document titles.

*tag-name1, tag-name2, ...*

For each document indexed, the first tag name matching any name in the list (case insensitive) is used to identify the document title.

## **Search syntax for XML documents**

Using an XML search expression, you can use the DB2 Text Search engine to search specific portions of an XML document in a DB2 XML column.

## Syntax

► @xpath: ' XML search query '

### XML search query:

| location-path | [ search-predicate ]

#### @xpath:

The keyword that starts a text search query on an XML document.

#### XML search query

A text search query used by DB2 Text Search to search XML documents. The query is enclosed in single quotation marks. The XML search query is an XML search expression that consists of a location path specifying the portion of the XML document to search and an optional predicate that specifies the search criteria.

#### location-path

An XML search expression that uses a subset of the XPath abbreviated syntax to specify an XML document node or attribute. More information is provided in the "Location path" section.

#### search-predicate

The optional search criteria used by DB2 Text Search when searching an XML document. More information is provided in the "Search predicate" section.

The DB2 Text Search engine returns the XML document if it finds the text specified in the *search-predicate* in the specified nodes or attributes of the XML document.

## Location path

When performing a text search on an XML document, DB2 Text Search uses local node and attribute names and a subset of the XPath syntax to specify nodes and attributes in an XML document. DB2 Text Search supports the following XML search elements:

- Local node or attribute names
- . (period) as the current context node
- / or // as the separator character
- @ as the abbreviated symbol for attribute

**Name normalization:** XML node and attribute names are not normalized when they are indexed for use by the DB2 Text Search engine: they are not converted to lowercase, tokenized, or modified in any way. Case is significant in XML node and attribute names, so the strings that you use for them in queries must match exactly the names appearing in documents to get a match.

**Namespace handling:** When creating a text search index, you can use XML documents that contain XML namespace specifiers, but namespace specifiers are not retained in the index. For example, the tag <nsdoc:heading> is indexed under heading only, and the query term @xpath:'/nsdoc:heading' is parsed as @xpath:'/heading'. XML namespace prefixes are discarded during query parsing.

## Examples

The following example is a valid text search query using XML search that searches for the term snow shovel in the description node of product information:

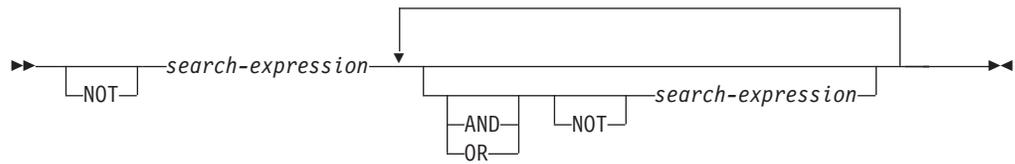
```
@xpath:'/info/product/description[. contains("snow shovel")]'
```

The following example is not a valid text search query using XML search because it uses "..", the XML search abbreviation for parent::node():

```
@xpath:'/info/product/description/..@ID[. contains("A2")]'
```

## Search predicate

### Syntax



### *search-expression*

A DB2 Text Search XML search query. DB2 Text Search uses a search expression to search node or attribute values in an XML document.

You can use the following operators to create search expressions:

- Logical operators: AND, OR, and NOT
- Containment operators: contains and excludes
- Comparison operators: =, >, <, >=, <=, and !=

### Note:

Comparison operators can be applied to attribute values only, not node values.

Thus, for the `<root><aaa id="10">100</aaa><aaa id="11">101</aaa></root>`, the following query is invalid:

```
select id from testtable where contains(item,'@xpath:'/root/aaa[. > 20]''')>0
```

An example of a valid query would be:

```
select id from testtable where contains(item,'@xpath:'/root/aaa/@id[. > 20]''')>0
```

You can combine the comparison and containment operators with the logical operators AND, OR and NOT to create complex search expressions. You can also use parentheses to group expressions.

Use single or double quotation marks to enclose a string. A string that contains quotation marks cannot be enclosed by the same type of quotation marks. For example, a string enclosed in single quotation marks cannot contain a single quotation mark.

**Operator Precedence:** In XML search predicates, comparison operators take precedence over logical operators, and all logical operators have the same precedence. You can use parentheses to ensure intended evaluation precedence.

**Attribute value and node value normalization:** Free text in XML documents (text between tags, not inside a tag itself) and attribute values are normalized before indexing. Free text in XML queries (in containment operators) is normalized in the same way that it is in non-XML queries.

## Example

The following example uses an XML search query to search for products that contain the term snow shovel in the product description and that have a price lower than \$29.99.

```
@xpath:'/info/product [(description contains("snow shovel"))
and (@price < 29.99)]'
```

## Comparison expressions

Comparison expressions compare the value of an attribute with a specified value.

## Syntax

► *path-expression*—*operator*—*literal* ◀

### *path-expression*

The path expression using a subset of the XML search abbreviated syntax to specify a node or attribute.

### *operator*

The type of comparison to perform. The operator can be one of the following types:

- = *path-expression* value is equal to *literal*.
- > *path-expression* value is greater than *literal*.
- < *path-expression* value is less than *literal*.
- >= *path-expression* value is greater than or equal to *literal*.
- <= *path-expression* value is less than or equal to *literal*.
- != *path-expression* value is not equal to *literal*.

### *literal*

A string or number used to compare against the *path-expression* node or attribute value.

Enclose the string in single or double quotation marks. A string that contains quotation marks cannot be enclosed by the same type of quotation marks. For example, a string enclosed in single quotation marks cannot contain a single quotation mark. Use the backslash character (\) to escape double quotation marks (").

If the string contains double quotation marks, you can enclose the string in single quotation marks. The following example shows a string containing double quotation marks enclosed in single quotation marks:

```
'he said "Hello, World"'
```

If the a string contains single quotation marks, you can enclose the string in escaped double quotation marks. The following example shows a string containing a single quotation mark enclosed in double quotation marks:

```
"the cat's toy\"
```

DB2 Text Search features such as phrases, wildcards, and synonyms are not supported in XML search queries.

## Example

The following example uses the = operator to find product IDs equal to the string 100-200-101:

```
@xpath:'/info/product/@pid[. = "100-200-101"]'
```

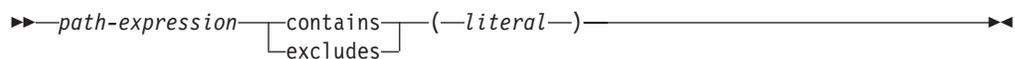
### Note:

The only comparison operators that are supported with string arguments are = and !+, so <, <=, >, >= cannot be used. All six operators are supported with numeric arguments. Numeric arguments are supported for comparison to attribute values, but not to tag(node) content

## Containment expressions

Containment expressions determine whether the value of a node or an attribute contains a specified value.

## Syntax



### *path-expression*

The XML search expression that specifies an XML node or attribute.

### **contains**

An expression that specifies that *path-expression* value contains *literal*.

### **excludes**

An expression that specifies that *path-expression* value excludes *literal*.

### *literal*

A string used to compare against the *path-expression* node or attribute value.

Use single or double quotation marks to enclose a string. A string cannot contain enclosing quote type: for example, a string enclosed in single quotation marks cannot contain a single quotation mark. Use the backslash character (\) to escape double quotation marks (").

If the string contains double quotation marks, you can enclose the string in single quotation marks.

The following example shows a string containing double quotation marks enclosed in single quotation marks:

```
'he said "Hello, World"'
```

If the string contains single quotation marks, you can enclose the string in escaped double quotation marks. The following example shows a string containing a single quotation mark enclosed in double quotation marks:

```
\ "the cat's toy\"
```

DB2 Text Search features such as phrases, wildcards, and synonyms are not supported in XML search queries.

## Example

The following example uses the XQuery abbreviated syntax for path expressions to specify that the description node excludes the term ice scraper:

```
@xpath:'/info/product/description[. excludes('ice scraper')]'
```

---

## Chapter 7. SQL and XML built-in search functions

You can use the following DB2 built-in search functions in DB2 Text Search. The schema of these functions is SYSIBM.

### CONTAINS

Returns a NULL or an INTEGER value of 0 or 1 depending on whether the input text document matches the text search condition

### SCORE

Returns a NULL or a DOUBLE value between 0 and 1 indicating the extent to which the text document meets the search criteria.

### xmlcolumn-contains

Returns a NULL or an INTEGER value 1 or 0 depending on whether the input text document of XML data type matches the text search condition

---

## CONTAINS function

The CONTAINS function searches a text search index using criteria that you specify in a search argument and returns a value that indicates whether a match is found.

### Function syntax

```
▶▶ CONTAINS ((column-name) , (search-argument) [(1)] , (string-constant))
▶▶)
```

### Notes:

1 *string-constant* must conform to the rules for search-argument-options.

### search-argument-options:

```
(1)
|-----|
| QUERYLANGUAGE == locale |
| RESULTLIMIT == value |
| SYNONYM == [OFF | ON] |
```

### Notes:

1 You cannot specify the same clause more than once.

The schema is SYSIBM.

### Function parameters

#### *column-name*

A qualified or unqualified name of a column that has a text search index that is to be searched. The column must exist in the table or view

identified in the FROM clause in the statement and the column of the table, or the column of the underlying base table of the view, must have an associated text search index (SQLSTATE 38H12). The underlying expression of the column of a view must be a simple column reference to the column of an underlying table, either directly or through another, nested view.

*search-argument*

An expression that returns a value that is a string value (except a LOB) that contains the terms to be searched for and is not be all blanks or the empty string (SQLSTATE 42815). The string value that results from the expression should be less than or equal to 4096 bytes (SQLSTATE 42815). The value is converted to Unicode before it is used to search the text search index. The maximum number of terms per query must not exceed 1024 (SQLSTATE 38H10).

*string-constant*

A string constant that specifies the search argument options that are in effect for the function.

The options that you can specify as part of the *search-argument-options* are as follows:

**QUERYLANGUAGE = locale**

Specifies the locale that the DB2 Text Search engine uses when performing a text search on a DB2 text column. The value can be any of the supported locales. If you do not specify **QUERYLANGUAGE**, the default is the locale of the text search index. If the **LANGUAGE** parameter of the text search index is **AUTO**, the default value for **QUERYLANGUAGE** is **en\_US**.

**RESULTLIMIT = value**

If the optimizer chooses a plan that calls the search engine for each row of the result set to obtain the **SCORE**, then the **RESULTLIMIT** option has no effect on performance. However, if the search engine is called once for the entire result set, **RESULTLIMIT** acts like a **FETCH FIRST** clause.

When using multiple text searches that specify **RESULTLIMIT** in the same query, use the same *search-argument*. If you use different *search-argument* values, you might not receive the results that you expect.

**SYNONYM = OFF | ON**

Specifies whether to use a synonym dictionary that is associated with the text search index. The default is **OFF**. To use synonyms, add the synonym dictionary to the text search index using the Synonym Tool.

**OFF** Do not use a synonym dictionary.

**ON** Use the synonym dictionary associated with the text search index.

The result of the function is a large integer. If the second argument can be null, the result can be null; if the second argument is null, the result is null. If the third argument is null, the result is as if you did not specify the third argument. **CONTAINS** returns the integer value 1 if the document contains a match for the criteria specified in the search argument. Otherwise, it returns 0.

CONTAINS is a non-deterministic function.

**Note:** You must take additional steps when using parameter markers as a search argument inside the text search functions. Parameter markers do not have a type when precompiled in JDBC and ODBC programs, but the search argument in the text search functions must resolve to a string value. Because the unknown type of the parameter marker cannot be resolved to a string value (SQLCODE -418), you must explicitly cast the parameter marker to the VARCHAR data type.

## Examples

- The following query is used to find all of the employees who have COBOL in their resumes. The text search argument is not case-sensitive.

```
SELECT EMPNO
FROM EMP_RESUME
WHERE RESUME_FORMAT = 'ascii'
AND CONTAINS(RESUME, 'COBOL') = 1
```

- In the following C program, the exact term ate is searched for in the COMMENT column:

```
char search_arg[100]; /* input host variable */
...
EXEC SQL DECLARE C3 CURSOR FOR
SELECT CUSTKEY
FROM CUSTOMERS
WHERE CONTAINS(COMMENT, :search_arg) = 1
ORDER BY CUSTKEY;
strcpy(search_arg, "ate");
EXEC SQL OPEN C3;
...
```

- The following query is used to find any 10 students who wrote online essays that contain the phrase fossil fuel in Spanish, which is combustible fósil. A synonym dictionary was created for the associated text search index. Because only 10 students are needed, the query is optimized by using the **RESULTLIMIT** option to limit the number of results from the underlying text search server.

```
SELECT FIRSTNAME, LASTNAME
FROM STUDENT ESSAYS
WHERE CONTAINS(TERM_PAPER, 'combustible fósil',
'QUERYLANGUAGE= es_ES RESULTLIMIT = 10 SYNONYM=ON') = 1
```

---

## SCORE function

The SCORE function searches a text search index using criteria that you specify in a search argument and returns a relevance score that measures how well a document satisfies the query as compared with the other documents in the column.

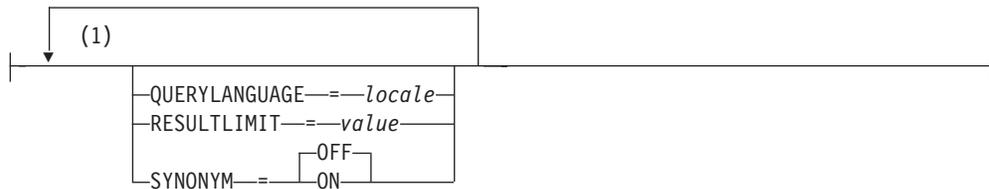
### Function syntax

```
►► SCORE ((column-name, search-argument) (1) , string-constant)
```

#### Notes:

- 1 *string-constant* must conform to the rules for search-argument-options.

## search-argument-options:



### Notes:

- 1 You cannot specify the same clause more than once.

The schema is SYSIBM.

## Function parameters

### *column-name*

A qualified or unqualified name of a column that has a text search index that is to be searched. The column must exist in the table or view identified in the FROM clause in the statement and the column of the table, or the column of the underlying base table of the view, must have an associated text search index (SQLSTATE 38H12). The underlying expression of the column of a view must be a simple column reference to the column of an underlying table, either directly or through another, nested view.

### *search-argument*

An expression that returns a value that is a string value (except a LOB) that contains the terms to be searched for and is not be all blanks or the empty string (SQLSTATE 42815). The string value that results from the expression should be less than or equal to 4096 bytes (SQLSTATE 42815). The value is converted to Unicode before it is used to search the text search index. The maximum number of terms per query must not exceed 1024 (SQLSTATE 38H10).

### *string-constant*

A string constant that specifies the search argument options that are in effect for the function.

The options that you can specify as part of the *search-argument-options* are as follows:

#### **QUERYLANGUAGE = locale**

Specifies the locale that the DB2 Text Search engine uses when performing a text search on a DB2 text column. The value can be any of the supported locales. If you do not specify **QUERYLANGUAGE**, the default is the locale of the text search index. If the **LANGUAGE** parameter of the text search index is **AUTO**, the default value for **QUERYLANGUAGE** is **en\_US**.

#### **RESULTLIMIT = value**

If the optimizer chooses a plan that calls the search engine for each row of the result set to obtain the **SCORE**, then the **RESULTLIMIT** option has no effect on performance. However, if the search engine is called once for the entire result set, **RESULTLIMIT** acts like a **FETCH FIRST** clause.

When using multiple text searches that specify **RESULTLIMIT** in the same query, use the same *search-argument*. If you use different *search-argument* values, you might not receive the results that you expect.

**Note:** If the number of results is an issue, limit the number of results through a refinement of the search terms, rather than by using **RESULTLIMIT**. Because **RESULTLIMIT** returns at most the specified number of results with no consideration of their scores, the highest-ranking documents might not be included.

**SYNONYM = OFF | ON**

Specifies whether to use a synonym dictionary that is associated with the text search index. The default is OFF. To use synonyms, add the synonym dictionary to the text search index using the Synonym Tool.

**OFF** Do not use a synonym dictionary.

**ON** Use the synonym dictionary associated with the text search index.

The result of the function is a double-precision floating-point number. If the second argument can be null, the result can be null; if the second argument is null, the result is null. If the third argument is null, the result is as if you did not specify the third argument.

The result is greater than 0 but less than 1 if the column contains a match for the search criteria specified by the search argument. The more frequently a match is found, the larger the result value. If the column does not contain a match, the result is 0.

SCORE is a non-deterministic function.

**Note:** You must take additional steps when using parameter markers as a search argument inside the text search functions. Parameter markers do not have a type when precompiled in JDBC and ODBC programs, but the search argument in the text search functions must resolve to a string value. Because the unknown type of the parameter marker cannot be resolved to a string value (SQLCODE -418), you must explicitly cast the parameter marker to the VARCHAR data type.

### Example

- The following query is used to generate a list of employees in order of how well their resumes satisfy the query "programmer AND (java OR cobol)", along with a relevance value that is normalized between 0 and 100:

```
SELECT EMPNO,
 INTEGER(SCORE(RESUME,
 'programmer AND (java OR cobol)') * 100) AS RELEVANCE
FROM EMP_RESUME
WHERE RESUME_FORMAT = 'ascii'
 AND CONTAINS(RESUME, 'programmer AND (java OR cobol)') = 1
ORDER BY RELEVANCE DESC
```

---

## xmlcolumn-contains function

The db2-fn:xmlcolumn-contains function returns a sequence of XML documents from an XML data column based on a text search performed by the DB2 Text Search engine for specified search terms.

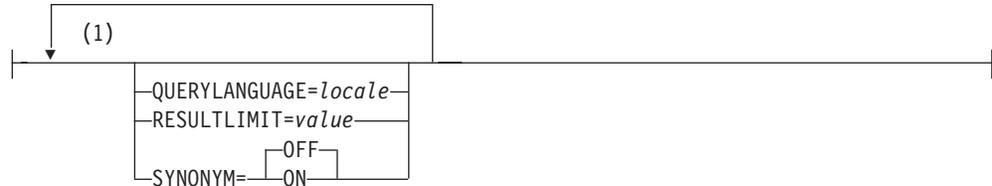
## Syntax

db2-fn:xmlcolumn-contains(*string-literal*,*search-argument* (1) ,*options-string-literal*)

### Notes:

- 1 *options-string-literal* must conform to the rules for *search-argument-options*.

### search-argument-options:



### Notes:

- 1 You can specify each option only once.

### *string-literal*

Specifies the name of a XML data type column to be searched by db2-fn:xmlcolumn-contains. The value of *string-literal* is case sensitive and must match the case of the table and column name. You must qualify the column name using a table name or a view name. The SQL schema name is optional. If you do not specify the SQL schema name, the value of CURRENT SCHEMA is used.

The column must have a text search index.

### *search-argument*

An expression that returns an atomic string value or an empty sequence. The string cannot be all space characters or an empty string. The string must be castable to the type VARCHAR according to the rules of XMLCAST with a maximum length of 4096 bytes.

### *options-string-literal*

Specifies the search argument options that are in effect for the function.

The options that you can specify as part of the *search-argument-options* are as follows:

#### **QUERYLANGUAGE** = *locale*

Specifies the locale that the DB2 Text Search engine uses when performing a text search on a DB2 text column. The value can be any of the supported locales. If you do not specify **QUERYLANGUAGE**, the default is the locale of the text search index. If the **LANGUAGE** parameter of the text search index is AUTO, the default value for **QUERYLANGUAGE** is en\_US.

#### **RESULTLIMIT** = *value*

If the optimizer chooses a plan that calls the search engine for each row of the result set to obtain the SCORE, then the **RESULTLIMIT** option has no effect on performance. However, if the search engine is called once for the entire result set, **RESULTLIMIT** acts like a FETCH FIRST clause.

When using multiple text searches that specify **RESULTLIMIT** in the same query, use the same *search-argument*. If you use different *search-argument* values, you might not receive the results that you expect. For an example of what might happen when using multiple text searches and a solution, see the last example in “Examples.”

**SYNONYM = OFF | ON**

Specifies whether to use a synonym dictionary that is associated with the text search index. The default is **OFF**. To use synonyms, add the synonym dictionary to the text search index using the Synonym Tool.

**OFF** Do not use a synonym dictionary.

**ON** Use the synonym dictionary associated with the text search index.

## Returned values

The returned value is a sequence that is the concatenation of the non-null XML values from the column that is specified by *string-literal*. The non-null XML values are returned in a nondeterministic order. The XML values are the XML documents where the SQL **CONTAINS** function using *search-argument* for the column specified by *string-literal* would return 1. If there are no such XML values, an empty sequence is returned.

If *search-argument* is an empty sequence, an empty sequence is returned. If *search-argument* is an empty string or string containing all space characters, an error is returned. If the third argument is null, the result is as if you did not specify the third argument.

If the column that you specify using *string-literal* does not have a text search index, an error is returned.

The `db2-fn:xmlcolumn-contains` function is related to the `db2-fn:sqlquery` function, and both functions can produce the same result. However, the arguments of the two functions differ in case sensitivity. The first argument, *string-literal*, in the `db2-fn:xmlcolumn-contains` function is processed by XQuery and is case sensitive. Because table names and column names in a DB2 database are uppercase by default, the first argument of `db2-fn:xmlcolumn-contains` is usually uppercase. The first argument of the `db2-fn:sqlquery` function is processed by SQL, which automatically converts identifiers to uppercase.

The following function calls are equivalent and return the same results assuming that the **PRODUCT** table is in the schema currently assigned to **CURRENT SCHEMA**:

```
db2-fn:xmlcolumn-contains("PRODUCT.DESCRPTION", "snow shovel")
```

```
db2-fn:sqlquery("select description from product
where contains(description, 'snow shovel') = 1")
```

## Examples

The following examples use the DB2 Text Search engine to perform searches. The columns being searched are XML columns and have a text search index.

The first function searches for XML documents stored in the **PRODUCT.DESCRPTION** column that contain the words `snow` and `shovel`. The function sets the maximum number of returned documents to two. If the text

search returns a large number of documents, you can optimize the search by using the **RESULTLIMIT** option to limit the maximum number of documents returned.

```
db2-fn:xmlcolumn-contains('PRODUCT.DESCRPTION', 'snow shovel', 'RESULTLIMIT=2')
```

The function returns the XML documents that match the search criteria. The documents might contain more than just a product description. For example, the following XML fragment consists of two product descriptions from an XML column. Each document contains a product description and information such as the product name, price, weight, and product ID.

```
<product xmlns="http://posample.org" pid="100-100-01">
 <description>
 <name>Snow Shovel, Basic 22 inch</name>
 <details>Basic Snow Shovel, 22 inches wide, straight handle with
 D-Grip</details>
 <price>9.99</price>
 <weight>1 kg</weight>
 </description>
</product>
<product xmlns="http://posample.org" pid="100-101-01">
 <description>
 <name>Snow Shovel, Deluxe 24 inch</name>
 <details>A Deluxe Snow Shovel, 24 inches wide, ergonomic curved handle
 with D-Grip</details>
 <price>19.99</price>
 <weight>2 kg</weight>
 </description>
</product>
```

The following function searches the XML column `STUDENT_ESSAYS.ABSTRACTS` for 10 student essays that contain the phrase `fossil fuel` in Spanish, which is `combustible fósil`. The function specifies `es_ES` (Spanish as spoken in Spain) as the language to use for the text search and uses the synonym dictionary that was created for the associated text search index. The function optimizes the search by using **RESULTLIMIT** to limit the number of results.

```
db2-fn:xmlcolumn-contains('STUDENT_ESSAYS.ABSTRACTS', '"combustible fosil"',
 'QUERYLANGUAGE=es_ES RESULTLIMIT=10 SYNONYM=ON')
```

The following example uses `db2-fn:xmlcolumn-contains` to find XML documents stored in the `PRODUCT.DESCRPTION` column that contain the word `ergonomic`. The expression returns the name of the product whose price is less than 20.

```
xquery
declare default element namespace "http://posample.org";
db2-fn:xmlcolumn-contains(
 'PRODUCT.DESCRPTION', 'ergonomic')/product/description[price < 20]/name
```

The previous expression returns only the name elements from the returned XML documents. For example, if the term `ergonomic` is in the product description of the product `Snow Shovel, Deluxe 24 inch`, the expression returns a name element similar to the following one:

```
<name xmlns="http://posample.org" >Snow Shovel, Deluxe 24 inch</name>
```

The following expression uses `db2-fn:xmlcolumn-contains` to find the XML documents from the `PRODUCT.DESCRPTION` column that contain the words `ice` and `scraper`. The expression uses the product IDs from the product descriptions to find purchase orders in the `PURCHASEORDER` table that contain the product IDs. The expression returns the customer IDs from purchase orders that contain the product IDs from the matched XML description documents.

```

xquery
declare default element namespace "http://posample.org";
for $po in db2-fn:sqlquery('
 select XMLElement(Name "po", XMLElement(Name "custid", purchaseorder.custid),
 XMLElement(Name "porder", purchaseorder.porder))
 from purchaseorder')
let $product := db2-fn:xmlcolumn-contains('PRODUCT.DESCRPTION',
 'ice scraper')/product
where $product/@pid = $po/porder/PurchaseOrder/item/partid
order by $po/custid
return $po/custid

```

The expression returns custid elements containing the customer IDs. The elements are in ascending order. For example, if three purchase orders matched and the purchase orders had customer IDs 1001, 1002, and 1003, the expression returns the following elements:

```

<custid xmlns="http://posample.org">1001</custid>
<custid xmlns="http://posample.org">1002</custid>
<custid xmlns="http://posample.org">1003</custid>

```

If there are multiple text searches in the same query, the DB2 Text Search engine combines the multiple text search results and returns them. For example, the following SELECT statement searches for employee resumes that contain the exact phrases ruby on rails and ajax web. The WHERE clause contains two text searches. Each text search returns a maximum of 10 results, and each text search uses a different search argument to search for employee resumes. The statement might return fewer than 10 employee IDs even if there are more than 10 employee resumes that contain both phrases.

```

SELECT EMPNO FROM EMP_RESUME
WHERE XMLEXISTS('db2-fn:xmlcolumn-contains(''EMP_RESUME.XML_FORMAT'',
 ''"ruby on rails"'', ''RESULTLIMIT=10'')')
AND XMLEXISTS('db2-fn:xmlcolumn-contains(''EMP_RESUME.XML_FORMAT'',
 ''"ajax web"'', ''RESULTLIMIT=10'')')

```

For the previous statement, DB2 Text Search returns at most 10 rows for each text search. However, if the resumes in the returned rows contain only one of the phrases (not both phrases), no employee IDs are returned.

One way to modify the SELECT statement is to combine the two text searches in the WHERE clause into a single text search. The following statement uses a single text search and returns employee IDs whose resumes have both the phrase ruby on rails and ajax web:

```

SELECT EMPNO FROM EMP_RESUME
WHERE XMLEXISTS('db2-fn:xmlcolumn-contains(''EMP_RESUME.XML_FORMAT'',
 ''"ruby on rails" AND "ajax web"'', ''RESULTLIMIT=10'')')

```



---

## Chapter 8. Administration commands for DB2 Text Search

There are a number of commands that allow you to administer DB2 Text Search at the instance, database, table, and text-index levels. You run all of the commands using `db2ts`.

Use the instance-level administration commands to start and stop the DB2 Text Search instance services and clean up text search indexes that are no longer usable:

**db2ts START FOR TEXT**

Starts the DB2 Text Search instance services

**db2ts STOP FOR TEXT**

Stops the DB2 Text Search instance services

**db2ts CLEANUP FOR TEXT**

Cleans up any text search collections that are not usable

Use the database-level administration commands to set up or disable databases for DB2 Text Search and clear command locks:

**db2ts ENABLE DATABASE FOR TEXT**

Enables the current database to create, manage, and use text search indexes

**db2ts DISABLE DATABASE FOR TEXT**

Disables DB2 Text Search for a database and drops a number of text search catalog tables and views

**db2ts CLEAR COMMAND LOCKS**

Deletes all command locks for a specific text search index or for all indexes in a database

Use table- and index-level commands to create and manipulate text search indexes on columns of a table:

**db2ts CREATE INDEX**

Creates a text search index

**db2ts DROP INDEX**

Drops a text search index associated with a text column

**db2ts ALTER INDEX**

Changes the characteristics of a text search index

**db2ts UPDATE INDEX**

Populates or updates a text search index based on the current contents of a text column

**db2ts CLEAR EVENTS FOR TEXT**

Deletes events from the `SYSIBMTS.TSEVENT` view, an events view that provides information about indexing status and errors

**db2ts HELP**

Displays the list of **db2ts** command options and information about specific error messages

---

## db2ts START FOR TEXT

This command:

- Starts the DB2 Text Search instance services that support other DB2 Text Search administration commands and the ability to reference text search indexes in SQL queries.
- Starts services (daemons on UNIX) on the host machine running the DB2 database server. These services are responsible for the scheduling of text search index updates on the DB2 database server and for text search engine processing. Instance services are started under the authorization-name of the DB2 instance owner. If the instance services are already running, the command has no effect.

For execution, the command needs to be prefixed with **db2ts** at the command line.

### Authorization

Instance owner must hold DBADM with DATAACCESS authority for the current DB2 instance.

### Required connection

This command must be issued from the DB2 database server.

### Command syntax

▶▶—START FOR TEXT—▶▶

### Command parameters

None

---

## db2ts STOP FOR TEXT

This command stops the DB2 Text Search instance services.

For execution, the command needs to be prefixed with **db2ts** at the command line.

### Authorization

Instance owner must hold DBADM with DATAACCESS authority for the current DB2 instance.

### Required connection

This command must be issued from the DB2 database server.

### Command syntax

▶▶—STOP FOR TEXT—▶▶

## Command parameters

None

## Usage notes

Note that multiple commands cannot be executed concurrently on a text search index if they may conflict. If this command is issued while a conflicting command is running, an error will occur and the command will fail, after which you can try to run the command again. To avoid interrupting the execution of currently running commands, make sure no other administrative commands like update index are still active before issuing the stop command. Some of the conflicting commands are:

- **DROP INDEX**
- **UPDATE INDEX**
- **CLEAR EVENTS FOR INDEX**
- **ALTER INDEX**
- **DISABLE DATABASE FOR TEXT**
- **STOP FOR TEXT**

---

## db2ts CLEANUP FOR TEXT

This command cleans up any DB2 Text Search collections within the instance that are obsolete. A collection can become obsolete if:

- A database with text search indexes is dropped before DB2 Text Search has been disabled for the database.
- A table is dropped before the text search indexes, associated with it, have been dropped.

**Note:** A text search collection refers to the underlying representation of a text search index. There is a one-to-one relationship between a text search collection and a text search index. While the commands operate on text search indexes, tools operate on text search collections. Query the SYSIBMTS.TSCOLLECTIONNAMES catalog table to determine the text search collection for a text search index. See “Administration Tool for DB2 Text Search” for additional information.

For execution, the command needs to be prefixed with **db2ts** at the command line.

## Authorization

To issue the command successfully, the user must be the DB2 instance owner with DBADM and DATAACCESS authority.

## Required connection

This command must be issued from the DB2 database server.

## Command syntax

▶▶—CLEANUP FOR TEXT—◀◀



DATAACCESS privileges to the instance owner, in case the instance owner misses these privileges for this database. For a successful grant of privileges, the user must have SECADM privilege for the database and cannot be the instance owner (a user cannot grant privileges to herself/himself).

## Usage notes

When executed successfully, this command:

- Enables the DB2 Text Search feature for the database.
- Establishes DB2 Text Search database configuration default values in the view SYSIBMTS.TSDEFAULTS.
- Creates the following DB2 Text Search administrative views in the SYSIBMTS schema:
  - SYSIBMTS.TSDEFAULTS
  - SYSIBMTS.TSLOCKS
  - SYSIBMTS.TSINDEXES
  - SYSIBMTS.TSCONFIGURATION
  - SYSIBMTS.TSCOLLECTIONNAMES

The tables are created in the default table space (IBMDEFAULTGROUP) of the database.

Changes to the file system on the DB2 server: None.

---

## db2ts DISABLE DATABASE FOR TEXT

This command somewhat reverses the changes (e.g., drops the text-search related tables and view) done by the command **ENABLE DATABASE FOR TEXT**.

When issued, this command:

- Disables the DB2 Text Search feature for the database.
- Will delete tables and views such as:
  - SYSIBMTS.TSDEFAULTS
  - SYSIBMTS.TSLOCKS
  - SYSIBMTS.TSINDEXES
  - SYSIBMTS.TSCONFIGURATION
  - SYSIBMTS.TSCOLLECTIONNAMES

The tables are removed from the default table space (IBMDEFAULTGROUP) of the database.

- If the **FORCE** option is specified, all text index information is removed from the database and all associated collections are deleted. In addition, the text service is updated to remove any remaining update schedule information. See the “db2ts DROP INDEX command” for reference.

For execution, the command needs to be prefixed with **db2ts** at the command line.

## Authorization

The privileges held by the authorization ID of the statement must include DBADM authority.

## Prerequisite

Instance owner must hold DBADM with DATAACCESS authority. The SYSADM no longer holds SECADM nor DBADM privilege in Version 9.7. SECADM must explicitly grant DBADM with DATAACCESS authority to the instance owner before running the **DISABLE DATABASE** command.

## Required connection

Database

## Command syntax

```
►►—DISABLE DATABASE FOR TEXT—FORCE—| connection options —|►►
```

### connection options:

```
|—CONNECT TO—database-name—USER—username—USING—password—|
```

## Command parameters

### FORCE

Specifies that all text search indexes be forcibly dropped from the database.

If this option is not specified and text search indexes are defined for this database, the command will fail.

If this option is specified and DB2 Text Search service has not been started (the db2ts **START FOR TEXT** command has not been issued), the text search indexes (collections) are not dropped.

### CONNECT TO *database-name*

This clause specifies the database to which a connection will be established. The database must be on the local system. If specified, this clause takes precedence over the environment variable DB2DBDFT. This clause can be omitted if the following are all true:

- The DB2DBDFT environment variable is set to a valid database name.
- The user running the command has the required authorization to connect to the database server.

### USER *username* USING *password*

This clause specifies the authorization name and password that will be used to establish the connection.

## Usage notes

This command does not influence the DB2 Net Search Extender enablement status of the database. It deletes the DB2 Text Search catalog tables and views that are created by the **ENABLE FOR TEXT** command.

Before dropping a DB2 database that has text search index definitions, issue this command and make sure that the text indexes and collections have been removed successfully.

If some indexes could not be deleted using the **FORCE** option, the collection names are written to the **db2diag** log file. If the text search index command **DISABLE DATABASE FOR TEXT** is not executed before the CLP command **DROP DATABASE**, the text search index services must also be cleaned up using the **CLEANUP FOR TEXT** command. See the **DROP INDEX** command for more about dropping indexes, and the **CLEANUP FOR TEXT** command for information about text search collections and their relationship to text search indexes.

**Note:** The user is discouraged from usage that results in orphaned collections, i.e., collections that remain defined on the text search server but are not used by DB2. Here are some cases that cause orphaned collections:

- When a **DROP DATABASE** CLP command or **DROP TABLE** statement is executed without running a **DISABLE DATABASE FOR TEXT** command.
- When a **DISABLE DATABASE FOR TEXT** command is executed using the **FORCE** option.
- Some other error conditions. The **CLEANUP FOR TEXT** command can be used in some scenarios.

Note that multiple commands cannot be executed concurrently on a text search index if they may conflict. If this command is issued while a conflicting command is running, an error will occur and the command will fail, after which you can try to run the command again. Some of the conflicting commands are:

- **DROP INDEX**
- **UPDATE INDEX**
- **CLEAR EVENTS FOR INDEX**
- **ALTER INDEX**
- **DISABLE DATABASE FOR TEXT**
- **STOP FOR TEXT**

---

## db2ts CLEAR COMMAND LOCKS

Removes all command locks for a specific text search index or for all text search indexes in the database. A command lock is created at the beginning of a text search index command, and is destroyed when it is done. It prevents undesirable conflict between different commands.

A cleanup is done automatically of all locks associated with processes that are no longer alive. This is done to make a text search index accessible to a new search request.

Use of this command is required in the rare case that locks remain in place due to an unexpected system behavior, and need to be cleaned up explicitly.

For execution, the command needs to be prefixed with **db2ts** at the command line.

### Authorization

The *username* for the database connection must have DBADM authority if an index name is not specified. For clearing a command lock on a specific index, the *username* for the database connection must have CONTROL privilege on the table for which the text search index was created.

## Prerequisite

Instance owner must hold DBADM with DATAACCESS authority. The SYSADM no longer holds SECADM nor DBADM privilege in Version 9.7. SECADM must explicitly grant DBADM with DATAACCESS authority to the instance owner before running the **CLEAR COMMAND LOCKS** command.

## Required connection

Database

## Command syntax

```
►► CLEAR COMMAND LOCKS ───────────────────┬── FOR TEXT ───────────────────►
 └──┬── FOR INDEX index-name ───┘
 |
 └── connection options ───►
```

### connection options:

```
┌──┐
└── CONNECT TO database-name ───────────┬── USER username USING password ─┘
```

## Command parameters

### FOR INDEX *index-name*

The name of the index as specified in the **CREATE INDEX** command.

### CONNECT TO *database-name*

This clause specifies the database to which a connection will be established.

The database must be on the local system. If specified, this clause takes precedence over the environment variable DB2DBDFT. This clause can be omitted if the following are all true:

- The DB2DBDFT environment variable is set to a valid database name.
- The user running the command has the required authorization to connect to the database server.

### USER *username* USING *password*

This clause specifies the authorization name and password that will be used to establish the connection.

## Usage notes

You would invoke this command because the process owning the command lock is dead. In this case, the command (represented by the lock) may not have completed, and the index may not be operational. You need to take appropriate action. For example, the process executing the **DROP INDEX** command dies suddenly. It has deleted some index data, but not all the catalog and collection information. The command lock is left intact. After clearing the **DROP INDEX** command lock, you may wish to re-execute the **DROP INDEX** command. In another example, the process executing the **CREATE INDEX** command dies suddenly. It has created some index catalog and collection information, but not all. The command lock is left intact. After clearing the **CREATE INDEX** command lock, you can execute the **DROP INDEX** and **CREATE INDEX** commands.

When this command is issued, the content of the DB2 Text Search view SYSIBMTS.TSLOCKS is updated.

---

## db2ts CREATE INDEX

Creates a text search index for a text column which allows the column data to be searched using text search functions.

The index will not contain any data until the text search **UPDATE INDEX** command is explicitly executed by the user, or implicitly issued by the text search service, according to the defined update frequency for the index.

For execution, the command needs to be prefixed with **db2ts** at the command line.

### Authorization

The privileges held by the authorization ID of the statement must include at least one of the following:

One of:

- CONTROL privilege on the table on which the index is defined
  - INDEX privilege on the table on which the index is defined
- and one of the following:
- IMPLICIT\_SCHEMA authority on the database, if the implicit or explicit schema name of the index does not exist
  - CREATEIN privilege on the schema, if the schema name of the index refers to an existing schema
- DBADM authority

### Prerequisite

Instance owner must hold DBADM with DATAACCESS authority. The SYSADM no longer holds SECADM nor DBADM privilege in Version 9.7. SECADM must explicitly grant DBADM with DATAACCESS authority to the instance owner before running the **CREATE INDEX** command.

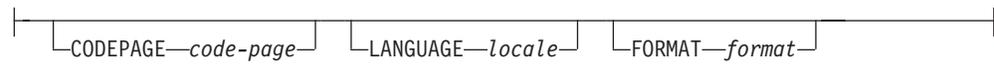
### Required connection

Database

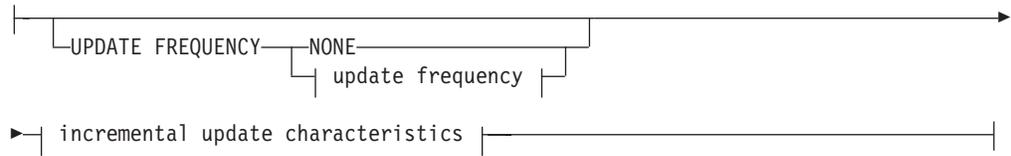
### Command syntax

```
►► CREATE INDEX index-name FOR TEXT ON table-name
► ┌ (text-column-name)
 └ (function-name (text-column-name))
► | text default information | update characteristics |
► | storage options | index configuration options | connection options |►►
```

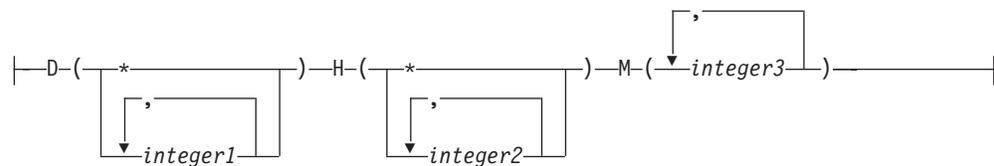
**text default information:**



**update characteristics:**



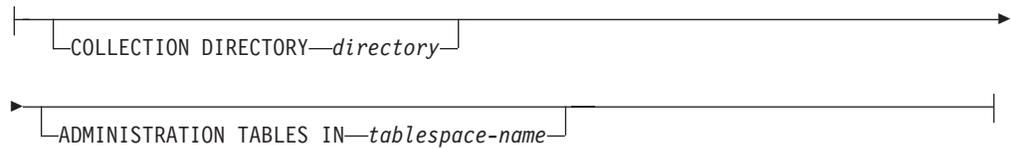
**update frequency:**



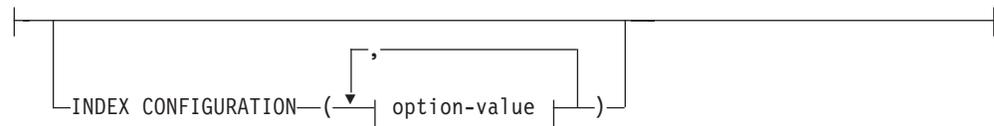
**incremental update characteristics:**



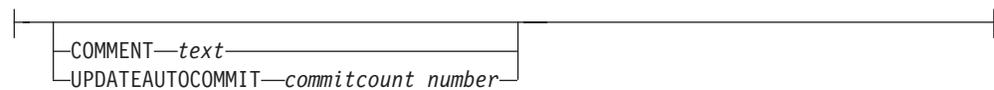
**storage options:**



**index configuration options:**



**option-value:**



**connection options:**

```
CONNECT TO database-name
 USER username USING password
```

## Command parameters

### **CREATE INDEX** *index-name*

Specifies the name of the index (optionally schema qualified) to be created, that will uniquely identify the text search index within the database. The index name must adhere to the naming restrictions for DB2 indexes.

### **ON** *table-name*

The table name containing the text column. Text search indexes cannot be created on the following tables:

- range-partitioned tables
- federated tables
- materialized query tables
- views

### *text-column-name*

The column name of the column to be indexed. The column must be of one of the following data types: CHAR, VARCHAR, LONG VARCHAR, CLOB, DBCLOB, BLOB, GRAPHIC, VARGRAPHIC, LONG VARGRAPHIC, or XML. If the data type of the column is not one of these, use a transformation function specified with *function-schema.function-name* to convert the column type to one of the valid types. Alternatively, you can specify a user-defined external function that accesses the text documents to be indexed. Only a single text search index can be created for a column.

### *function-name*(*text-column-name*)

Specifies the schema qualified name, conforming to DB2 naming conventions, of an external scalar function that accesses text documents in a column that is not of a supported type for text searching. Performs a data type conversion of that value and returns the value as one of the supported data types for text searching. Its task is to perform a column type conversion. This function must take only one parameter and return only one value.

### **CODEPAGE** *code-page*

Specifies the DB2 code page (CODEPAGE) to be used when indexing text documents. The default value is specified by the value in the view SYSIBMTS.TSDEFAULTS, where DEFAULTNAME='CODEPAGE' (which happens to be the database code page). This argument only applies to binary data types, i.e., the column type or return type from a transformation function must be BLOB or character-type FOR BIT DATA.

### **LANGUAGE** *locale*

Specifies the language to be used by DB2 Text Search for language specific processing of a document during indexing. If you do not specify a locale, the database territory will be used to determine the default setting for **LANGUAGE**. If you would like to have your documents automatically scanned to determine the locale, specify *locale* as AUTO.

### **FORMAT** *format*

Specifies the format of text documents in the column. The supported formats include: TEXT, XML, HTML and INSO. DB2 Text Search needs this information when indexing documents. If the format is not specified, the default value is used. The default value is in the view SYSIBMTS.TSDEFAULTS, where DEFAULTNAME='FORMAT'. For columns of data type XML, the default

format 'XML' is used, regardless of the value of DEFAULTNAME. Rich text support must be installed to use the INSO format.

#### **UPDATE FREQUENCY**

Specifies the frequency with which index updates are made. The index will be updated, if the number of changes is at least the value set for **UPDATE MINIMUM**. The update frequency **NONE** indicates that no further index updates will be made. This can be useful for a text column in a table with data that will not change. It is also useful when the user intends to manually update the index (using the **UPDATE INDEX** command). Automatic updates can only be done if the **START FOR TEXT** command has been run and the DB2 Text Search instance services are running.

The default frequency value is taken from the view SYSIBMTS.TSDEFAULTS, where DEFAULTNAME='UPDATEFREQUENCY'.

#### **NONE**

No further index updates are made. The update has to be started manually.

**D** The day(s) of the week when the index is updated.

- \* Every day of the week.

*integer1*

Specific days of the week, from Sunday to Saturday: 0 to 6

**H** The hour(s) of the specified day(s) when the index is updated.

- \* Every hour of the day.

*integer2*

Specific hours of the day, from midnight to 11 pm: 0 to 23

**M** The minute(s) of the specified hour(s) when the index is updated.

*integer3*

Specified as top of the hour (0), or in multiples of 5 minute increments after the hour: 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 or 55

#### **UPDATE MINIMUM** *minchanges*

Specifies the minimum number of changes to text documents before the index is updated incrementally at the time specified in **UPDATE FREQUENCY**. Positive integer values only are allowed. The default value is taken from the view SYSIBMTS.TSDEFAULTS, where DEFAULTNAME='UPDATEMINIMUM'.

**Note:** This value is ignored during an **UPDATE INDEX** command (unless the **USING UPDATE MINIMUM** option is used there). A small value increases consistency between the table column and the text search index. However, it also causes higher performance overhead.

#### **COLLECTION DIRECTORY** *directory*

The directory in which the text search index is stored. By default the collection data will be located in DBPATH/db2collections where the value for DBPATH is the file path used for creating the database. You must specify the absolute path. The maximum length of the absolute path name is 215 characters. Different indexes are organized in subdirectories named *index identifier* under COLLECTION DIRECTORY where index identifier is a system generated identifier.

#### **ADMINISTRATION TABLES IN** *tablespace-name*

Specifies the name of an existing regular table space for the administration tables created for the index. If not specified, the table space of the base table for which the index is being created is used.

## INDEX CONFIGURATION (*option-value*)

Specifies additional index related values as option value string pairs. The following values are supported:

Table 6. Specifications for *option-value*

| Option           | Value                     | Data type                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------|---------------------------|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COMMENT          | <i>text</i>               | String value less than 512 bytes | Adds a string comment value to the REMARKS column in the DB2 Text Search catalog view TSINDEXES. It also adds the string comment value as the description of the collection.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| UPDATEAUTOCOMMIT | <i>commitcount_number</i> | Integer                          | Starting with Version 9.7 Fix Pack 3 and later fix packs, this specifies the number of index updates after which a commit is executed to automatically preserve the previous work for either initial or incremental updates. <ul style="list-style-type: none"><li>• For initial updates, the index update will process batches of documents from a base table after the trigger to capture data updates is activated. After the amount of documents updated reaches the COMMITCOUNT number, the server will do an interim commit. Log entries generated by unprocessed documents will be removed from the staging table. Using the UPDATEAUTOCOMMIT option for an initial text index update will lead to a significant increase of execution time.</li><li>• For incremental updates, log entries which have been processed will be removed correspondingly from the staging table with each interim commit. COMMITCOUNT counts the number of documents updated, not the number of staging table entries.</li></ul> |

**Remember:** Non-numeric values must be enclosed in single quotes. A single quote character within a string value must be represented by two consecutive single quotes.

**Example:**

```
INDEX CONFIGURATION (COMMENT 'Index on User''s Guide column')
```

### CONNECT TO *database-name*

This clause specifies the database to which a connection will be established. The database must be on the local system. If specified, this clause takes precedence over the environment variable DB2DBDFT. This clause can be omitted if the following are all true:

- The DB2DBDFT environment variable is set to a valid database name.
- The user running the command has the required authorization to connect to the database server.

### USER *username* USING *password*

This clause specifies the authorization name and password that will be used to establish the connection.

## Usage notes

All limits and naming conventions, that apply to DB2 database objects and queries, also apply to DB2 Text Search features and queries. DB2 Text related identifiers must conform to the DB2 naming conventions. In addition, there are some additional restrictions. For example, these identifiers can only be of the form:

```
[A-Za-z][A-Za-z0-9@#$_]*
```

or

```
"[A-Za-z][A-Za-z0-9@#$_]*"
```

With the successful execution of the **CREATE INDEX** command:

- DB2 Text Search server data is updated. A collection of name *instance\_database-name\_index-identifier\_number* is created, as in the following example:

```
tigertail_MYTSDB_TS250517_0000
```

The collection name can be retrieved from the SYSIBM.TSCOLLECTIONNAMES view (column COLLECTIONNAME).

- DB2 Text Search catalog information is updated. An index staging table is created in the specified table space with appropriate DB2 indexes. In addition, an index event table is created in the specified table space.
- When DB2 Text Search coexists with DB2 Net Search Extender and an active Net Search Extender index already exists for the table column, the new text index is set to inactive.
- The newly created text search index is not automatically populated. The **UPDATE INDEX** command must be executed either manually or automatically (as a result of an update schedule having been defined for the index through the specification of the **UPDATE FREQUENCY** option) for the text search index to be populated.
- The Text Search index data file on the DB2 database server is updated. Scheduled update information is recorded for each index in the instance.

Usage restrictions:

- A primary key must be defined for the table. In DB2 Text Search, a multi-column DB2 primary key can be used without type limitations. The number of primary key columns is limited to 2 columns less than the number of primary key columns allowed by DB2.
- The total length of all primary key columns for a table with DB2 Text Search indexes is limited to 15 bytes less than the maximum total primary key length allowed by DB2. Refer to the restrictions of the DB2 CREATE INDEX statement.

Note that multiple commands cannot be executed concurrently on a text search index if they may conflict. If this command is issued while a conflicting command is running, an error will occur and the command will fail, after which you can try to run the command again. Some of the conflicting commands are:

- **DROP INDEX**
- **UPDATE INDEX**
- **CLEAR EVENTS FOR INDEX**
- **ALTER INDEX**
- **DISABLE DATABASE FOR TEXT**
- **STOP FOR TEXT**

---

## db2ts DROP INDEX

Drops an existing text search index.

For execution, the command needs to be prefixed with **db2ts** at the command line.

## Authorization

The privileges held by the authorization ID of the statement must include at least one of the following:

- CONTROL privilege on the table on which the index is defined
- DBADM authority

## Prerequisite

Instance owner must hold DBADM with DATAACCESS authority. The SYSADM no longer holds SECADM nor DBADM privilege in Version 9.7. SECADM must explicitly grant DBADM with DATAACCESS authority to the instance owner before running the **DROP INDEX** command.

## Required connection

Database

## Command syntax

```
►► DROP INDEX index-name FOR TEXT | connection options |
```

### connection options:

```
| CONNECT TO database-name | USER username USING password |
```

## Command parameters

### **DROP INDEX** *index-name*

The schema and name of the index as specified in the **CREATE INDEX** command. It uniquely identifies the text search index in a database.

### **CONNECT TO** *database-name*

This clause specifies the database to which a connection will be established. The database must be on the local system. If specified, this clause takes precedence over the environment variable DB2DBDFT. This clause can be omitted if the following are all true:

- The DB2DBDFT environment variable is set to a valid database name.
- The user running the command has the required authorization to connect to the database server.

### **USER** *username* **USING** *password*

This clause specifies the authorization name and password that will be used to establish the connection.

## Usage notes

Note that multiple commands cannot be executed concurrently on a text search index if they may conflict. If this command is issued while a conflicting command is running, an error will occur and the command will fail, after which you can try to run the command again. Some of the conflicting commands are:

- **DROP INDEX**

- **UPDATE INDEX**
- **CLEAR EVENTS FOR INDEX**
- **ALTER INDEX**
- **DISABLE DATABASE FOR TEXT**
- **STOP FOR TEXT**

Dropping the user table in DB2 does not trigger the dropping of text search indexes. They must be dropped manually before or after dropping the table (preferably before dropping the table). After a text search index is dropped, text search is no longer possible on the corresponding text column. If you plan to create a new text search on the same text column, you must first disconnect from the database and then reconnect before creating the new text search index.

Changes to the database:

- Update the DB2 Text Search catalog information.
- Drop the index staging/event tables.
- Delete triggers on the user text table.
- The collection associated with the DB2 Text Search index definition is destroyed.

---

## db2ts ALTER INDEX

This command changes the update characteristics of an index.

For execution, the command needs to be prefixed with **db2ts** at the command line.

### Authorization

The privileges held by the authorization ID of the statement must include at least one of the following:

- CONTROL privilege on the table on which the text index is defined
- DBADM authority

### Prerequisite

The instance owner must hold DBADM with DATAACCESS authority. The SYSADM no longer holds SECADM nor DBADM privilege in Version 9.7. SECADM must explicitly grant DBADM with DATAACCESS authority to the instance owner before running the **ALTER INDEX** command.

### Required connection

Database

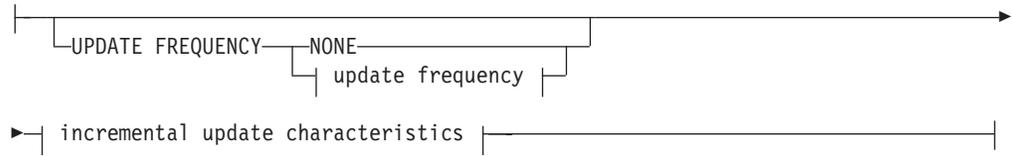
### Command syntax

```

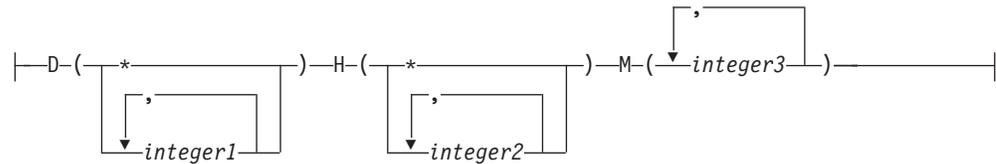
▶▶—ALTER INDEX—index-name—FOR TEXT—| update characteristics |—| options |—▶
▶—| connection options |—————▶▶

```

**update characteristics:**



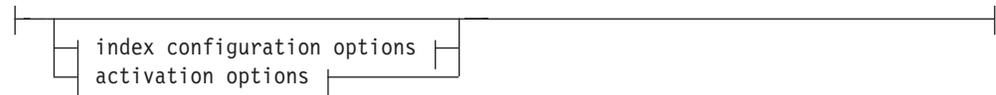
**update frequency:**



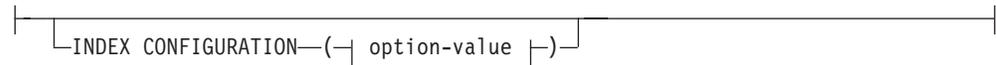
**incremental update characteristics:**



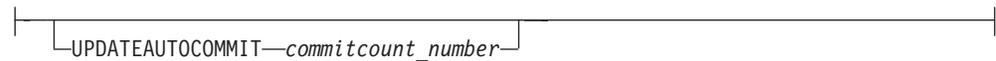
**options:**



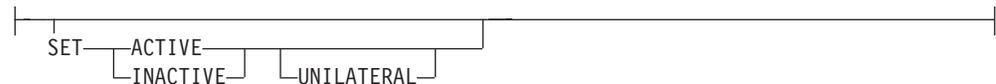
**index configuration options:**



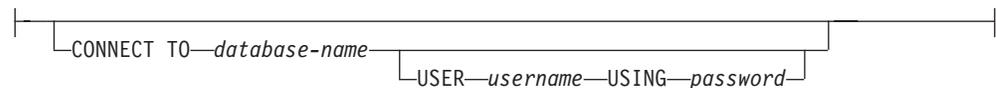
**option-value:**



**activation options:**



**connection options:**



## Command parameters

### **ALTER INDEX** *index-name*

The schema and name of the index as specified in the **CREATE INDEX** command. It uniquely identifies the text search index in a database.

### **UPDATE FREQUENCY**

Specifies the frequency with which index updates are made. The index will be updated, if the number of changes is at least the value set for **UPDATE MINIMUM**. The update frequency **NONE** indicates that no further index updates will be made. This can be useful for a text column in a table with data that will not change. It is also useful when the user intends to manually update the index (using the **UPDATE INDEX** command). Automatic updates can only be done if the **START FOR TEXT** command has been run and the DB2 Text Search instance services are running.

The default frequency value is taken from the view SYSIBMTS.TSDEFAULTS, where DEFAULTNAME='UPDATEFREQUENCY'.

### **NONE**

No automatic updates will be applied to the text index. Any further index update will have to be started manually.

**D** The day(s) of the week when the index is updated.

\* Every day of the week.

*integer1*

Specific days of the week, from Sunday to Saturday: 0 to 6

**H** The hour(s) of the specified day(s) when the index is updated.

\* Every hour of the day.

*integer2*

Specific hours of the day, from midnight to 11 pm: 0 to 23

**M** The minute(s) of the specified hour(s) when the index is updated.

*integer3*

Specified as top of the hour (0), or in multiples of 5 minute increments after the hour: 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 or 55

If you do not specify the **UPDATE FREQUENCY** option, the frequency settings are left unchanged.

### **UPDATE MINIMUM** *minchanges*

Specifies the minimum number of changes to text documents that must occur before the index is incrementally updated. Multiple changes to the same text document are treated as separate changes. If you do not specify the **UPDATE MINIMUM** option, the setting is left unchanged.

### **INDEX CONFIGURATION** (*option-value*)

Starting with Version 9.7 Fix Pack 3 and later fix packs, this is an optional input argument of type VARCHAR(32K) that allows altering text index configuration settings. The following option is supported:

Table 7. Specifications for option-value

| Option           | Value                     | Data type | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------|---------------------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UPDATEAUTOCOMMIT | <i>commitcount_number</i> | Integer   | <p>Specifies the number of index updates after which a commit is executed to preserve the previous work automatically for either initial or incremental updates.</p> <ul style="list-style-type: none"> <li>• For initial updates, the index update will process batches of documents from a base table after the trigger to capture data updates is activated. After the amount of documents updated reaches the COMMITCOUNT number, the server will do an interim commit. Log entries generated by unprocessed documents will be removed from the staging table. Using the UPDATEAUTOCOMMIT option for an initial text index update will lead to a significant increase of execution time.</li> <li>• For incremental updates, log entries which have been processed will be removed correspondingly from staging table with each interim commit. COMMITCOUNT counts the number of documents updated, not the number of staging table entries.</li> </ul> |

*activation options*

Starting with Version 9.7 Fix Pack 3 and later fix packs, this input argument of type integer sets the status of a text index.

**ACTIVE**

Sets the text index status to active

**INACTIVE**

Sets the text index status to inactive

**UNILATERAL**

Specifies a unilateral change which only affects the status of DB2 Text Search indexes. If this argument is specified, only the status of a DB2 Text Search index is changed to active or inactive. Without the UNILATERAL argument, the activation status of the DB2 Text Search and DB2 Net Search Extender indexes is jointly switched so that only one of the text indexes is active.

**Note:** After altering the status of a text search index from active to inactive, or vice versa, issue the **FLUSH PACKAGE CACHE DYNAMIC** command. This removes all cached dynamic SQL statements from the package cache and forces the next request for the same SQL statement to be implicitly compiled by DB2.

**CONNECT TO** *database-name*

This clause specifies the database to which a connection will be established. The database must be on the local system. If specified, this clause takes precedence over the environment variable DB2DBDFT. This clause can be omitted if the following are all true:

- The DB2DBDFT environment variable is set to a valid database name.
- The user running the command has the required authorization to connect to the database server.

**USER** *username* **USING** *password*

This clause specifies the username and password that will be used to establish the connection.

## Usage notes

All limits and naming conventions, that apply to DB2 database objects and queries, also apply to DB2 Text Search features and queries. DB2 Text Search related identifiers must conform to the DB2 naming conventions. In addition, there are some additional restrictions. For example, these identifiers can only be of the form:

```
[A-Za-z][A-Za-z0-9@#$_]*
```

or

```
"[A-Za-z][A-Za-z0-9@#$_]*"
```

Note that multiple commands cannot be executed concurrently on a text search index if they may conflict. If this command is issued while a conflicting command is running, an error will occur and the command will fail, after which you can try to run the command again. Some of the conflicting commands are:

- **ALTER INDEX**
- **CLEAR EVENTS FOR INDEX**
- **DROP INDEX**
- **UPDATE INDEX**
- **DISABLE DATABASE FOR TEXT**
- **STOP FOR TEXT**

Changes to the database: Updates the DB2 Text Search catalog information.

---

## db2ts UPDATE INDEX

This command updates the text search index (collection in DB2 Text Search) to reflect the current contents of the text column with which the index is associated. While the update is being performed, a search is possible. Until completion of the update, the search operates on a partially updated index.

For execution, the command needs to be prefixed with **db2ts** at the command line.

### Authorization

The privileges held by the authorization ID of the statement must include at least one of the following:

- **CONTROL** privilege on the table on which the text index is defined
- **DATAACCESS** authority

### Prerequisite

Instance owner must hold **DBADM** with **DATAACCESS** authority. The **SYSADM** no longer holds **SECADM** nor **DBADM** privilege in Version 9.7. **SECADM** must explicitly grant **DBADM** with **DATAACCESS** authority to the instance owner before running the **UPDATE INDEX** command.

### Required connection

Database



Table 8. Specifications for option-value

| Option           | Value                     | Data type | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------|---------------------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UPDATEAUTOCOMMIT | <i>commitcount_number</i> | Integer   | <p>Specifies the number of index updates after which a commit is executed to preserve the previous work automatically for either initial or incremental updates.</p> <ul style="list-style-type: none"> <li>• For initial updates, the index update will process batches of documents from a base table after the trigger to capture data updates is activated. After the amount of documents updated reaches the COMMITCOUNT number, the server will do an interim commit. Log entries generated by unprocessed documents will be removed from the staging table. Using the UPDATEAUTOCOMMIT option for an initial text index update will lead to a significant increase of execution time.</li> <li>• For incremental updates, log entries which have been processed will be removed correspondingly from staging table with each interim commit. COMMITCOUNT counts the number of documents updated, not the number of staging table entries.</li> </ul> |

**CONNECT TO** *database-name*

This clause specifies the database to which a connection will be established. The database must be on the local system. If specified, this clause takes precedence over the environment variable DB2DBDFT. This clause can be omitted if the following are all true:

- The DB2DBDFT environment variable is set to a valid database name.
- The user running the command has the required authorization to connect to the database server.

**USER** *username* **USING** *password*

This clause specifies the authorization name and password that will be used to establish the connection.

## Usage notes

All limits and naming conventions, that apply to DB2 database objects and queries, also apply to DB2 Text Search features and queries. DB2 Text Search related identifiers must conform to the DB2 naming conventions. In addition, there are some additional restrictions. For example, these identifiers can only be of the form:

```
[A-Za-z][A-Za-z0-9@#$_]*
```

or

```
"[A-Za-z][A-Za-z0-9@#$_]*"
```

This command does not return until all index update processing is completed. The duration depends on the number of documents to be indexed and the number of documents already indexed. The collection name for the index can be retrieved from the SYSIBMTS.TSCOLLECTIONNAMES view (column COLLECTIONNAME).

Note that multiple commands cannot be executed concurrently on a text search index if they may conflict. If this command is issued while a conflicting command is running, an error will occur and the command will fail, after which you can try to run the command again. Some of the conflicting commands are:

- **UPDATE INDEX**
- **CLEAR EVENTS FOR INDEX**
- **ALTER INDEX**
- **DROP INDEX**
- **DISABLE DATABASE FOR TEXT**
- **STOP FOR TEXT**

**Note:** In cases of individual document errors, the documents must be corrected. The primary keys of the erroneous documents can be looked up in the event table for the index. The next **UPDATE INDEX** command will reprocess these documents if the corresponding rows in the user table are modified.

Changes to the database:

- Insert rows to the event table (including parser error information from DB2 Text Search).
- Delete from the index staging table in case of incremental updates.
- Before first update, create triggers on the user text table.
- The collection is updated.
- New or changed documents are parsed and indexed.
- Deleted documents are discarded from the index.

---

## db2ts CLEAR EVENTS FOR INDEX

This command deletes indexing events from an index's event table used for administration. The name of this table can be found in the view SYSIBMTS.TSINDEXES in column EVENTVIEWNAME.

Every index update operation that processes at least one document produces informational and, in some cases, error entries in the event table. For automatic updates, this table has to be regularly inspected. Document specific errors have to be corrected (by changing the document content). After correcting the errors, the events can be cleared (and should be, in order not to consume too much space).

For execution, the command needs to be prefixed with **db2ts** at the command line.

## Authorization

The privileges held by the authorization ID of the statement must include at least one of the following:

- CONTROL privilege on the table on which the index is defined
- DBADM authority

## Prerequisite

Instance owner must hold DBADM with DATAACCESS authority. The SYSADM no longer holds SECADM nor DBADM privilege in Version 9.7. SECADM must explicitly grant DBADM with DATAACCESS authority to the instance owner before running the **CLEAR EVENTS** command.

## Required connection

Database

## Command syntax

```
▶▶ CLEAR EVENTS FOR INDEX—index-name—FOR TEXT—| connection options |▶▶
```

### connection options:

```
|-----|
| CONNECT TO—database-name—|
| USER—username—USING—password—|
```

## Command parameters

*index-name*

The name of the index as specified in the **CREATE INDEX** command. The index name must adhere to the naming restrictions for DB2 indexes.

**CONNECT TO** *database-name*

This clause specifies the database to which a connection will be established. The database must be on the local system. If specified, this clause takes precedence over the environment variable DB2DBDFT. This clause can be omitted if the following are all true:

- The DB2DBDFT environment variable is set to a valid database name.
- The user running the command has the required authorization to connect to the database server.

**USER** *username* **USING** *password*

This clause specifies the authorization name and password that will be used to establish the connection.

## Usage notes

All limits and naming conventions, that apply to DB2 database objects and queries, also apply to DB2 Text Search features and queries. DB2 Text Search related identifiers must conform to the DB2 naming conventions. In addition, there are some additional restrictions. For example, these identifiers can only be of the form:



### *command*

The first keywords that identify a DB2 Text Search command:

- ENABLE
- DISABLE
- CREATE
- DROP
- ALTER
- UPDATE
- CLEAR (for both CLEAR COMMAND LOCKS and CLEAR EVENTS FOR INDEX)
- CLEANUP
- START
- STOP

*sqlcode* SQLCODE for message returned by db2ts command (within or outside the administration stored procedure) or text search query.

*sqlstate* Sqlstate returned by command, administration stored procedure, or text search query.

### *error-identifier*

An identifier is part of the *text-search-error-msg* that is embedded in error messages. This identifier starts with 'CIE' and is of the form CIE $nnnnn$  where  $nnnnn$  is a number. This identifier represents the specific error that is returned upon an error during text search. It may also be returned in an informational message upon completion of a text search command or in the message printed at the completion of a text search administration procedure. If the identifier does not start with 'CIE', then **db2ts help** cannot provide information about the *error-identifier*. For example, db2ts cannot provide help for a message with an *error-identifier* such as IQQR0012E.

## Usage notes

When using a UNIX shell, it might be necessary to supply the arguments to **db2ts** using double-quotes, as in the following example:

```
db2ts "? CIE00323"
```

Without the quotes, the shell tries to match the wildcard with the contents of the working directory and it may give unexpected results.

If the first keyword of any db2ts command is specified, the syntax of the identified command is displayed. For the two db2ts commands that share the same first keyword (**CLEAR COMMAND LOCKS** and **CLEAR EVENTS FOR INDEX**), the syntax of both commands will be displayed when `db2ts help clear` is issued, but each command may be specifically displayed by adding the second keyword to distinguish them, for example `db2ts help clear events`. If a parameter is not specified after **?** or **HELP**, db2ts lists all available db2ts commands.

Specifying a *sqlcode*, *sqlstate*, or CIE *error-identifier* will return information about that code, state, or error identifier. For example,

```
db2ts help SQL20423
```

or

db2ts ? 38H10

or

db2ts ? C1E00323

**Note:** For the following Text Search commands, the command help options listed are not valid at this time:

**db2ts DISABLE DATABASE FOR TEXT**  
[SERVER]

**db2ts ENABLE DATABASE FOR TEXT**  
[SERVER]

**db2ts UPDATE INDEX**  
[REORGANIZE] and [PARSE ONLY]



---

## Chapter 9. DB2 Text Search routines

DB2 Text Search provides several administrative SQL routines for running commands and for returning the result messages of the commands that you run and the result message reason codes.

You can run the following **db2ts** commands using the administrative SQL routines:

- **ENABLE DATABASE**
- **DISABLE DATABASE**
- **CREATE INDEX**
- **DROP INDEX**
- **ALTER INDEX**
- **UPDATE INDEX**
- **CLEAR EVENTS**
- **CLEAR COMMAND LOCKS**

---

### SYSTS\_ADMIN\_CMD stored procedure - Run text search administration commands

The SYSTS\_ADMIN\_CMD procedure is used by applications to run text search administrative commands using the SQL CALL statement.

#### Syntax

```
▶▶ SYSTS_ADMIN_CMD (—command-string—, —message-locale—, —message—) ▶▶
```

The schema is SYSPROC.

#### Procedure parameter

##### *command-string*

An input argument of type VARCHAR (32K) that specifies a single text search index administration command that is to be executed. The command syntax is the same as the DB2 Text Search command with the exception of the connection options. Connection options are not supported through this procedure. Commands that are run through this procedure use the current connection.

##### *message-locale*

An input argument of type VARCHAR (33) that specifies the required language for any error message text returned. If the argument is null or an empty string, or the message files for the specified locale are not available on the server, 'en\_US' is used.

##### *message*

An output argument of type VARCHAR (32K) that specifies a warning or informational message for an operation that is considered successful.

## Authorization

EXECUTE privilege on the SYSTS\_ADMIN\_CMD procedure.

The procedure currently supports the following DB2 Text Search commands:

- ALTER INDEX
- CLEAR COMMAND LOCKS
- CLEAR EVENTS
- CREATE INDEX
- DISABLE DATABASE
- DROP INDEX
- ENABLE DATABASE
- UPDATE INDEX

## Example

Update text search index MYTEXTINDEX in schema DB2TS and return any error messages in English.

```
CALL SYSPROC.SYSTS_ADMIN_CMD
('UPDATE INDEX DB2TS.MYTEXTINDEX FOR TEXT','en_US', ?)";
```

The following example is a sample output from this query.

```
Value of output parameters

Parameter Name : MESSAGE
Parameter Value : CIE00001 Operation completed successfully.

Return Status = 0
```

## Usage notes

- If the execution of the command is not successful, SQLCODE -20427 and SQLSTATE 38H14 is returned with the text search specific error message. For example, if index MYTEXTINDEX already exists and the following statement is issued:

```
CALL SYSPROC.SYSTS_ADMIN_CMD ('CREATE INDEX MYTEXTINDEX FOR TEXT
ON DB2TS.TEXTBOOKS (STORY)', 'en_US', ?)
```

the index creation will fail with the following error message.

```
SQL20427N An error occurred during a text search administration
procedure or command. The error message is "CIE00201 Text search
index "DB2TS ".MYTEXTINDEX" already exists. ". SQLSTATE=38H14
```

- If an SQLCODE is returned by the procedure, the message might be truncated. Full message information can be found in the **db2diag** log files.

---

## SYSTS\_ENABLE procedure - Enable current database for text search

This procedure enables DB2 Text Search for the current database.

This procedure must be issued successfully before text search indexes on columns in tables within the database can be created.

This procedure issues the **ENABLE DATABASE FOR TEXT** text search administration command on the database server.

## Syntax

►►SYSTS\_ENABLE(—*message\_locale*—,—*message*—)◄◄

The schema is SYSPROC.

## Procedure parameters

### *message\_locale*

An input argument of type VARCHAR(33) that specifies the locale to be used for any error message returned. If the argument is null or an empty string, or the message files for the specified locale are not available on the server, 'en\_US' is used.

### *message*

An output argument of type VARCHAR(32K) that specifies a warning or informational message for a successfully completed operation.

## Authorization

The user must have DBADM privilege to execute the ENABLE DATABASE command.

## Examples

*Example 1:* Enable the database for text search and return any error messages in English.

```
CALL SYSPROC.SYSTS_ENABLE('en_US', ?)
```

The following example is a sample output for this query.

```
Value of output parameters

Parameter Name : MESSAGE
Parameter Value : Operation completed successfully.

Return Status = 0
```

*Example 2:* In the following example, SYSTS\_ENABLE is called on a database that is already enabled for text search. This results in an error message to the caller.

```
CALL SYSPROC.SYSTS_ENABLE('en_US', ?)
```

The following example is a sample output for this query.

```
SQL20427N An error occurred during a text search administration
procedure or command. The error message from the text search
product is "CIE00322 Specified or default database already
enabled for text. ". SQLSTATE 38H14
```

## Usage notes

- Text search administration procedures use an existing connection to the database. The current transaction might be committed or rolled back depending on the completion of the procedures. It is therefore recommended to commit all transaction changes before executing a text search administration procedure to avoid any unexpected impact from a commit or rollback in the procedure. One way to achieve this is to turn on AUTOCOMMIT.

**Note:** You may run into a deadlock if a transaction is not committed before executing a text search administration operation, as the same database objects may be affected and the operation waits for the previous step to commit. This may occur, for example, if AUTOCOMMIT is turned off and a table is created followed by creating a text index without an explicit commit between the two transactions.

- When this procedure is run,
  - this procedure creates database objects, such as text search administration catalog tables and views, in the schema SYSIBMTS. These objects are placed in the default table space of the database (IBMDEFAULTGROUP).
  - the established database defaults for text search index are available in view SYSIBMTS.TSDEFAULTS.
  - and when the command has successfully completed, the text search catalog tables and views are created and are available.

---

## SYSTS\_DISABLE procedure - Disable current database for text search

The procedure disables DB2 Text Search for the current database.

Once the Text Search feature has been disabled, text search indexes and commands are no longer available for use with the database.

The procedure issues a **DISABLE DATABASE FOR TEXT** text search administration command on the database server.

### Syntax

►►SYSTS\_DISABLE(—*options*—, —*message\_locale*—, —*message*—)◀◀

The schema is SYSPROC.

### Procedure parameters

#### *options*

An input argument of type VARCHAR(128) that specifies the options to be used when disabling the database. The argument can be set to FORCE. When this value is specified, all indexes are dropped and the Text Search feature is disabled by force. No text search indexes are preserved and no error message or warning is returned. If the argument is null or an empty string, an attempt is made to disable the Text Search feature for the database.

#### *message\_locale*

An input argument of type VARCHAR(33) that specifies the locale to be used for any error message returned. If the argument is null or an empty string, or the message files for the specified locale are not available on the server, 'en\_US' is used.

#### *message*

An output argument of type VARCHAR(32K) that specifies a warning or informational message for a successfully completed operation.

### Authorization

The privileges held by the authorization ID of the statement must include DBADM authority.

## Examples

*Example 1:* In the following example, Text Search is disabled for a database using the SYSTS\_DISABLE procedure. The FORCE option is specified to ensure that the feature is disabled even if text search indexes still exist on tables in the database. Error messages are specified requested to be returned in English. The *message* output parameter is set to an informational message string.

```
CALL SYSPROC.SYSTS_DISABLE('FORCE', 'en_US', ?)
```

The following example is a sample output from this query.

```
Value of output parameters

Parameter Name : MESSAGE
Parameter Value : Operation completed successfully.
```

```
Return Status = 0
```

*Example 2:* In the following example, Text Search is disabled for a database with existing text search indexes using the SYSTS\_DISABLE procedure without specifying the FORCE option. This results in an error message to the caller. It is preferable to drop all existing text search indexes before disabling the Text Search feature or alternatively to specify the FORCE option for the *options* input parameter value.

```
CALL SYSPROC.SYSTS_DISABLE('', 'en_US', ?)
```

The following example is a sample output from this query.

```
SQL20427N An error occurred during a text search administration
procedure or command. The error message is "CIE00326 Text search
index active in specified or default database. ". SQLSTATE 38H14
```

## Usage notes

- Text search administration procedures use an existing connection to the database. The current transaction might be committed or rolled back depending on the completion of the procedures. As such, you might want to commit all transaction changes to avoid any unexpected impact from such a commit or rollback. One way to achieve this is to turn on AUTOCOMMIT.

**Note:** You may run into a deadlock if a transaction is not committed before executing a text search administration operation, as the same database objects may be affected and the operation waits for the previous step to commit. This may occur, for example, if AUTOCOMMIT is turned off and a table is created followed by creating a text index without an explicit commit between the two transactions.

- Multiple procedures or commands cannot be executed concurrently on a text search index if they might conflict. Some of the conflicting procedures and commands are:
  - SYSTS\_ALTER procedure or ALTER INDEX db2ts command
  - SYSTS\_CLEAR\_EVENTS procedure or CLEAR EVENTS FOR INDEX db2ts command
  - SYSTS\_DISABLE procedure or DISABLE DATABASE FOR TEXT db2ts command
  - STOP FOR TEXT db2ts command
  - SYSTS\_UPDATE procedure or UPDATE INDEX db2ts command

If there is a conflict, the procedure returns an SQLCODE -20426 and SQLSTATE 38H13.

- When this procedure is run,
  - the DB2 Text Search catalog information is updated. The index log and event tables are dropped. Triggers on the user text table are deleted.
  - if the **FORCE** option is specified, all text index information is removed from the database and all associated collections are deleted. In addition, the text service is updated to remove any remaining update schedule information. See the "db2ts DROP INDEX command" or "SYSTS\_DROP procedure" for reference.
- This procedure does not influence the DB2 Net Search Extender enablement status of the database. It deletes the DB2 Text Search catalog tables and views that are created by the SYSTS\_ENABLE procedure or the **ENABLE FOR TEXT** command.
- Before dropping a DB2 database that has text search index definitions, run this procedure and make sure that the text indexes and collections have been removed successfully.
- If some indexes could not be deleted using the **FORCE** option, the collection names are written to the **db2diag** log files. If the text search index procedure SYSTS\_DISABLE or the command **DISABLE DATABASE FOR TEXT** is not executed before the CLP command **DROP DATABASE**, the text search index services must also be cleaned up using the **CLEANUP FOR TEXT** command. See the SYSTS\_DROP procedure or **DROP INDEX** command for more about dropping indexes, and the **CLEANUP FOR TEXT** command for information about text search collections and their relationship to text search indexes.

**Note:** The user is discouraged from usage that results in orphaned collections, that is, collections that remain defined on the text search server but are not used by DB2. Here are some cases that may cause orphaned collections:

- When a **DROP DATABASE** CLP command or DROP TABLE statement is executed without running the SYSTS\_DISABLE procedure or a **DISABLE DATABASE FOR TEXT** command.
- When the SYSTS\_DISABLE procedure is run or a **DISABLE DATABASE FOR TEXT** command is executed using the **FORCE** option.
- Some other error conditions. The **CLEANUP FOR TEXT** command can be used in some scenarios.

---

## SYSTS\_CREATE procedure - Create a text search index on a column

This procedure creates a text search index for a text column which allows the column data to be searched using text search functions.

Once the text search index is created, the column can be searched using text search functions in queries. The index will not contain any data until the text search **UPDATE INDEX** command or SYSTS\_UPDATE procedure is explicitly executed by the user, or implicitly executed by the text search instance level services, according to the defined update frequency for the index.

The procedure issues a **CREATE INDEX** text search administration command on the database server.

### Syntax

►► SYSTS\_CREATE (—*index\_schema*—, —*index\_name*—, —| text source |—, —————►  
 ►| options |—, —*message\_locale*—, —*message*—) —————►►

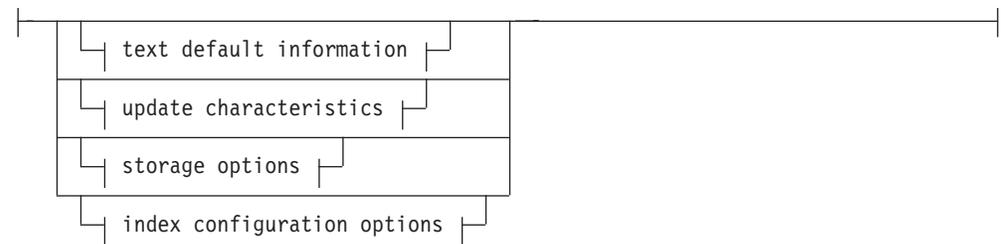
**text source:**

|—*table-name*—(—| text column name |—), —————|

**text column name:**

|—*column-name*—  
 |—*function-name* (*column-name*)—|

**options:**



**text default information:**

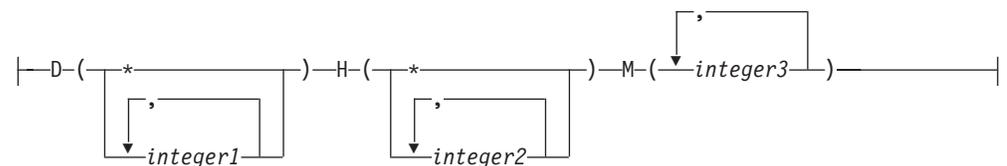
|—CODEPAGE—*code-page*—| —LANGUAGE—*language*—| —FORMAT—*format*—|

**update characteristics:**

|—UPDATE FREQUENCY—NONE—|  
 |—update frequency—|

►| incremental update characteristics | —————|

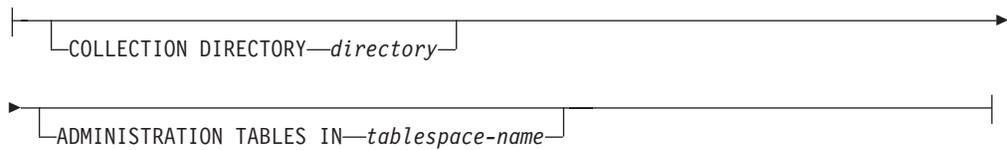
**update frequency:**



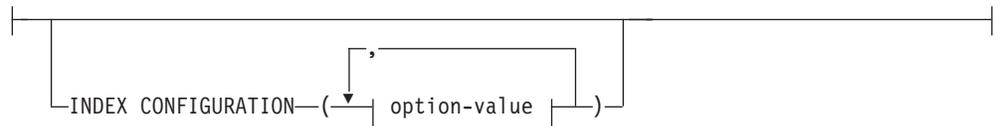
**incremental update characteristics:**

|—UPDATE MINIMUM—*minchanges*—|

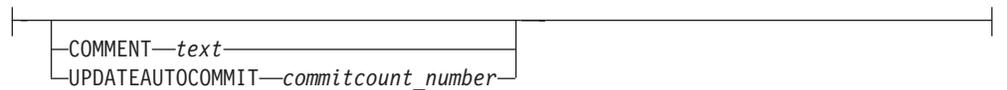
### storage options:



### index configuration options:



### option-value:



The schema is SYSPROC.

## Procedure parameters

### *index\_schema*

An input argument of type VARCHAR(128) that specifies the schema of the text search index. The *index\_schema* must follow the naming restriction for DB2 schema names. If the argument is null or an empty string, the value of CURRENT SCHEMA is used. The *index\_schema* is case-sensitive.

### *index\_name*

An input argument of type VARCHAR(128) that specifies the name of the index. Together with *index\_schema*, it uniquely identifies a text search index in a database. The *index\_name* is case-sensitive.

### text source

An input argument of type VARCHAR(1024) that specifies the name of the column to be indexed. The options are:

#### *table-name*

The table name containing the text column. Text search indexes cannot be created on the following tables:

- range-partitioned tables
- federated tables
- materialized query tables
- views

The *table-name* is case-sensitive.

#### text column name

The column name of the column to be indexed.

#### *column-name*

The column must be of one of the following data types: CHAR, VARCHAR, LONG VARCHAR, CLOB, DBCLOB, BLOB, GRAPHIC,

VARGRAPHIC, LONG VARGRAPHIC, or XML. If the data type of the column is not one of these, use a transformation function specified with *function-schema.function-name* to convert the column type to one of the valid types. Refer to the *function-name (column-name)* for syntax and details. Alternatively, you can specify a user-defined external function that accesses the text documents to be indexed. Only a single text search index can be created for a column. The *column-name* is case-sensitive.

*function-name (column-name)*

Specifies the schema qualified name, that conforms to DB2 naming conventions, of an external scalar function that accesses text documents in a column that is not of a supported type for text searching. Performs a data type conversion of that value, and returns the value as one of the supported data types for text searching. Its task is to perform a column type conversion. This function must take only one parameter and return only one value. The *function-name (column-name)* is case-sensitive.

**options**

An input argument of type VARCHAR(32K) that specifies the options to be used. If no options are needed, the argument can be null or an empty string. The available options are:

**CODEPAGE** *code-page*

Specifies the DB2 code page (CODEPAGE) to be used when indexing text documents. The default value is specified by the value in the view SYSIBM.TSDEFAULTS, where DEFAULTNAME='CODEPAGE' (which happens to be the database code page). This argument only applies to binary data types, that is, the column type or return type from a transformation function must be BLOB or character-type FOR BIT DATA.

**LANGUAGE** *language*

Specifies the language to be used by DB2 Text Search for language specific processing of a document during indexing. If you do not specify a locale, the database territory will be used to determine the default setting for LANGUAGE. If you would like to have your documents automatically scanned to determine the locale, specify locale as AUTO.

**FORMAT** *format*

Specifies the format of text documents in the column. The supported formats include: TEXT, XML, HTML and INSO. DB2 Text Search needs this information when indexing documents. If the format is not specified, the default value is used. The default value is in the view SYSIBM.TSDEFAULTS, where DEFAULTNAME='FORMAT'. For columns of data type XML, the default format 'XML' is used, regardless of the value of DEFAULTNAME. Rich text support must be installed to use the INSO format.

**UPDATE FREQUENCY**

Specifies the frequency with which index updates are made. The index will be updated, if the number of changes is at least the value set for **UPDATE MINIMUM**. The update frequency **NONE** indicates that no further index updates will be made. This can be useful for a text column in a table with data that will not change. It is also useful when the user intends to manually update the index (using the **UPDATE INDEX** command). Automatic updates can only be done if the **START FOR TEXT** command has been run and the DB2 Text Search instance services are running.

The default frequency value is taken from the view SYSIBMTS.TSDEFAULTS, where DEFAULTNAME='UPDATEFREQUENCY'.

**NONE**

No further index updates are made. The update has to be started manually.

**D** The day(s) of the week when the index is updated.

\* Every day of the week.

*integer1*

Specific days of the week, from Sunday to Saturday: 0 to 6

**H** The hour(s) of the specified day(s) when the index is updated.

\* Every hour of the day.

*integer2*

Specific hours of the day, from midnight to 11 pm: 0 to 23

**M** The minute(s) of the specified hour(s) when the index is updated.

*integer3*

Specified as top of the hour (0), or in multiples of 5 minute increments after the hour: 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 or 55

**UPDATE MINIMUM** *minchanges*

Specifies the minimum number of changes to text documents before the index is updated incrementally at the time specified in **UPDATE FREQUENCY**. Positive integer values only are allowed. The default value is taken from the view SYSIBMTS.TSDEFAULTS, where DEFAULTNAME='UPDATEMINIMUM'.

**Note:** This value is ignored during an **UPDATE INDEX** command (unless the **USING UPDATE MINIMUM** option is used there). A small value increases consistency between the table column and the text search index. However, it also causes higher performance usage.

**COLLECTION DIRECTORY** *directory*

The directory in which the text search index is stored. By default the collection data will be located in DBPATH/db2collections where the value for DBPATH is the file path used for creating the database. You must specify the absolute path. The maximum length of the absolute path name is 215 characters. Different indexes are organized in subdirectories named *index identifier* under COLLECTION DIRECTORY where index identifier is a system generated identifier.

**ADMINISTRATION TABLES IN** *tablespace-name*

Specifies the name of an existing regular table space for the administration tables created for the index. If not specified, the table space of the base table for which the index is being created is used.

**INDEX CONFIGURATION** (*option-value*)

Specifies additional index related values as option value string pairs. The following values are supported:

Table 9. Specifications for option-value

| Option            | Value                                | Data type                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|--------------------------------------|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COMMENT           | <i>text</i>                          | String value less than 512 bytes | Adds a string comment value to the REMARKS column in the DB2 Text Search catalog view TSINDEXES. It also adds the string comment value as the description of the collection.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| UPDATEAUTO COMMIT | <i>commitcount</i><br><i>_number</i> | Integer                          | Starting with DB2 Version 9.7 Fix Pack 3, it specifies the number of index updates after which a commit is executed to preserve the previous work automatically for either initial or incremental updates. <ul style="list-style-type: none"> <li>For initial updates, the index update will process batches of documents from a base table after the trigger to capture data updates is activated. After the amount of documents updated reaches the COMMITCOUNT number, the server will do an interim commit. Log entries generated by unprocessed documents will be removed from staging table. Using the UPDATEAUTOCOMMIT option for an initial text index update will lead to a significant increase of execution time.</li> <li>For incremental updates, log entries which have been processed will be removed correspondingly from the staging table with each interim commit. COMMITCOUNT counts the number of documents updated, not the number of staging table entries.</li> </ul> |

**Remember:** Non-numeric values must be enclosed in single quotation marks. A single quote character within a string value must be represented by two consecutive single quotation marks.

**Example:**

```
INDEX CONFIGURATION (COMMENT 'Index on User''s Guide column')
```

*message\_locale*

An input argument of type VARCHAR(33) that specifies the locale to be used for any error message returned. If the argument is null or an empty string, or the message files for the specified locale are not available on the server, 'en\_US' is used.

*message*

An output argument of type VARCHAR(32K) that specifies a warning or informational message for a successfully completed operation.

**Authorization**

The privileges held by the authorization ID of the statement must include at least one of the following authorities:

- CONTROL privilege on the table on which the index is defined
  - INDEX privilege on the table on which the index is defined
- and one of the following authorities:
- IMPLICIT\_SCHEMA authority on the database, if the implicit or explicit schema name of the index does not exist

- CREATEIN privilege on the schema, if the schema name of the index refers to an existing schema
- DBADM authority

## Examples

*Example 1:* In the following example, a text search index with *index\_schema* 'db2ts' and *index\_name* 'myTextIndex' is created using the SYSTS\_CREATE procedure. The option 'UPDATE MINIMUM 10' specifies that at least 10 changes should be made to the text documents associated with the index before an incremental update of the index should be performed. Any error messages are requested to be returned in English. When the underlying text search command runs successfully, the output parameter message is set to indicate the status of the command execution.

```
CALL SYSPROC.SYSTS_CREATE('db2ts', 'myTextIndex',
 'myUserSchema.myBaseTable (myTextColumn)', 'UPDATE MINIMUM 10',
 'en_US', ?)
```

The following example is a sample output from this query.

```
Value of output parameters

Parameter Name : MESSAGE
Parameter Value : Operation completed successfully.
Return Status = 0
```

*Example 2:* In the following example, SYSTS\_CREATE is called to create a text search index with *index\_schema* 'db2ts' and *index\_name* 'myTextIndex'. No options are specified. In this example, the index already exists which results in an error message being returned to the caller.

```
CALL SYSPROC.SYSTS_CREATE('db2ts', 'myTextIndex',
 'myUserSchema.myBaseTable (myTextColumn)', '', 'en_US', ?)
```

The following example is a sample output from this query.

```
SQL20427N An error occurred during a text search administration
procedure or command. The error message is "CIE00201 Text search
index "db2ts"."myTextIndex" already exists. "
```

## Usage notes

- Text search administration procedures use an existing connection to the database. The current transaction might be committed or rolled back depending on the completion of the procedures. As such, you might want to commit all transaction changes to avoid any unexpected impact from such a commit or rollback. One way to achieve this is to turn on AUTOCOMMIT for the connection.

**Note:** You may run into a deadlock if a transaction is not committed before executing a text search administration operation, as the same database objects may be affected and the operation waits for the previous step to commit. This may occur, for example, if AUTOCOMMIT is turned off and a table is created followed by creating a text index without an explicit commit between the two transactions.

- Multiple procedures or commands cannot be executed concurrently on a text search index if they might conflict. Some of the conflicting procedures and commands are:
  - SYSTS\_ALTER procedure or ALTER INDEX db2ts command
  - SYSTS\_CLEAR\_EVENTS procedure or CLEAR EVENTS FOR INDEX db2ts command

- SYSTS\_DISABLE procedure or DISABLE DATABASE FOR TEXT db2ts command
- STOP FOR TEXT db2ts command
- SYSTS\_UPDATE procedure or UPDATE INDEX db2ts command

If there is a conflict, the procedure returns an SQLCODE -20426 and SQLSTATE 38H13.

- With the successful execution of the **CREATE INDEX** command:
  - DB2 Text Search server data is updated. A collection of name *instance\_database-name\_index-identifier\_number* is created, as in the following example:  
tigertail\_MYTSDB\_TS250517\_0000
  - The collection name can be retrieved from the SYSIBMTS.TSCOLLECTIONNAMES view (column COLLECTIONNAME).
  - DB2 Text Search catalog information is updated. An index staging table is created in the specified table space with appropriate DB2 indexes. In addition, an index event table is created in the specified table space.
  - When DB2 Text Search coexists with DB2 Net Search Extender and an active Net Search Extender index already exists for the table column, the new text index is set to inactive.
  - The newly created text search index is not automatically populated. The SYSTS\_UPDATE procedure or **UPDATE INDEX** command must be executed either manually or automatically (as a result of an update schedule having been defined for the index through the specification of the **UPDATE FREQUENCY** option) for the text search index to be populated.
  - The Text Search index data file on the DB2 database server is updated. Scheduled update information is recorded for each index in the instance.

Usage restrictions:

- A primary key must be defined for the table. In DB2 Text Search, a multicolumn DB2 primary key can be used without type limitations. The number of primary key columns is limited to 2 columns less than the number of primary key columns allowed by DB2.
- The total length of all primary key columns for a table with DB2 Text Search indexes is limited to 15 bytes less than the maximum total primary key length allowed by DB2. Refer to the DB2 restrictions of DB2 CREATE INDEX statement.

---

## SYSTS\_DROP procedure - Drop a text search index

This procedure drops an existing text search index associated with any table column.

After successful execution of this procedure, text search queries cannot be run on that column.

The procedure issues a **DROP INDEX** text search administration command on the database server.

### Syntax

►►SYSTS\_DROP—(—*index\_schema*—,—*index\_name*—,——————►

►*message\_locale*—,—*message*—)—————▶

The schema is SYSPROC.

## Procedure parameters

### *index\_schema*

An input argument of type VARCHAR(128) that specifies the schema of the text search index. The *index\_schema* must follow the naming restriction for DB2 schema names. If the argument is null or an empty string, the value of CURRENT SCHEMA is used. The *index\_schema* is case-sensitive.

### *index\_name*

An input argument of type VARCHAR(128) that specifies the name of the index. Together with *index\_schema*, it uniquely identifies a text search index in a database. The *index\_name* is case-sensitive.

### *message\_locale*

An input argument of type VARCHAR(33) that specifies the locale to be used for any error message returned. If the argument is null or an empty string, or the message files for the specified locale are not available on the server, 'en\_US' is used.

### *message*

An output argument of type VARCHAR(32K) that specifies a warning or informational message for a successfully completed operation.

## Authorization

The privileges held by the authorization ID of the statement must include at least one of the following authorities:

- CONTROL privilege on the table on which the index is defined
- DBADM authority

## Examples

*Example 1:* In the following example, the text search index that was created with *index\_schema* 'db2ts' and *index\_name* 'myTextIndex' is being dropped. Any error messages are requested to be returned in English. When the procedure succeeds, the output parameter message indicative of the successful operation is returned to the caller.

```
CALL SYSPROC.SYSTS_DROP('db2ts', 'myTextIndex', 'en_US', ?)
```

The following example is a sample output from this query.

```
Value of output parameters

Parameter Name : MESSAGE
Parameter Value : Operation completed successfully.

Return Status = 0
```

*Example 2:* In the following example, SYSTS\_DROP is called to drop a text search index with *index\_schema* 'db2ts' and *index\_name* 'myTextIndex'. This index does not exist and results in an error.

```
CALL SYSPROC.SYSTS_DROP('db2ts', 'myTextIndex', 'en_US', ?)
```

The following example is a sample output from this query.

SQL20427N An error occurred during a text search administration procedure or command. The error message is "CIE00316 Text search index "db2ts"."myTextIndex" does not exist. ". SQLSTATE 38H14

## Usage notes

- Text search administration procedures use an existing connection to the database. The current transaction might be committed or rolled back depending on the completion of the procedures. It is therefore recommended to commit all transaction changes before executing a text search administration procedure to avoid any unexpected impact from a commit or rollback in the procedure. One way to achieve this is to turn on AUTOCOMMIT.

**Note:** You may run into a deadlock if a transaction is not committed before executing a text search administration operation, as the same database objects may be affected and the operation waits for the previous step to commit. This may occur, for example, if AUTOCOMMIT is turned off and a table is created followed by creating a text index without an explicit commit between the two transactions.

- Multiple procedures or commands cannot be executed concurrently on a text search index if they might conflict. Some of the conflicting procedures and commands are:
  - SYSTS\_ALTER procedure or ALTER INDEX db2ts command
  - SYSTS\_CLEAR\_EVENTS procedure or CLEAR EVENTS FOR INDEX db2ts command
  - SYSTS\_DISABLE procedure or DISABLE DATABASE FOR TEXT db2ts command
  - STOP FOR TEXT db2ts command
  - SYSTS\_UPDATE procedure or UPDATE INDEX db2ts command

If there is a conflict, the procedure returns an SQLCODE -20426 and SQLSTATE 38H13.

- Dropping the user table in DB2 does not trigger dropping of indexes, they must be dropped manually before or after dropping the table.
- When this procedure is run,
  - the text search catalog information is updated. The index staging and event tables are dropped. Triggers on the user table are deleted.
  - the index entry in the Text Search Index Data file is deleted. The file contains a persistent representation of update schedules (also empty ones) for each index in the instance.
  - the collection associated with the text search index definition is removed.
- If, after dropping a text search index, you plan to create a new one on the same text column, you must first disconnect from and then reconnect to the database before creating the new text search index.

---

## SYSTS\_ALTER procedure - Change the update characteristics of an index

This procedure changes the update characteristics of an index.

The procedure issues an **ALTER INDEX** text search administration command on the database server.

## Syntax

```
SYSTS_ALTER ((index_schema , index_name , update characteristics)
options (, message_locale , message)
```

### update characteristics:

```
UPDATE FREQUENCY (NONE | update frequency)
UPDATE MINIMUM minchanges
```

### update frequency:

```
D (* (integer1)) H (* (integer2)) M (integer3)
```

### options:

```
index configuration options
activation options
```

### index configuration options:

```
INDEX CONFIGURATION (option-value)
```

### option-value:

```
UPDATEAUTOCOMMIT commitcount_number
```

### activation options:

```
SET (ACTIVE | INACTIVE | UNILATERAL)
```

The schema is SYSPROC.

## Procedure parameters

*index\_schema*

An input argument of type VARCHAR(128) that specifies the schema of the

text search index. The *index\_schema* must follow the naming restriction for DB2 schema names. If the argument is null or an empty string, the value of CURRENT SCHEMA is used. The *index\_schema* is case-sensitive.

*index\_name*

An input argument of type VARCHAR(128) that specifies the name of the index. Together with *index\_schema*, it uniquely identifies a text search index in a database. The *index\_name* is case-sensitive.

**update characteristics**

An input argument of type VARCHAR(32K) that specifies the alter options. The alter options allowed are as follows:

**UPDATE FREQUENCY**

Specifies the frequency with which index updates are made. The index will be updated, if the number of changes is at least the value set for **UPDATE MINIMUM**. The update frequency **NONE** indicates that no further index updates will be made. This can be useful for a text column in a table with data that will not change. It is also useful when the user intends to manually update the index (using the **UPDATE INDEX** command). Automatic updates can only be done if the **START FOR TEXT** command has been run and the DB2 Text Search instance services are running.

The default frequency value is taken from the view SYSIBMTS.TSDEFAULTS, where DEFAULTNAME='UPDATEFREQUENCY'.

**NONE**

No automatic updates will be applied to the text index. Any further index update will have to be started manually.

**D** The day(s) of the week when the index is updated.

\* Every day of the week.

*integer1*

Specific days of the week, from Sunday to Saturday: 0 to 6

**H** The hour(s) of the specified day(s) when the index is updated.

\* Every hour of the day.

*integer2*

Specific hours of the day, from midnight to 11 pm: 0 to 23

**M** The minute(s) of the specified hour(s) when the index is updated.

*integer3*

Specified as top of the hour (0), or in multiples of 5 minute increments after the hour: 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 or 55

If you do not specify the **UPDATE FREQUENCY** option, the frequency settings are left unchanged.

**UPDATE MINIMUM** *minchanges*

Specifies the minimum number of changes to text documents that must occur before the index is incrementally updated. Multiple changes to the same text document are treated as separate changes. If you do not specify the **UPDATE MINIMUM** option, the setting is left unchanged.

**INDEX CONFIGURATION** (*option-value*)

Starting with Version 9.7 Fix Pack 3 and later fix packs, this is an optional

input argument of type VARCHAR(32K) that allows altering text index configuration settings. The following option is supported:

Table 10. Specifications for option-value

| Option               | Value                                | Data type | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------------|--------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UPDATEAUTO<br>COMMIT | <i>commitcount</i><br><i>_number</i> | Integer   | <p>Specifies the number of index updates after which a commit is executed to preserve the previous work automatically for either initial or incremental updates.</p> <ul style="list-style-type: none"> <li>For initial updates, the index update will process batches of documents from a base table after the trigger to capture data updates is activated. After the amount of documents updated reaches the COMMITCOUNT number, the server will do an interim commit. Log entries generated by unprocessed documents will be removed from staging table. Using the UPDATEAUTOCOMMIT option for an initial text index update will lead to a significant increase of execution time.</li> <li>For incremental updates, log entries which have been processed will be removed correspondingly from staging table with each interim commit. COMMITCOUNT counts the number of documents updated, not the number of staging table entries.</li> </ul> |

#### *activation options*

Starting with Version 9.7 Fix Pack 3 and later fix packs, this input argument of type integer sets the status of a text index.

#### **ACTIVE**

Sets the text index status to active

#### **INACTIVE**

Sets the text index status to inactive

#### **UNILATERAL**

Specifies a unilateral change which only affects the status of DB2 Text Search indexes. If this argument is specified, only the status of a DB2 Text Search index is changed to active or inactive. Without the UNILATERAL argument, the activation status of the DB2 Text Search and DB2 Net Search Extender indexes is jointly switched so that only one of the text indexes is active.

**Note:** After altering the status of a text search index from active to inactive, or vice versa, issue the **FLUSH PACKAGE CACHE DYNAMIC** command. This removes all cached dynamic SQL statements from the package cache and forces the next request for the same SQL statement to be implicitly compiled by DB2.

#### *message\_locale*

An input argument of type VARCHAR(33) that specifies the locale to be used for any error message returned. If the argument is null or an empty string, or the message files for the specified locale are not available on the server, 'en\_US' is used.

#### *message*

An output argument of type VARCHAR(32K) that specifies a warning or informational message for a successfully completed operation.

## Authorization

The privileges held by the authorization ID of the statement must include at least one of the following authorities:

- CONTROL privilege on the table on which the text index is defined
- DBADM authority

## Examples

*Example 1:* In the following example, the update characteristics of a text search index are being altered. This index was originally created with *index\_schema* 'db2ts' and *index\_name* 'myTextIndex'. By using 'UPDATE FREQUENCY NONE', the intention is to make no further updates to the text search index as possibly no changes are expected for the associated table column. Any error messages are requested to be returned in English. When the procedure succeeds, the output parameter message indicative of the successful operation is returned to the caller.

```
CALL SYSPROC.SYSTS_ALTER('db2ts', 'myTextIndex',
 'UPDATE FREQUENCY NONE', 'en_US', ?)
```

The following example is a sample output from this query.

```
Value of output parameters

Parameter Name : MESSAGE
Parameter Value : Operation completed successfully.
```

```
Return Status = 0
```

*Example 2:* In the following example, the SYSTS\_ALTER stored procedure is called to alter the update-characteristics for a text search index with *index\_schema* 'db2ts' and *index\_name* 'myTextIndex'. The intention is to ensure that updates to the index occur every hour on the hour. However, this index does not exist and results in an error.

```
CALL SYSPROC.SYSTS_ALTER('db2ts', 'myTextIndex',
 'update frequency D(*) H(*) M(0)', 'en_US', ?)
```

The following example is a sample output from this query.

```
SQL20427N An error occurred during a text search administration
procedure or command. The error message is "CIE00316 Text search
index "db2ts"."myTextIndex" does not exist. ". SQLSTATE 38H14
```

## Usage notes

- Text search administration procedures use an existing connection to the database. The current transaction might be committed or rolled back depending on the completion of the procedures. As such, you might want to commit all transaction changes to avoid any unexpected impact from such a commit or rollback. One way to achieve this is to turn on AUTOCOMMIT for the connection.

**Note:** You may run into a deadlock if a transaction is not committed before executing a text search administration operation, as the same database objects may be affected and the operation waits for the previous step to commit. This may occur, for example, if AUTOCOMMIT is turned off and a table is created followed by creating a text index without an explicit commit between the two transactions.

- Multiple procedures or commands cannot be run concurrently on a text search index if they might conflict. Some of the conflicting procedures and commands are:
  - SYSTS\_ALTER procedure or ALTER INDEX db2ts command
  - SYSTS\_CLEAR\_EVENTS procedure or CLEAR EVENTS FOR INDEX db2ts command
  - SYSTS\_DISABLE procedure or DISABLE DATABASE FOR TEXT db2ts command
  - SYSTS\_DROP procedure or DROP INDEX db2ts command
  - STOP FOR TEXT db2ts command
  - SYSTS\_UPDATE procedure or UPDATE INDEX db2ts command

If there is a conflict, the procedure returns an SQLCODE -20426 and SQLSTATE 38H13.

- When this procedure is run,
  - the content of the DB2 Text Search view SYSIBMTS.TSLOCKS is updated.
  - the index entry in the Text Search Index Data file is updated. The file contains a persistent representation of update schedules (also empty ones) for each index in the instance.
- The result of activating indexes depends on the original index status. The following table describes the results.

Table 11. Status changes without invalid index:

| Initial DB2 Text Search or Net Search Extender Status | Request Active    | Request Active Unilateral | Request Inactive  | Request Inactive Unilateral |
|-------------------------------------------------------|-------------------|---------------------------|-------------------|-----------------------------|
| Active / Inactive                                     | No change         | No change                 | Inactive / Active | Inactive / Inactive         |
| Inactive / Active                                     | Active / Inactive | Error                     | No change         | No change                   |
| Inactive / Inactive                                   | Active / Inactive | Active / Inactive         | Inactive / Active | No change                   |

SQL20427N and CIE0379E error messages are returned for active index conflicts.

---

## SYSTS\_UPDATE procedure - Update the text search index

This procedure updates the text search index to reflect the current contents of the text column with which the index is associated.

While the update is being performed, a search is possible. Until completion of the update, the search operates on a partially updated index.

The procedure issues an **UPDATE INDEX** text search administration command on the database server.

### Syntax

```

▶▶ SYSTS_UPDATE (—index_schema—, —index_name—, ——————▶
▶ —update_options—, —message_locale—, —message—) —————▶▶

```

The schema is SYSPROC.

## Procedure parameters

### *index\_schema*

An input argument of type VARCHAR(128) that specifies the schema of the text search index. The *index\_schema* must follow the naming restriction for DB2 schema names. If the argument is null or an empty string, the value of CURRENT SCHEMA is used. The *index\_schema* is case-sensitive.

### *index\_name*

An input argument of type VARCHAR(128) that specifies the name of the index. Together with *index\_schema*, it uniquely identifies a text search index in a database. The *index\_name* is case-sensitive.

### *update\_options*

An input argument of type VARCHAR(32K) that specifies update options. The possible values are:

- USING UPDATE MINIMUM: this setting respects the UPDATE MINIMUM settings from the **CREATE INDEX** text search administration command and the SYSTS\_CREATE procedure.
- UPDATEAUTOCOMMIT: this setting overrides the commitcount defined for the text index for the duration of this update execution.
- NULL or an empty string (''): the update is unconditionally started when the procedure is called.

### *message\_locale*

An input argument of type VARCHAR(33) that specifies the locale to be used for any error message returned. If the argument is null or an empty string, or the message files for the specified locale are not available on the server, 'en\_US' is used.

### *message*

An output argument of type VARCHAR(32K) that specifies a warning or informational message for a successfully completed operation.

## Authorization

The privileges held by the authorization ID of the statement must include at least one of the following authorities:

- CONTROL privilege on the table on which the text index is defined
- DATAACCESS authority

## Examples

*Example 1:* In the following example, the text search index that was created with *index\_schema* 'db2ts' and *index\_name* 'myTextIndex' is being updated. A NULL value in the place of the *update\_options* means that an update is unconditionally started when the stored procedure is called. Any error messages are requested to be returned in English. When the procedure succeeds, the output parameter message indicative of the successful operation is returned to the caller.

```
CALL SYSPROC.SYSTS_UPDATE('db2ts', 'myTextIndex', '', 'en_US', ?)
```

The following example is a sample output from this query.

```
Value of output parameters

Parameter Name : MESSAGE
Parameter Value : Operation completed successfully.

Return Status = 0
```

*Example 2:* In the following example, SYSTS\_UPDATE is called to update a text search index with *index\_schema* 'db2ts' and *index\_name* 'myTextIndex'. This index does not exist and results in an error.

```
CALL SYSPROC.SYSTS_UPDATE('db2ts', 'myTextIndex', 'USING UPDATE MINIMUM',
'en_US', ?)
```

The following example is a sample output from this query.

```
SQL20427N An error occurred during a text search administration
procedure or command. The error message is "CIE00316 Text search
index "db2ts"."myTextIndex" does not exist. ". SQLSTATE 38H14
```

## Usage notes

- Text search administration procedures use an existing connection to the database. The current transaction might be committed or rolled back depending on the completion of the procedures. As such, you might want to commit all transaction changes to avoid any unexpected impact from such a commit or rollback. One way to achieve this is to turn on AUTOCOMMIT.
- Multiple procedures or commands cannot be run concurrently on a text search index if they might conflict. Some of the conflicting procedures and commands are:
  - SYSTS\_ALTER procedure or ALTER INDEX db2ts command
  - SYSTS\_CLEAR\_EVENTS procedure or CLEAR EVENTS FOR INDEX db2ts command
  - SYSTS\_DISABLE procedure or DISABLE DATABASE FOR TEXT db2ts command
  - SYSTS\_DROP procedure or DROP INDEX db2ts command
  - STOP FOR TEXT db2ts command
  - SYSTS\_UPDATE procedure or UPDATE INDEX db2ts command

If there is a conflict, the procedure returns an SQLCODE -20426 and SQLSTATE 38H13.

- This procedure does not return until all index update processing is completed. The duration depends on the number of documents to be indexed and the number of documents already indexed. The collection name for the index can be retrieved from the SYSIBMTS.TSCOLLECTIONNAMES view (column COLLECTIONNAME).
- When there are individual document errors, the documents must be corrected. The primary keys of the erroneous documents can be looked up in the event table for the index. By changing the corresponding rows in the user table, the next call to SYSTS\_UPDATE will reprocess these documents.
- When this procedure is run,
  - rows are inserted into the event table (including parser error information). Information is deleted from the index staging table in case of incremental updates. Before the first update, it creates triggers on the user table.
  - the collection is updated: new or changed documents are parsed and indexed and deleted documents are discarded from the index.

---

## SYSTS\_CLEAR\_EVENTS procedure - Delete indexing events from an index's event table

This procedure deletes indexing events from an index's event table used for administration.

The name of the event table can be found in the view SYSIBMTS.TSINDEXES in column EVENTVIEWNAME. Every index update operation that processes at least one document produces informational and, in some cases, error entries in the event table. For automatic updates, the event table has to be regularly inspected. Document specific errors must be corrected by changing the document content. After correcting the errors, the events can be cleared (and should be, in order not to consume too much space).

The procedure issues a **CLEAR EVENTS FOR INDEX** text search administration command on the database server.

## Syntax

```
►►—SYSTS_CLEAR_EVENTS—(—index_schema—,—index_name—,——————►
►—message_locale—,—message—)—————►◄
```

The schema is SYSPROC.

## Procedure parameters

### *index\_schema*

An input argument of type VARCHAR(128) that specifies the schema of the text search index. The *index\_schema* must follow the naming restriction for DB2 schema names. If the argument is null or an empty string, the value of CURRENT SCHEMA is used. The *index\_schema* is case-sensitive.

### *index\_name*

An input argument of type VARCHAR(128) that specifies the name of the index. Together with *index\_schema*, it uniquely identifies a text search index in a database. The *index\_name* is case-sensitive.

### *message\_locale*

An input argument of type VARCHAR(33) that specifies the locale to be used for any error message returned. If the argument is null or an empty string, or the message files for the specified locale are not available on the server, 'en\_US' is used.

### *message*

An output argument of type VARCHAR(32K) that specifies a warning or informational message for a successfully completed operation.

## Authorization

The privileges held by the authorization ID of the statement must include at least one of the following authorities:

- CONTROL privilege on the table on which the index is defined
- DBADM authority

## Examples

*Example 1:* In the following example, SYSTS\_CLEAR\_EVENTS is being called for a text search index that was created with *index\_schema* 'db2ts' and *index\_name* 'myTextIndex'. Any error messages are requested to be returned in English. When the procedure succeeds, the output parameter message indicative of the successful operation is returned to the caller.

```
CALL SYSPROC.SYSTS_CLEAR_EVENTS('db2ts', 'myTextIndex', 'en_US', ?)
```

The following example is a sample output from this query.

```
Value of output parameters

Parameter Name : MESSAGE
Parameter Value : Operation completed successfully.

Return Status = 0
```

*Example 2:* In the following example, SYSTS\_CLEAR\_EVENTS is called to clear the event table entries for a text search index with *index\_schema* 'db2ts' and *index\_name* 'myTextIndex'. This index does not exist and results in an error.

```
CALL SYSPROC.SYSTS_CLEAR_EVENTS('db2ts', 'myTextIndex', 'en_US', ?)
```

The following example is a sample output from this query.

```
SQL20427N An error occurred during a text search administration
procedure or command. The error message is "CIE00316 Text search
index "db2ts"."myTextIndex" does not exist. ". SQLSTATE 38H14
```

## Usage notes

- Text search administration procedures use an existing connection to the database. The current transaction might be committed or rolled back depending on the completion of the procedures. It is therefore recommended to commit all transaction changes before executing a text search administration procedure to avoid any unexpected impact from a commit or rollback in the procedure. One way to achieve this is to turn on AUTOCOMMIT.

**Note:** You may run into a deadlock if a transaction is not committed before executing a text search administration operation, as the same database objects may be affected and the operation waits for the previous step to commit. This may occur, for example, if AUTOCOMMIT is turned off and a table is created followed by creating a text index without an explicit commit between the two transactions.

- Multiple procedures or commands cannot be run concurrently on a text search index if they might conflict. Some of the conflicting procedures and commands are:
  - SYSTS\_ALTER procedure or ALTER INDEX db2ts command
  - SYSTS\_DISABLE procedure or DISABLE DATABASE FOR TEXT db2ts command
  - SYSTS\_DROP procedure or DROP INDEX db2ts command
  - STOP FOR TEXT db2ts command
  - SYSTS\_UPDATE procedure or UPDATE INDEX db2ts command

If there is a conflict, the procedure returns an SQLCODE -20426 and SQLSTATE 38H13.

- When regular updates are scheduled (see UPDATE FREQUENCY options in SYSTS\_CREATE or SYSTS\_ALTER procedures), the event table should be checked regularly.
- To clean up the DB2 Text Search event table for a text search index, use the SYSTS\_CLEAR\_EVENTS procedure or **CLEAR EVENTS FOR INDEX** db2ts command after you have checked the reason for the event and removed the source of the error.
- Ensure that changes have been made to all rows referenced in the event table. By changing the rows in the user table, you ensure that when you run the

SYSTS\_UPDATE procedure or **UPDATE INDEX** db2ts command again, an attempt is made to index the erroneous documents again.

- When this command is issued, the event table is cleared.

---

## SYSTS\_CLEAR\_COMMANDLOCKS procedure - Remove command locks for text search indexes

This procedure removes all command locks for a specific text search index or for all text search indexes in the database.

A command lock is created at the beginning of a text search index command, and is destroyed when the command has completed. It prevents undesirable conflict between different commands.

A cleanup is done automatically of all locks associated with processes that are no longer alive. This is done to make a text search index accessible to a new search request. Use of this procedure is required in the rare case that locks remain in place due to an unexpected system behavior, and need to be cleaned up explicitly.

This procedure issues the **CLEAR COMMAND LOCKS** text search administration command on the database server.

### Syntax

```
►—SYSTS_CLEAR_COMMANDLOCKS—(—index_schema—,—index_name—,——————►
►—message_locale—,—message—)—————►◄
```

The schema is SYSPROC.

### Procedure parameters

#### *index\_schema*

An input argument of type VARCHAR(128) that specifies the schema of the text index. The *index\_schema* must follow the naming restriction for DB2 schema names. If the argument is null or an empty string, the value of CURRENT SCHEMA is used. The *index\_schema* is case-sensitive.

#### *index\_name*

An input argument of type VARCHAR(128) that specifies the name of the index. Together with *index\_schema*, it uniquely identifies a text search index in a database. If the argument is null or an empty string, the procedure deletes command locks for all text search indexes in the database. The *index\_name* is case-sensitive.

#### *message\_locale*

An input argument of type VARCHAR(33) that specifies the locale to be used for any error message returned. If the argument is null or an empty string, or the message files for the specified locale are not available on the server, 'en\_US' is used.

#### *message*

An output argument of type VARCHAR(32K) that specifies a warning or informational message for a successfully completed operation.

## Authorization

The *username* for the database connection must have DBADM authority if an index name is not specified. For clearing a command lock on a specific index, the *username* for the database connection must have CONTROL privilege on the table for which the text search index was created.

## Examples

*Example 1:* In the following example, SYSTS\_CLEAR\_COMMANDLOCKS is issued for a text search index with *index\_schema* 'db2ts' and *index\_name* 'myTextIndex'. Error messages are requested to be returned in English. When the procedure succeeds, the output parameter message indicative of the successful operation is returned to the caller.

```
CALL SYSPROC.SYSTS_CLEAR_COMMANDLOCKS('db2ts', 'myTextIndex', 'en_US', ?)
```

The following example is a sample output from this query.

```
Value of output parameters

Parameter Name : MESSAGE
Parameter Value : Operation completed successfully.

Return Status = 0
```

*Example 2:* In the following example, SYSTS\_CLEAR\_COMMANDLOCKS is called to clear the command locks for a text search index with *index\_schema* 'db2ts' and *index\_name* 'myTextIndex'. This index does not exist and the procedure returns an error message.

```
CALL SYSPROC.SYSTS_CLEAR_COMMANDLOCKS('db2ts', 'myTextIndex', 'en_US', ?)
```

The following example is a sample output from this query.

```
SQL20427N An error occurred during a text search administration
procedure or command. The error message is "CIE00316 Text search
index "db2ts"."myTextIndex" does not exist. ". SQLSTATE 38H14
```

## Usage notes

- Text search administration procedures use an existing connection to the database. The current transaction might be committed or rolled back depending on the completion of the procedures. It is therefore recommended to commit all transaction changes before executing a text search administration procedure to avoid any unexpected impact from a commit or rollback in the procedure. One way to achieve this is to turn on AUTOCOMMIT.

**Note:** You may run into a deadlock if a transaction is not committed before executing a text search administration operation, as the same database objects may be affected and the operation waits for the previous step to commit. This may occur, for example, if AUTOCOMMIT is turned off and a table is created followed by creating a text index without an explicit commit between the two transactions.

- The process and thread information in the view SYSIBMTS.TSLOCKS can be used to check if the thread or process that holds the lock still exists. The locks for existing processes belonging to running text search administration procedure or command (for example, SYSTS\_UPDATE or UPDATE INDEX) should not be cleared.
- You would invoke this procedure because the process owning the command lock is dead. In this case, the command (represented by the lock) may not have

completed, and the index may not be operational. You need to take appropriate action. For example, the process executing the DROP INDEX command dies suddenly. It has deleted some index data, but not all the catalog and collection information. The command lock is left intact. After clearing the DROP INDEX command lock, you may want to re-execute the SYSTS\_DROP procedure. In another example, the process executing the SYSTS\_CREATE procedure dies suddenly. It has created some index catalog and collection information, but not all. The command lock is left intact. After clearing the command lock, you can execute the SYSTS\_DROP and SYSTS\_CREATE procedures.

- When this procedure is run, the content of the DB2 Text Search view SYSIBMTS.TSLOCKS is updated.



---

## Chapter 10. Text search administrative views

DB2 Text Search creates and maintains several administrative views that describe the text search indexes in a database and their properties.

Do not update any of these views unless specifically instructed to do so.

The following views reflect the current configuration of your system:

- Database-level views:
  - SYSIBMTS.TSDEFAULTS
  - SYSIBMTS.TSLOCKS
- Index-level views:
  - SYSIBMTS.TSINDEXES
  - SYSIBMTS.TSCONFIGURATION
  - SYSIBMTS.TSCOLLECTIONNAMES
  - SYSIBMTS.TSEVENT\_#####
  - SYSIBMTS.TSSTAGING\_#####

---

### SYSIBMTS.TSDEFAULTS view

SYSIBMTS.TSDEFAULTS displays all the default values for all text search indexes in a database.

The default values are available as attribute-value pairs in this view.

Table 12. SYSIBMTS.TSDEFAULTS view

| Column name  | Data type     | Nullable? | Description                                            |
|--------------|---------------|-----------|--------------------------------------------------------|
| DEFAULTNAME  | VARCHAR (30)  | NO        | Database default parameters for text search            |
| DEFAULTVALUE | VARCHAR (512) | NO        | Values for database default parameters for text search |

The following values are used as defaults for the db2ts **CREATE INDEX**, **ALTER INDEX**, **UPDATE INDEX**, and **CLEAR EVENTS FOR INDEX** commands:

- CODEPAGE. The initial default code page for new indexes is the database code page.
- FORMAT. The initial default for the document format is plain text.
- LANGUAGE. The initial default for document indexing is en\_US.
- UPDATEFREQUENCY. The initial default for the update schedule for new indexes is NONE.
- UPDATEMINIMUM. The initial default for updating new indexes is 1, meaning that incremental updates can be done after every change.
- UPDATEAUTOCOMMIT. The initial default for updating new indexes is 0, meaning that there will be no intermediate commits when documents are read from DB2 text columns. This value is reserved, and you cannot change it.

You cannot use **db2ts** commands to change the default values at the database level.

---

## SYSIBMTS.TSLOCKS view

You can view command lock information at the database and index level using SYSIBMTS.TSLOCKS.

Table 13. SYSIBMTS.TSLOCKS view

| Column name    | Data type    | Nullable? | Description                                                                                                                                          |
|----------------|--------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| COMMAND        | VARCHAR(30)  | NO        | Name of the command that created the lock. Possible values are: CREATE INDEX, ALTER INDEX, DROP INDEX, UPDATE INDEX, CLEAR EVENTS, DISABLE DATABASE. |
| LOCKSCOPE      | VARCHAR(30)  | NO        | Scope of the lock. Possible values are: DATABASE or INDEX.                                                                                           |
| INDSCHEMA      | VARCHAR(128) | YES       | Schema name of the text search index (only for LOCKSCOPE = INDEX)                                                                                    |
| INDNAME        | VARCHAR(128) | YES       | Unqualified name of the text search index (only for LOCKSCOPE = INDEX)                                                                               |
| PROCESSID      | INTEGER      | NO        | Identifier of the process that created the lock                                                                                                      |
| THREADID       | INTEGER      | NO        | Identifier of the thread that created the lock                                                                                                       |
| LOCKCREATETIME | TIMESTAMP    | NO        | Time stamp when the lock was granted                                                                                                                 |

You can use the process or thread information to check whether the process or thread still exists. If they no longer exist, remove the lock by running the **db2ts CLEAR COMMAND LOCKS** command.

---

## SYSIBMTS.TSCOLLECTIONNAMES view

The SYSIBMTS.TSCOLLECTIONNAMES view displays the names of collections.

Each row represents a collection for a text search index.

Table 14. SYSIBMTS.TSCOLLECTIONNAMES view

| Column name    | Data type    | Nullable? | Description                                                 |
|----------------|--------------|-----------|-------------------------------------------------------------|
| INDSCHEMA      | VARCHAR(128) | NO        | Schema name of the text search index                        |
| INDNAME        | VARCHAR(128) | NO        | Unqualified name of the text search index                   |
| COLLECTIONNAME | VARCHAR(132) | NO        | Name of the associated collection on the text search server |

---

## SYSIBMTS.TSINDEXES view

You can view the db2ts **CREATE INDEX** command parameters and generated settings using SYSIBMTS.TSINDEXES.

You can query the view to obtain information about the indexes. This example uses the index schema:

```
db2 "SELECT COLNAME from SYSIBMTS.TSINDEXES where INDSCHEMA=schema-name
and INDNAME=index-name"
```

Table 15. SYSIBMTS.TSINDEXES view

| Column name | Data type    | Nullable? | Description                            |
|-------------|--------------|-----------|----------------------------------------|
| INDSCHEMA   | VARCHAR(128) | NO        | Schema name for the text search index. |

Table 15. SYSIBMTS.TSINDEXES view (continued)

| Column name          | Data type      | Nullable? | Description                                                                                                                                                                                                                                        |
|----------------------|----------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| INDNAME              | VARCHAR(128)   | NO        | Unqualified name of the text search index.                                                                                                                                                                                                         |
| TABSHEMA             | VARCHAR(128)   | NO        | Schema name of the base table.                                                                                                                                                                                                                     |
| TABNAME              | VARCHAR(128)   | NO        | Unqualified name of the base table.                                                                                                                                                                                                                |
| COLNAME              | VARCHAR(128)   | NO        | Column that the text search index was created on.                                                                                                                                                                                                  |
| CODEPAGE             | INTEGER        | NO        | Document code page for the text search index.                                                                                                                                                                                                      |
| LANGUAGE             | VARCHAR(5)     | NO        | Document language for the text search index.                                                                                                                                                                                                       |
| FORMAT               | VARCHAR(30)    | YES       | Document format.                                                                                                                                                                                                                                   |
| FUNCTIONSCHEMA       | VARCHAR(128)   | YES       | Schema for the column type.                                                                                                                                                                                                                        |
| FUNCTIONNAME         | VARCHAR(18)    | YES       | Name of the column type conversion function.                                                                                                                                                                                                       |
| COLLECTIONDIRECTORY  | VARCHAR(512)   | YES       | Directory for the text search index files.                                                                                                                                                                                                         |
| UPDATEFREQUENCY      | VARCHAR(300)   | NO        | Trigger criterion for applying updates to the index.                                                                                                                                                                                               |
| UPDATEMINIMUM        | INTEGER        | YES       | Minimum number of entries in the log table before an incremental update is performed. A lower value means better consistency between the table column and the text search index. However, it also increases the text search indexing memory usage. |
| EVENTVIEWSCHEMA      | VARCHAR(128)   | NO        | Schema for the event view created for the text search index (always SYSIBMTS).                                                                                                                                                                     |
| EVENTVIEWNAME        | VARCHAR(128)   | NO        | Name of the event view created for the text search index.                                                                                                                                                                                          |
| STAGINGVIEWSCHEMA    | VARCHAR(128)   | YES       | Schema for the log view created for the text search index (always SYSIBMTS).                                                                                                                                                                       |
| STAGINGVIEWNAME      | VARCHAR(128)   | YES       | Name of the log view created for the text search index.                                                                                                                                                                                            |
| REORGAUTOMATIC       | INTEGER        | YES       | Reserved (not supported in this release). The value is always 1.                                                                                                                                                                                   |
| RECREATEONUPDATE     | INTEGER        | NO        | Reserved (not supported in this release). The value is always 0.                                                                                                                                                                                   |
| ATTRIBUTES           | VARCHAR(32000) | YES       | Reserved (not supported in this release).                                                                                                                                                                                                          |
| COLLECTIONNAMEPREFIX | VARCHAR(128)   | NO        | Prefix of the collection name on the text search server.                                                                                                                                                                                           |
| COMMENT              | VARCHAR(512)   | YES       | Comment specified in the index properties of the <b>CREATE INDEX</b> command.                                                                                                                                                                      |

## SYSIBMTS.TSCONFIGURATION view

Information about index configuration parameters is available in the SYSIBMTS.TSCONFIGURATION view.

Each row represents a configuration parameter of the text search index.

Following is an example of a query against the view that uses the index name:

```
db2 "SELECT VALUE from SYSIBMTS.TSCONFIGURATION where INDSHEMA=schema-name
and INDNAME=ind-name and PARAMETER ='parameter'"
```

Table 16. SYSIBMTS.TSCONFIGURATION view

| Column name | Data type    | Nullable? | Description                               |
|-------------|--------------|-----------|-------------------------------------------|
| INDSCHEMA   | VARCHAR(128) | NO        | Schema name of the text search index      |
| INDNAME     | VARCHAR(128) | NO        | Unqualified name of the text search index |
| PARAMETER   | VARCHAR(30)  | NO        | Name of a configuration parameter         |
| VALUE       | VARCHAR(512) | NO        | Value of the parameter                    |

The `PARAMETER` column contains the names of the text search index configuration parameters specified with the `CREATE INDEX` statement and the names of some of the parameters from the `SYSIBMTS.TSDEFAULTS` view.

## SYSIBMTS.TSEVENT view

The event view provides information about indexing status and error events.

A database might have multiple views with the prefix `SYSIBMTS.TSEVENT_`. Each view is differentiated by the `nnnnnn` value, an internal identifier that points to the corresponding text index that the view is associated with. To determine the text search index associated with a particular view, query the view `SYSIBMTS.TSINDEXES`, searching for the schema name and view name in the columns `EVENTVIEWSCHEMA` and `EVENTVIEWNAME`. The query returns a single row that describes the text search index and user table in question.

The number of columns in this view depends on the number of primary key columns in the user table. The columns `PK1..PKnn` are similar to the primary key columns of the user table. The data type of each of the columns in the view exactly corresponds to the data type of the corresponding primary key column.

Each row in this view represents a message from an **UPDATE INDEX** command on the text search index. For instance, a row might indicate that an **UPDATE INDEX** command has started or has completed. Alternatively, a row might describe a problem that occurred when a text document was being indexed. You can identify the text document by retrieving the primary key column values from the row in this view and looking them up in the user table.

**Note:** If a document is inserted and then the same document is deleted before the next index update, the event table will indicate that two documents were deleted.

You can clear events by using the db2ts **CLEAR EVENTS FOR INDEX** command.

Table 17. Event view

| Column name | Data type                                                   | Nullable? | Description                                                                                                                                                                                                                                                                                                                                                 |
|-------------|-------------------------------------------------------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OPERATION   | INTEGER                                                     | YES       | The operation (insert, update, or delete) on the base table to be reflected in the text search index                                                                                                                                                                                                                                                        |
| TIME        | TIMESTAMP                                                   | YES       | Time stamp of event entry creation                                                                                                                                                                                                                                                                                                                          |
| SEVERITY    | INTEGER                                                     | YES       | If the message corresponds to a single document, one of the following values: <ul style="list-style-type: none"> <li>• 1 = Informational</li> <li>• 4 = Parts of the document were indexed but there was a warning, as indicated by the message</li> <li>• 8 = The document was not indexed, as indicated by the message</li> <li>• 0= Otherwise</li> </ul> |
| SQLCODE     | INTEGER                                                     | YES       | SQLCODE for the associated error, if any                                                                                                                                                                                                                                                                                                                    |
| MESSAGE     | VARCHAR(1024)                                               | YES       | Text information about the specific error                                                                                                                                                                                                                                                                                                                   |
| PK01        | Data type of the first primary key column of the base table | YES       | Value of the first primary key column of the base table of the text search index for the row being processed when the event occurred                                                                                                                                                                                                                        |

Table 17. Event view (continued)

| Column name | Data type                                                  | Nullable? | Description                                                                                                                         |
|-------------|------------------------------------------------------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------|
| PK $nn$     | Data type of the last primary key column of the base table | YES       | Value of the last primary key column of the base table of the text search index for the row being processed when the event occurred |

Informational events, such as starting, committing, and finishing update processing are also available in this view. In this case, PK01, PK $nn$  and OPERATION all have NULL values. The code page and the locale of MESSAGE correspond to the database settings.

---

## SYSIBMTS.TSSTAGING view

The staging table stores the change operations on the user table that requires synchronization with the text search index.

Triggers are created on the user table to feed change information into the staging table. The **UPDATE INDEX** command reads the entries and deletes them after successful synchronization.

The database might have multiple views with the prefix SYSIBMTS.TSSTAGING\_. Each view is differentiated by the  $nnnnnn$  value, an internal identifier that points to the corresponding text index that the view is associated with. To determine the text search index associated with a particular view, query the view SYSIBMTS.TSINDEXES, searching for the schema name and view name in the columns STAGINGVIEWSHEMA and STAGINGVIEWNAME. The query returns a single row that describes the text search index and user table in question.

The number of columns in this view depends on the number of primary key columns in the user table. The columns PK1..PK $nn$  are similar to the primary key columns of the user table. The data type of each of the columns in the view exactly corresponds to the data type of the corresponding primary key column.

Each row in this view represents an insert, a delete, or an update operation on a user table row or text document. You can identify the text document by retrieving the primary key column values from the row in this view and looking them up in the user table.

You can use the following query to obtain information about the view:

```
db2 "SELECT STAGINGVIEWSHEMA, STAGINGVIEWNAME from SYSIBMTS.TSINDEXES
where INDSHEMA=schema-name and INDNAME=index-name"
```

Table 18. SYSIBMTS.TSSTAGING view

| Column Name | Data type                                                   | Nullable? | Description                                                                                                                                                    |
|-------------|-------------------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OPERATION   | INTEGER                                                     | NO        | The operation (insert, update, or delete) on the base table to be reflected on the text search index.                                                          |
| TIME        | TIMESTAMP                                                   | No        | Sequence ID of a row (when an insert, an update, or a delete trigger is fired). This is a time stamp but does not exactly represent the time of the operation. |
| PK01        | Data type of the first primary key column of the base table | YES       | First primary key column of the base table.                                                                                                                    |

Table 18. SYSIBMTS.TSSTAGING view (continued)

| Column Name     | Data type                                                  | Nullable? | Description                                |
|-----------------|------------------------------------------------------------|-----------|--------------------------------------------|
| PK <sub>n</sub> | Data type of the last primary key column of the base table | YES       | Last primary key column of the base table. |

---

## Chapter 11. Text Search tools

---

### Configuration Tool for DB2 Text Search

The Configuration Tool is a command-line tool that you can use for the initial text search configuration or the post-installation customization of the text search configuration for a DB2 instance.

Using the text search configuration set up during the installation of your DB2 database product is preferable to using the Configuration Tool to manually configure DB2 Text Search because using the tool requires you to configure all of the parameters, whereas using the installer requires you to configure only two settings. However, the Configuration Tool is the only means of making subsequent changes to a number of the settings, including the following ones:

- Various paths, such as the configuration path, the log path, the installation path, and the temporary directory path
- The log level
- The heap size used

#### Tool syntax

```
►►--configTool—mandatory_command_option—————►
►--configPath—absolute-path-to-config-folder—————►
►┌──┐individual_command_specific_arguments──►
 └──_locale—five-char-locale-string──┘
```

#### Parameters

##### Mandatory command parameters

###### **configureHTTPListener**

Allows you to configure an administration HTTP port to allow communication between text search processes using TCP/IP. Specify the following argument:

**-adminHTTPPort** *value*

Specifies the port to be used by DB2 Text Search

##### **Note:**

- There are other arguments for **configureHTTPListener** that are listed in the Usage section, but you cannot use them with DB2 Database for Linux, UNIX, and Windows.
- You must stop the text search instance services before configuring the administration HTTP port.

###### **upgradeInstance**

You can update DB2 Text Search server for your instances by issuing the **upgradeInstance** command. Specify the following argument:

**-installedConfigPath** *value*

Specifies the fully qualified path to the installed configuration directory. For example,

- On Linux and UNIX operating systems, this path might be
  - installedConfigPath *\$DB2DIR/cfg/db2tss/config*
  - configPath *\$INSTHOME/sql1lib/db2tss/config*

where *INSTHOME* is the instance home directory and *DB2DIR* is the location of the DB2 Version 9.7 copy.

- For Windows operating systems, this path might be
  - installedConfigPath "*%DB2PATH%\CFG\DB2TSS\CONFIG*"
  - configPath "*%INSTPROFDIR%\instance-name\DB2TSS\CONFIG*"

where *DB2PATH* is the location of the DB2 Version 9.7 copy and *INSTPROFDIR* is the instance profile directory.

### **configureParams**

Allows you to configure the various configurable parameters for DB2 Text Search. Specify one of the following arguments:

#### **-logPath** *path*

Specifies the log path to be used by DB2 Text Search.

#### **-installPath** *path*

Specifies the fully qualified path to the text search installation directory. For example, on UNIX platforms, this path might be *\$HOME/sql1lib/db2tss..*

#### **-tempDirPath** *path*

Specifies the temporary directory path to be used by DB2 Text Search.

#### **-maxHeapSize** *value*

Specifies the heap size. The default value is 1.5 Gigabytes.

#### **-logLevel** *value*

Specifies the logging level for messages in the log file for the text search engine. The settings (in order of severity) are as follows:

- FINEST
- FINER
- FINE
- WARNING
- SEVERE
- OFF

### **generateToken**

Generates the authentication token. The following argument is the only one that you can use:

#### **-seed** *value*

Acts as an arbitrary string for generating an authentication token. Given the same seed value, the Configuration Tool generates the same token every time.

### **printToken**

Prints the authentication token. This option does not require you to stop the instance services.

### **printAll**

Prints to standard output the values for all of the properties that are configurable. This option does not require you to stop the instance services.

**printAdminHTTPPort**

Prints the current value for the administration HTTP port. This command does not require you to stop the instance services.

**-configPath** *absolute-path-to-config-folder*

Specifies the absolute path to the text search configuration folder, such as `/opt/ibm/search/config`. Note that when you use **configureParams**, **configPath** is used not only to point to the configuration directory but also to modify the configuration to use the specified configuration path.

**-locale** *locale*

Specifies the five-character locale for writing any error messages that are returned during the use of the tool, such as `en_US`, `de_DE`, or `fr_FR`. The default is `en_US`. For details see Supported locales

**help**

Prints the Configuration Tool command usage. The usage information is also printed if you run the Configuration Tool without a required argument.

**Usage**

- Unless otherwise noted in a parameter description, you must stop the text search instance services to configure the parameter. For the configuration settings that require the instance services to be stopped, the Configuration Tool detects the related condition and returns a message indicating that the instance services are running and must be stopped.
- The Configuration Tool and the other command-line tools do not authenticate users. However, you must have access to the configuration files to run these tools. On Windows platforms, if the extended DB2 security model is implemented, only members of the DB2ADMNS group can run the command-line tools. If the extended DB2 security model is not implemented, the DB2 database administrator must control access to these tools.
- The Configuration Tool does not update the services file. If you use the Configuration Tool for first-time configuration or changes to an existing configuration, you must update the services file manually.

---

## Administration tool for DB2 Text Search

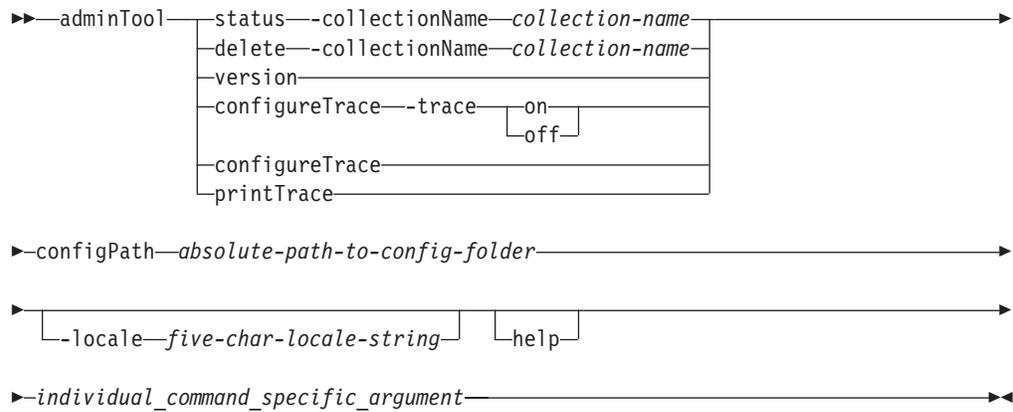
The administration tool is a command-line tool that provides a convenient way to perform several administrative tasks related to DB2 Text Search.

The tasks include the following ones:

- Checking the status and properties of text search collections
- Deleting orphaned text search collections
- Reporting the version of the server

The text search instance services must be running for the Administration Tool to remove text search collections. You must append the `.sh` suffix to all decoupled Text Search server tools on supported platforms unless mentioned otherwise, for example, `adminTool.sh`.

**Tool syntax**



## Parameters

### delete

Deletes the text search collection specified by **-collectionName**.

This option removes only orphaned collections. Ensure that the collection is not associated with a valid text index. Also do not delete a collection while documents are being indexed.

### status

Provides the status of a text search collection, including the index size, the number of documents present, and the progress of index updates.

**-collectionName** *collection-name*

Specifies the text search collection for which you want the status printed.

### version

Displays the version of the server in the following format:

```
version.release.modification.fixpack-protocol-build
```

where,

- `version.release.modification.fixpack`: the standard version numbering convention used by IBM products
- `protocol`: number indicating the protocol that is used to serialize request arguments and data between the server and client.
- `build`: number indicating the server build

For example,

```
1.0.0.0-1.1-290
```

indicates server version 1, protocol 1.1, build 290

### supportedClientVersions

Prints the list of supported client versions.

### configureTrace

Enables or disables tracing.

#### -trace

- `ON`: changes the logging level to `FINEST`
- `OFF`: changes the logging level to `INFO`

### printTrace

Prints the current trace setting.

**-collectionName** *collection-name*

Specifies the name of the collection that you want to delete or obtain status information about.

**-configPath** *absolute-path-to-config-folder*

Specifies the absolute path to the text search configuration folder.

**-locale** *five-char-locale-string*

Specifies the five character locale for writing any error messages that are returned during the use of the tool. If you do not specify this parameter, the server locale is used.

**help**

Prints the Administration Tool command usage. The usage information is also printed if you run the Administration Tool without a required argument.

## Examples

To delete a collection:

```
adminTool delete -configPath /opt/ibm/search/config
-collectionName db2inst1_SAMPLE_TS242320_0000
```

To print the status of all of the collections that use the French locale, issue the following command:

```
adminTool status -configPath c:\Programs\IBM\Search\config -locale fr_FR
```

To turn on tracing, issue the following command:

```
adminTool configureTrace -configPath /opt/ibm/search/config -trace on
```

## Usage notes

The error or trace log is, by default, written to AdminTool.0.log in the default log directory. You can format the log by using the Log Formatter tool.

Run the administration tool from the <ECMTS\_HOME>\bin directory on the ECM Text Search server.

- On Linux or UNIX systems: **adminTool.sh**
- On Windows systems: **adminTool.bat**

---

## Synonym Tool for DB2 Text Search

The Synonym Tool is a command-line tool that provides a convenient way to add synonym dictionaries to or remove synonym dictionaries from text search indexes.

You do not have to stop the text search instance services to use the Synonym Tool.

### synonymTool syntax

```
►► synonymTool { importSynonym-specs | removeSynonym --collectionName collection-name | help }
►► -configPath absolute-path-to-config-folder ►►
```

## importSynonym-specs:

```
|—importSynonym—synonymFile—absolute-path-to-syn-file—→
▶--collectionName—collection-name—replace—true—
|—false—|
```

## Parameters

### importSynonym

Adds the synonym dictionary specified by **-synonymFile** to the text search collection specified by **-collectionName**.

**-synonymFile** *absolute-path-to-syn-file*

Specifies the XML file that is to be used as a synonym dictionary.

### -replace

- true

Specifies that the synonyms in the file specified by **-synonymFile** are to replace the current synonyms of the existing synonym dictionary associated with the text search collection.

- false

Specifies that the synonyms in the file specified by **-synonymFile** are to be appended to the existing synonym dictionary associated with the text search collection.

### removeSynonym

Removes any synonym dictionary from the text search collection specified by **-collectionName**.

**-collectionName** *collection-name*

Specifies the text search collection.

**-configPath** *absolute-path-to-config-folder*

Specifies the absolute path to the text search configuration folder.

### help

Prints the Synonym Tool command usage. The usage information is also printed if you run the Synonym Tool without a required argument.

## Usage

The XML file that is to be used as a synonym dictionary must be on the text search server.

The error or trace log is, by default, written to `SynonymTool.0.log` in the default log directory.

---

## Log Formatter Tool for DB2 Text Search

DB2 Text Search logs system messages and trace messages to help you determine the source of problems that might occur. You can view and save the server logs using the Log Formatter Tool.

The server logs are located in the `INSTALL_HOME/log` directory. By default, the trace log is turned off, and the system log level is set to SEVERE. The server logs are

rotated, meaning that after a log file contains 10 MB of data, a new generation of the log file is created. At any given time, a maximum of five generations of log files are kept on the system.

## Tool syntax

```
▶▶ logformatter -f log-file [-l locale] [-o output-file] [-v] ▶▶
```

## Parameters

### **logformatter**

Enables server log viewing or saving.

#### **-f** *log-file*

Specifies the server log file that you want to format.

#### **-l** *locale*

Specifies the five-character locale for writing the reformatted log messages. The default is en\_US.

#### **-o** *output-file*

Specifies the output file where the reformatted log messages are to be written by using UTF-8 encoding. If you do not specify this option, a standard output file is used.

#### **-v**

Specifies that debugging messages are to be written in verbose mode.

#### **-?** Prints the help message.



## Appendix A. DB2 Text Search and Net Search Extender comparison

Even though queries performed using DB2 Text Search and Net Search Extender look similar, there can be some discrepancies between the query results. Therefore, it is important to be aware of the differences between the two search engines.

Use the following tables, which compare the features and functions of DB2 Text Search and Net Search Extender, to help you decide which search tool best suits your needs.

*Table 19. Administration and environments*

| Feature or function                  | DB2 Text Search                   | DB2 Net Search Extender                                                                    |
|--------------------------------------|-----------------------------------|--------------------------------------------------------------------------------------------|
| The command-line interface           | Supported                         | Supported                                                                                  |
| Stored procedures for administration | Supported                         | Unsupported                                                                                |
| External storage of text documents   | Unsupported                       | Supported                                                                                  |
| Partitioned database environments    | Unsupported                       | Supported on all platforms except Linux PPC, Solaris on AMD64 and Microsoft Cluster server |
| Error handling                       | SQL codes, product-specific codes | Product-specific codes                                                                     |
| Linguistic processing                | In all supported languages        | In English only                                                                            |

*Table 20. Searching*

| Feature or function                       | DB2 Text Search | DB2 Net Search Extender |
|-------------------------------------------|-----------------|-------------------------|
| XPath                                     | Supported       | Unsupported             |
| XQuery                                    | Supported       | Supported               |
| SQL/XML queries                           | Supported       | Supported               |
| SQL queries                               | Supported       | Supported               |
| SCORE function                            | Supported       | Supported               |
| CONTAINS and xmlcolumn-contains functions | Supported       | Supported               |
| NUMBEROFMATCHES scalar function           | Unsupported     | Supported               |
| Highlighting                              | Unsupported     | Supported               |
| Weight or boosting                        | Supported       | Supported               |
| RESULTLIMIT argument                      | Supported       | Supported               |
| Escape characters                         | Supported       | Supported               |
| Optional terms                            | Supported       | Unsupported             |
| Attribute search                          | Unsupported     | Supported               |

Table 20. Searching (continued)

| Feature or function                                                      | DB2 Text Search                                 | DB2 Net Search Extender                                     |
|--------------------------------------------------------------------------|-------------------------------------------------|-------------------------------------------------------------|
| Fielded search                                                           | For XML documents, using XML search expressions | For XML documents, by defining fields with a document model |
| Proximity search                                                         | Unsupported                                     | Supported                                                   |
| Fuzzy search                                                             | Unsupported                                     | Supported                                                   |
| Wildcard search                                                          | Supported                                       | Supported                                                   |
| Relationships in thesaurus (hierarchical, associative, and user-defined) | Unsupported                                     | Supported                                                   |
| Synonym dictionaries                                                     | Supported                                       | Supported                                                   |

Table 21. Indexing and data

| Feature or function                                                                                                  | DB2 Text Search                                   | DB2 Net Search Extender                                                         |
|----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|---------------------------------------------------------------------------------|
| Index reorganization                                                                                                 | Implicit reorganization supported                 | Implicit and explicit reorganization supported                                  |
| Index updates on triggers                                                                                            | Supported                                         | Supported                                                                       |
| Index updates using replication                                                                                      | Unsupported                                       | Supported                                                                       |
| Index re-creation on update                                                                                          | Unsupported                                       | Supported                                                                       |
| Index creation on view columns                                                                                       | Unsupported                                       | Supported                                                                       |
| Index creation on nicknames in federated databases                                                                   | Unsupported                                       | Supported                                                                       |
| Multiple indexes per column                                                                                          | Unsupported                                       | Supported                                                                       |
| Presorted indexes                                                                                                    | Unsupported                                       | Supported                                                                       |
| Caching of index information (subset of index information kept in memory to speed up retrieval for frequent queries) | Unsupported                                       | Supported                                                                       |
| Streaming of results                                                                                                 | Supported                                         | Unsupported                                                                     |
| Custom transformation functions                                                                                      | Supported                                         | Supported                                                                       |
| Use of model files to index XML documents                                                                            | Unsupported                                       | Supported                                                                       |
| XML namespaces                                                                                                       | Not directly supported (must use workaround)      | Limited support                                                                 |
| Stop word processing                                                                                                 | Unsupported                                       | Supported                                                                       |
| Document formats                                                                                                     | Text, HTML, XML, proprietary formats (using INSO) | Text, HTML, XML, General Purpose Format (GPP), proprietary formats (using INSO) |

**Note:** The scoring algorithms used by DB2 Text Search and Net Search Extender are different, thus the results when using the SCORE function are probably not the same. Even though score results are consistent among other documents in the same

search result, the score values for a particular document may be different between DB2 Text Search and Net Search Extender, therefore score limits used in a query may need to be adjusted.

---

## Migration to DB2 Text Search

Starting with DB2 Version 9.7 Fix Pack 3 and later fix packs, DB2 Text Search and Net Search Extender text indexes can coexist within the same database. A DB2 Text Search text index can now be created on the same column for which a Net Search Extender index already exists, however, only one text index can be active for the same column, and all text indexes used in a single query must be of the same type.

Administration commands can be executed for active and inactive text indexes. This allows to populate a DB2 Text Search index for a column while the existing Net Search Extender index is still actively used.

The query language used for DB2 Text Search differs syntactically and semantically from DB2 Net Search Extender. When a new DB2 Text Search index is ready for use, and the activation switched from the Net Search Extender index to the new index, the application must be switched correspondingly.

It is recommended to maintain the inactive Net Search Extender until you are satisfied that the new text index suits your needs. If yes, the inactive Net Search Extender index can be dropped, if no, it can be reactivated including the associated application switch.

To migrate your database follow the steps:

1. Start DB2 Text Search.
2. Enable the Net Search Extender database for DB2 Text Search.
3. Create and update a DB2 Text Search index on a column which has an existing Net Search Extender index.
4. Activate the new DB2 Text Search index using the ALTER INDEX command with the SET ACTIVE option and switch any associated applications.
5. Ensure the DB2 Text Search index is working successfully. Backup and then delete the Net Search Extender indexes.

If the coexistence feature is not available on your system, create a test database with production data, and use the test database to study DB2 Text Search and differences to Net Search Extender. To switch the text search solution in the production database, you must disable Net Search Extender, enable DB2 Text Search, and create and populate the DB2 Text Search indexes.



---

## Appendix B. Locales supported for DB2 Text Search

The following table lists the locales that DB2 Text Search supports for document processing.

*Table 22. Supported locales*

| Locale code | Language          | Territory                   |
|-------------|-------------------|-----------------------------|
| ar_AA       | Arabic            | Arabic countries or regions |
| cs_CZ       | Czech             | Czech Republic              |
| da_DK       | Danish            | Denmark                     |
| de_CH       | German            | Switzerland                 |
| de_DE       | German            | Germany                     |
| el_GR       | Greek             | Greece                      |
| en_AU       | English           | Australia                   |
| en_GB       | English           | United Kingdom              |
| en_US       | English           | United States               |
| es_ES       | Spanish           | Spain                       |
| fi_FI       | Finnish           | Finland                     |
| fr_CA       | French            | Canada                      |
| fr_FR       | French            | France                      |
| it_IT       | Italian           | Italy                       |
| ja_JP       | Japanese          | Japan                       |
| ko_KR       | Korean            | Korea, Republic of          |
| nb_NO       | Norwegian Bokmål  | Norway                      |
| nl_NL       | Dutch             | Netherlands                 |
| nn_NO       | Norwegian Nynorsk | Norway                      |
| pl_PL       | Polish            | Poland                      |
| pt_BR       | Portuguese        | Brazil                      |
| pt_PT       | Portuguese        | Portugal                    |
| ru_RU       | Russian           | Russia                      |
| sv_SE       | Swedish           | Sweden                      |
| zh_CN       | Chinese           | China                       |
| zh_TW       | Chinese           | Taiwan                      |



---

## Appendix C. DB2 commands

---

### db2iupgrade - Upgrade instance

Upgrades an instance to a DB2 copy of the current release from a DB2 copy of a previous release. The DB2 copy from where you are running the **db2iupgrade** command must support instance upgrade from the DB2 copy that you want to upgrade.

On Linux and UNIX operating systems, this command is located in the *DB2DIR/instance* directory, where *DB2DIR* represents the installation location where the new release of the DB2 database system is installed. This command does not support instance upgrade for a non-root installation.

On Windows operating systems, this command is located in the **DB2PATH\bin** directory, where **DB2PATH** is the location where the DB2 copy is installed. To move your instance profile from its current location to another location, use the **/p** option and specify the instance profile path. Otherwise, the instance profile will stay in its original location after the upgrade.

#### Authorization

Root user authority on Linux and UNIX operating systems or Local Administrator on Windows operating systems.

#### Command syntax

##### For Linux and UNIX operating systems

```
db2iupgrade [-d] [-k] [-a AuthType] [-u FencedID] InstName
```

##### For Windows operating systems

```
db2iupgrade InstName /u:username,password

/p:instance-profile-path /q /a:authType

/j-"TEXT_SEARCH" [,servicename] [,portnumber] /?
```

#### Command parameters

##### For Linux and UNIX operating systems

- d** Turns debug mode on. Use this option only when instructed by DB2 database support.
- k** Keeps the pre-upgrade instance type if it is supported in the DB2 copy

from where you are running the **db2iupgrade** command. If this parameter is not specified, the instance type is upgraded to the default instance type supported.

**-a** *AuthType*

Specifies the authentication type (SERVER, CLIENT or SERVER\_ENCRYPT) for the instance. The default is SERVER.

**-u** *FencedID*

Specifies the name of the user ID under which fenced user-defined functions and fenced stored procedures will run. This option is required when upgrading a DB2 client instance to a DB2 server instance.

*InstName*

Specifies the name of the instance.

### For Windows operating systems

*InstName*

Specifies the name of the instance.

**/u:***username,password*

Specifies the account name and password for the DB2 service. This option is required when upgrading a partitioned instance.

**/p:***instance-profile-path*

Specifies the new instance profile path for the upgraded instance.

**/q** Issues the **db2iupgrade** command in quiet mode.

**/a:***authType*

Specifies the authentication type (SERVER, CLIENT, or SERVER\_ENCRYPT) for the instance.

**/j "TEXT\_SEARCH"**

Configures the DB2 Text Search server using generated default values for service name and TCP/IP port number. This parameter cannot be used if the instance type is client.

**/j "TEXT\_SEARCH, servicename"**

Configures the DB2 Text Search server using the provided service name and an automatically generated port number. If the service name has a port number assigned in the services file, it uses the assigned port number.

**/j "TEXT\_SEARCH, servicename, portnumber"**

Configures the DB2 Text Search server using the provided service name and port number.

**/j "TEXT\_SEARCH, portnumber"**

Configures the DB2 Text Search server using a default service name and the provided port number. Valid port numbers must be within the 1024 - 65535 range.

**/?** Displays usage information for the **db2iupgrade** command.

### Usage notes

The **db2iupgrade** command calls the **db2ckupgrade** command with the **-not1** parameter, and specifies `update.log` as the log file for **db2ckupgrade**. The default log file created for **db2iupgrade** is `/tmp/db2ckupgrade.log.processID`. Verify that

local databases are ready for upgrade before upgrading the instance. The **-not1** parameter disables the check for type-1 indexes. The log file is created in the instance home directory for Linux and UNIX operating systems or in the current directory for Windows operating systems. The instance upgrade will not continue if the **db2ckupgrade** command returns any errors.

For partitioned database environments, you should run the **db2ckupgrade** command in all database partitions before you issue the **db2iupgrade** command. The **db2ckupgrade** command only returns errors for the database partition where you issue the **db2iupgrade** command. If you do not check whether all database partitions are ready for upgrade, subsequent database upgrades could fail even though the instance upgrade was successful. See **db2ckupgrade** for details.

The **db2iupgrade** command should be run once only, as root on instance owning node.

#### For Linux and UNIX operating systems

- If you use the **db2iupgrade** command to upgrade a DB2 instance from a previous version to the current version of a DB2 database system, the DB2 Global Profile Variables defined in an old DB2 database installation path will not be upgraded over to the new installation location. The DB2 Instance Profile Variables specific to the instance to be upgraded will be carried over after the instance is upgraded.
- If you are using the **su** command instead of the **login** command to become the root user, you must issue the **su** command with the **-** option to indicate that the process environment is to be set as if you had logged in to the system using the **login** command.
- You must not source the DB2 instance environment for the root user. Running the **db2iupgrade** command when you sourced the DB2 instance environment is not supported.
- On UNIX and Linux operating systems, if the IBM Tivoli Monitoring for Databases: DB2 Agent is installed and the DB2 instance is upgraded, the Monitoring Agent for DB2 instance is also created if the following are true:
  - The DB2 instance type is standalone, wse, or ese.
  - The DB2 instance is at version 9.5 (or higher).

In addition, ITMA must already be installed for the DB2 copy you are updating the instance for. Located in *DB2DIR/itma* directory, where *DB2DIR* represents the directory where the DB2 database product is installed.

- On AIX 6.1 (or higher), when running this command from a shared DB2 copy in a system workload partition (WPAR) global environment, this command must be run as the root user.

#### For Windows operating systems

- On Windows operating systems, if the IBM Tivoli Monitoring for Databases: DB2 Agent is installed and the DB2 copy instance is updated, the Monitoring Agent for DB2 instance is also created if the following are true:
  - The DB2 instance type is standalone, wse, or ese.
  - The default DB2 copy has the ITM agent component installed.
  - The DB2 instance is at version 9.5 (or higher).
  - There is no existing ITM for Databases product.

In addition, the following are also created after the creation of the Monitoring Agent for DB2 instance: the Monitoring Agent for DB2 instance files, the Windows service and the registry entries.

## db2icrt - Create instance

Creates DB2 instances.

On Linux and UNIX operating systems, this utility is located in the *DB2DIR/instance* directory, where *DB2DIR* represents the installation location where the current version of the DB2 database system is installed. On Windows operating systems, this utility is located under the **DB2PATH\bin** directory where **DB2PATH** is the location where the DB2 copy is installed.

The **db2icrt** command creates DB2 instances in the instance owner's home directory.

**Note:** This command is not available for a non-root installation of DB2 database products on Linux and UNIX operating systems.

### Authorization

Root user authority on Linux and UNIX operating systems or Local Administrator authority on Windows operating systems.

### Command syntax

#### For Linux and UNIX operating systems

```

▶▶ db2icrt [-h] [-d] [-a AuthType] [-p PortName]
 [-s InstType] [-u FencedID] InstName

```

#### For Windows operating systems

```

▶▶ db2icrt InstName [-s InstType] [-u UserName, Password]
 [-p InstProfPath] [-h HostName] [-r PortRange]
 [-j "TEXT_SEARCH [,servicename] [,portnumber]"] [-?]

```

### Command parameters

#### For Linux and UNIX operating systems

**-h | -?**  
Displays the usage information.

- d** Turns debug mode on. Use this option only when instructed by DB2 database support.
- a** *AuthType*  
Specifies the authentication type (SERVER, CLIENT or SERVER\_ENCRYPT) for the instance. The default is SERVER.
- p** *PortName*  
Specifies the port name or number used by the instance. This option does not apply to client instances.
- s** *InstType*  
Specifies the type of instance to create. Use the **-s** option only when you are creating an instance other than the default associated with the installed product from which you are running **db2icrt**. Valid values are:
  - client** Used to create an instance for a client. This is the default instance type for IBM Data Server Client, IBM Data Server Runtime Client, and DB2 Connect™ Personal Edition.
  - standalone**  
Used to create an instance for a database server with local clients. It is the default instance type for DB2 Personal Edition.
  - ese** Used to create an instance for a database server with local and remote clients with partitioned database environment support. This is the default instance type for DB2 Enterprise Server Edition or DB2 Advanced Enterprise Server Edition.
  - wse** Used to create an instance for a database server with local and remote clients. This is the default instance type for DB2 Workgroup Server Edition, DB2 Express Edition or DB2 Express-C, and DB2 Connect Enterprise Edition.

DB2 database products support their default instance types and the instance types lower than their default ones. For instance, DB2 Enterprise Server Edition supports the instance types of **ese**, **wse**, **standalone** and **client**.
- u** *Fenced ID*  
Specifies the name of the user ID under which fenced user-defined functions and fenced stored procedures will run. The **-u** option is required if you are not creating a client instance.

*InstName*

Specifies the name of the instance which is also the name of an existing user in the operating system. This has to be the last argument of the **db2icrt** command.

**For Windows operating systems**

*InstName*

Specifies the name of the instance.

**-s** *InstType*

Specifies the type of instance to create. Currently, there are four kinds of DB2 instance types. Valid values are:

- client** Used to create an instance for a client. This is the default instance type for IBM Data Server Client, IBM Data Server Runtime Client, and DB2 Connect Personal Edition.

**standalone**

Used to create an instance for a database server with local clients. It is the default instance type for DB2 Personal Edition.

**ese**

Used to create an instance for a database server with local and remote clients with partitioned database environment support. The *-s ese -u Username, Password*

options have to be used with **db2icrt** to create the ESE instance type and a partitioned database environment instance.

**wse**

Used to create an instance for a database server with local and remote clients. This is the default instance type for DB2 Workgroup Server Edition, DB2 Express Edition or DB2 Express-C, and DB2 Connect Enterprise Edition.

DB2 database products support their default instance types and the instance types lower than their default ones. For instance, DB2 Enterprise Server Edition supports the instance types of *ese*, *wse*, *standalone* and *client*.

**-u Username, Password**

Specifies the account name and password for the DB2 service. This option is required when creating a partitioned database instance.

**-p InstProfPath**

Specifies the instance profile path.

**-h HostName**

Overrides the default TCP/IP host name if there is more than one for the current machine. The TCP/IP host name is used when creating the default database partition (database partition 0). This option is only valid for partitioned database instances.

**-r PortRange**

Specifies a range of TCP/IP ports to be used by the partitioned database instance when running in MPP mode. For example, *-r 50000,50007*. The services file of the local machine will be updated with the following entries if this option is specified:

```
DB2_InstName baseport/tcp
DB2_InstName_END endpoint/tcp
```

**/j "TEXT\_SEARCH"**

Configures the DB2 Text Search server using generated default values for service name and TCP/IP port number. This parameter cannot be used if the instance type is *client*.

**/j "TEXT\_SEARCH, servicename"**

Configures the DB2 Text Search server using the provided service name and an automatically generated port number. If the service name has a port number assigned in the services file, it uses the assigned port number.

**/j "TEXT\_SEARCH, servicename, portnumber"**

Configures the DB2 Text Search server using the provided service name and port number.

**/j "TEXT\_SEARCH, portnumber"**

Configures the DB2 Text Search server using a default service name and the provided port number. Valid port numbers must be within the 1024 - 65535 range.

-? Displays usage information.

## Example

- On an AIX machine, to create an instance for the user ID `db2inst1`, issue the following command:

On a client machine:

```
DB2DIR/instance/db2icrt db2inst1
```

On a server machine:

```
DB2DIR/instance/db2icrt -u db2fenc1 db2inst1
```

where `db2fenc1` is the user ID under which fenced user-defined functions and fenced stored procedures will run.

## Usage notes

- The `instance_home/sqllib/db2tss/config` folder is created by **db2icrt** on Linux and UNIX operating systems. It is advised that you use a symbolic link to an area outside the `sqllib` directory.
- The **-s** option is intended for situations in which you want to create an instance that does not use the full functions of the system. For example, if you are using DB2 Enterprise Server Edition (ESE) on a UNIX operating system, but do not want partition capabilities, you could create a DB2 Workgroup Server Edition (WSE) instance, using the option **-s WSE**.
- To create a DB2 instance that supports Microsoft Cluster Server, first create an instance, then use the **db2mcs** command to migrate it to run in a MSCS instance.
- On Linux and UNIX operating systems, only one instance can be created under a user name. If you want to create an instance under a user name that already has an instance, you must drop the existing instance before creating the new one.
- When creating DB2 instances, consider the following restrictions:
  - If existing IDs are used to create DB2 instances, make sure that the IDs are not locked and do not have passwords expired.
- You can also use the **db2isetup** command to create and update DB2 instances using a graphical interface on all supported Linux and UNIX operating systems.
- On Linux and UNIX operating systems, if you are using the **su** command instead of the **login** command to become the root user, you must issue the **su** command with the **-** option to indicate that the process environment is to be set as if you had logged in to the system using the **login** command.
- On Linux and UNIX operating systems, you must not source the DB2 instance environment for the root user. Running **db2icrt** when you sourced the DB2 instance environment is not supported.
- On AIX 6.1 (or higher), when running this command from a shared DB2 copy in a system workload partition (WPAR) global environment, this command must be run as the root user.
- On Windows operating systems, if the IBM Tivoli Monitoring for Databases: DB2 Agent is installed and the DB2 instance is created, the Monitoring Agent for DB2 instance is also created if the following are true:
  - The DB2 instance type is `standalone`, `wse`, or `ese`.
  - The default DB2 copy has the ITM agent component installed.

- The DB2 instance is at version 9.5 (or higher).
- There is no existing ITM for Databases product.

In addition, the following are also created after the creation of the Monitoring Agent for DB2 instance: the Monitoring Agent for DB2 instance files, the Windows service and the registry entries.

## db2idrop - Remove instance

Removes a DB2 instance that was created by **db2icrt**.

You can only drop instances that are listed by **db2ilist** for the same DB2 copy where you are issuing **db2idrop** from.

On Linux and UNIX operating systems, this utility is located in the *DB2DIR/instance* directory, where *DB2DIR* represents the installation location where the current version of the DB2 database system is installed. On Windows operating systems, this utility is located under the **DB2PATH\bin** directory where **DB2PATH** is the location where the DB2 copy is installed.

**Note:** A non-root-installed DB2 instance, on Linux and UNIX operating systems, cannot be dropped using this command. The only option is to uninstall the non-root DB2 copy. See *Usage notes* below for more details.

### Authorization

Root user authority on Linux and UNIX operating systems or Local Administrator on Windows operating systems.

### Command syntax

#### For Linux and UNIX operating systems

```

▶▶ db2idrop [-d] [-f] [-h] [-?] InstName

```

#### For Windows operating systems

```

▶▶ db2idrop [-f] [-h] InstName

```

### Command parameters

#### For Linux and UNIX operating systems

*InstName*

Specifies the name of the instance.

**-d** Enters debug mode, for use by DB2 database support.

**-f** This parameter is deprecated.

Specifies the force applications flag. If this flag is specified all the applications using the instance will be forced to terminate.

**-h | -?** Displays the usage information.

### For Windows operating systems

*InstName*

Specifies the name of the instance.

**-f** Specifies the force applications flag. If this flag is specified all the applications using the instance will be forced to terminate.

**-h** Displays usage information.

### Example

If you created `db2inst1` on a Linux or UNIX operating system by issuing the following command:

```
/opt/IBM/db2/copy1/instance/db2icrt -u db2fenc1 db2inst1
```

To drop `db2inst1`, you must run the following command:

```
/opt/IBM/db2/copy1/instance/db2idrop db2inst1
```

### Usage notes

- Before an instance is dropped, ensure that the DB2 database manager has been stopped and that DB2 database applications accessing the instance are disconnected and terminated. DB2 databases associated with the instance can be backed up, and configuration data saved for future reference if needed.
- The **db2idrop** command does not remove any databases. Remove the databases first if they are no longer required. If the databases are not removed, they can always be catalogued under another DB2 copy of the same release and continued to be used.
- If you wish to save DB2 Text Search configurations and plan to reuse instance databases, you need to take the extra step of saving the config directory (on UNIX: `instance_home/sql1lib/db2tss/config` and on Windows: `instance_profile_path\instance_name\db2tss\config`) or config directory contents before issuing the **db2idrop** command. After the new instance is created, the config directory can be restored. However, restoring the config directory is only applicable if the new instance created is of the same release and fix pack level.
- A non-root-installed instance cannot be dropped on Linux and UNIX operating systems. To remove this DB2 instance, the only option available to the user is to uninstall the non-root copy of DB2 by running **db2\_deinstall -a**.
- On Linux and UNIX operating systems, if you are using the **su** command instead of the **login** command to become the root user, you must issue the **su** command with the **-** option to indicate that the process environment is to be set as if you had logged in to the system using the **login** command.
- On Linux and UNIX operating systems, you must not source the DB2 instance environment for the root user. Running **db2idrop** when you sourced the DB2 instance environment is not supported.
- On AIX 6.1 (or higher), when running this command from a shared DB2 copy in a system workload partition (WPAR) global environment, this command must be run as the root user.

- On Windows operating systems, if the IBM Tivoli Monitoring for Databases: DB2 Agent is installed and the DB2 instance is dropped, the Monitoring Agent for DB2 is stopped. All files, services and registry entries created for the Monitoring Agent for DB2 instance are also deleted.
- On Windows operating systems, if an instance is clustered with Microsoft Cluster Service (MSCS), then you can uncluster that instance by issuing the **db2mcs** or **db2iclus** command prior to dropping the instance.

---

## db2iupdt - Update instances

Updates an instance to run on a DB2 copy that has a new DB2 database product or feature installed, to run on a DB2 copy of the same version as the DB2 copy associated with the instance, or to upgrade the instance type to a higher level instance type.

The **db2iupdt** command can be issued against instances of the same version that are associated with the same or a different DB2 copy. In all cases, it will update the instance so that it runs against the code located in the same DB2 copy as where you issued the **db2iupdt** command. You should issue this command if:

- You install a fix pack and the automatic instance update fails.
- You install a new DB2 database product or feature to the DB2 copy associated to the DB2 instance.
- You want to update a DB2 instance from one DB2 copy to another DB2 copy of the same version of DB2 database product.

After a fix pack is installed on Linux and UNIX operating systems, the **db2iupdt** command is executed automatically.

To update an instance with **db2iupdt**, you must first stop all processes that are running for the instance.

### Authorization

Root user authority on UNIX and Linux operating systems or Local Administrator on Windows operating systems.

### Command syntax

#### For UNIX and Linux operating systems

```

▶▶ db2iupdt _h_ _d_ _k_ _D_ _s_ _a_ AuthType
_?

```

```

▶▶ _u_ FencedID _e_ InstName

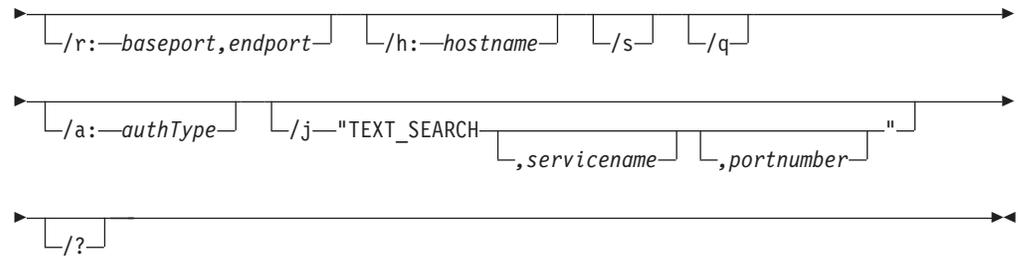
```

#### For Windows operating systems

```

▶▶ db2iupdt InstName /u:—username,password _p_—instance-profile-path

```



## Command parameters

### For UNIX and Linux operating systems

**-h | -?**

Displays the usage information.

**-d** Turns debug mode on.

**-k** Keeps the current instance type during the update.

**-D** Moves an instance from a higher code level on one path to a lower code level installed on another path.

**-s** Ignores the existing SPM log directory.

**-a** *AuthType*

Specifies the authentication type (SERVER, SERVER\_ENCRYPT or CLIENT) for the instance. The default is SERVER.

**-u** *Fenced ID*

Specifies the name of the user ID under which fenced user defined functions and fenced stored procedures will run. This option is only needed when converting an instance from a client instance to a non-client instance type. To determine the current instance type, refer to the node type parameter in the output from a **GET DBM CFG** command. If an instance is already a non-client instance, or if an instance is a client instance and is staying as a client instance (for example, by using the **-k** option), the **-u** option is not needed. The **-u** option can change the fenced user for an existing instance.

*InstName*

Specifies the name of the instance.

**-e** Updates every instance.

### For Windows operating systems

*InstName*

Specifies the name of the instance.

**/u:** *username,password*

Specifies the account name and password for the DB2 service.

**/p:** *instance-profile-path*

Specifies the new instance profile path for the updated instance.

**/r:** *baseport,endport*

Specifies the range of TCP/IP ports to be used by the partitioned database instance when running in MPP mode. When this option is specified, the services file on the local machine will be updated with the following entries:

```
DB2_InstName baseport/tcp
DB2_InstName_END endpoint/tcp
```

**/h:hostname**

Overrides the default TCP/IP host name if there are more than one TCP/IP host names for the current machine.

**/s** Updates the instance to a partitioned instance.

**/q** Issues the **db2iupdt** command in quiet mode.

**/a:authType**

Specifies *authType*, the authentication type (SERVER, CLIENT, or SERVER\_ENCRYPT) for the instance.

**/j "TEXT\_SEARCH"**

Configures the DB2 Text Search server using generated default values for service name and TCP/IP port number. This parameter cannot be used if the instance type is client.

**/j "TEXT\_SEARCH, servicename"**

Configures the DB2 Text Search server using the provided service name and an automatically generated port number. If the service name has a port number assigned in the services file, it uses the assigned port number.

**/j "TEXT\_SEARCH, servicename, portnumber"**

Configures the DB2 Text Search server using the provided service name and port number.

**/j "TEXT\_SEARCH, portnumber"**

Configures the DB2 Text Search server using a default service name and the provided port number. Valid port numbers must be within the 1024 - 65535 range.

**/?** Displays usage information for the **db2iupdt** command.

## Examples (UNIX and Linux)

- An instance, `db2inst2`, is associated to a DB2 copy of DB2 database product installed at `DB2DIR1`. You have another DB2 copy of DB2 database product on the same computer at `DB2DIR2` for the same version of the DB2 database product as that installed on `DB2DIR1`. To update the instance to run from the DB2 copy installed at `DB2DIR1` to the DB2 copy installed at `DB2DIR2`, issue the following command:

```
DB2DIR2/instance/db2iupdt db2inst2
```

If the DB2 copy installed at `DB2DIR2` is at level lower than the DB2 copy installed at `DB2DIR1`, issue:

```
DB2DIR2/instance/db2iupdt -D db2inst2
```

## Usage notes

### For all supported operating systems

- If you use the **db2iupdt** command to update a DB2 instance from one DB2 copy to another DB2 copy of the same version of DB2 database product, the DB2 Global Profile Variables defined in an old DB2 copy installation path will not be updated over to the new installation location. The DB2 Instance Profile Variables specific to the instance will be carried over after the instance is updated.

### For UNIX and Linux operating systems

- The **db2iupdt** command is located in the *DB2DIR/instance* directory, where *DB2DIR* is the location where the current version of the DB2 database product is installed.
- If you want to update a non-root instance, refer to the **db2nrupdt** non-root-installed instance update command. The **db2iupdt** does not support updating of non-root instances.
- If you are using the **su** command instead of the **login** command to become the root user, you must issue the **su** command with the **-** option to indicate that the process environment is to be set as if you had logged in to the system using the **login** command.
- You must not source the DB2 instance environment for the root user. Running **db2iupdt** when you sourced the DB2 instance environment is not supported.
- On UNIX and Linux operating systems, if the IBM Tivoli Monitoring for Databases: DB2 Agent is installed and the DB2 instance is updated, the Monitoring Agent for DB2 instance is also created if the following are true:
  - The DB2 instance type is standalone, wse, or ese.
  - The DB2 instance is at version 9.5 (or higher).

In addition, ITMA must already be installed for the DB2 copy you are updating the instance. Located in *DB2DIR/itma* directory, where *DB2DIR* represents the directory where the DB2 database product is installed.

- On AIX 6.1 (or higher), when running this command from a shared DB2 copy in a system workload partition (WPAR) global environment, this command must be run as the root user.

### For Windows operating systems

- The **db2iupdt** command is located in the **DB2PATH\bin** directory, where **DB2PATH** is the location where the current version of the DB2 database product is installed.
- The instance is updated to the DB2 copy from which you issue the **db2iupdt** command. However, to move your instance profile from its current location to another location, use the **/p** option and specify the instance profile path. Otherwise, the instance profile will stay in its original location after the instance update. Use the **db2iupgrade** command instead to upgrade to the current release from a previous release.
- On Windows operating systems, if the IBM Tivoli Monitoring for Databases: DB2 Agent is installed and the DB2 copy instance is updated, the Monitoring Agent for DB2 instance is also created if the following are true:
  - The DB2 instance type is standalone, wse, or ese.
  - The default DB2 copy has the ITM agent component installed.
  - The DB2 instance is at version 9.5 (or higher).
  - There is no existing ITM for Databases product.

In addition, the following are also created after the creation of the Monitoring Agent for DB2 instance: the Monitoring Agent for DB2 instance files, the Windows service and the registry entries.



---

## Appendix D. Overview of the DB2 technical information

DB2 technical information is available through the following tools and methods:

- DB2 Information Center
  - Topics (Task, concept and reference topics)
  - Help for DB2 tools
  - Sample programs
  - Tutorials
- DB2 books
  - PDF files (downloadable)
  - PDF files (from the DB2 PDF DVD)
  - printed books
- Command line help
  - Command help
  - Message help

**Note:** The DB2 Information Center topics are updated more frequently than either the PDF or the hardcopy books. To get the most current information, install the documentation updates as they become available, or refer to the DB2 Information Center at [ibm.com](http://ibm.com).

You can access additional DB2 technical information such as technotes, white papers, and IBM Redbooks® publications online at [ibm.com](http://ibm.com). Access the DB2 Information Management software library site at <http://www.ibm.com/software/data/sw-library/>.

### Documentation feedback

We value your feedback on the DB2 documentation. If you have suggestions for how to improve the DB2 documentation, send an email to [db2docs@ca.ibm.com](mailto:db2docs@ca.ibm.com). The DB2 documentation team reads all of your feedback, but cannot respond to you directly. Provide specific examples wherever possible so that we can better understand your concerns. If you are providing feedback on a specific topic or help file, include the topic title and URL.

Do not use this e-mail address to contact DB2 Customer Support. If you have a DB2 technical issue that the documentation does not resolve, contact your local IBM service center for assistance.

---

## DB2 technical library in hardcopy or PDF format

The following tables describe the DB2 library available from the IBM Publications Center at [www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss](http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss). English Version 9.7 manuals in PDF format can be downloaded from [www.ibm.com/support/docview.wss?uid=swg27015148](http://www.ibm.com/support/docview.wss?uid=swg27015148) and translated DB2 manuals in PDF format can be downloaded from [www.ibm.com/support/docview.wss?uid=swg27015149](http://www.ibm.com/support/docview.wss?uid=swg27015149).

Although the tables identify books available in print, the books might not be available in your country or region.

The form number increases each time a manual is updated. Ensure that you are reading the most recent version of the manuals, as listed below.

**Note:** The *DB2 Information Center* is updated more frequently than either the PDF or the hard-copy books.

*Table 23. DB2 technical information*

| <b>Name</b>                                                         | <b>Form Number</b> | <b>Available in print</b> | <b>Last updated</b> |
|---------------------------------------------------------------------|--------------------|---------------------------|---------------------|
| <i>Administrative API Reference</i>                                 | SC27-2435-03       | Yes                       | July, 2012          |
| <i>Administrative Routines and Views</i>                            | SC27-2436-03       | No                        | July, 2012          |
| <i>Call Level Interface Guide and Reference, Volume 1</i>           | SC27-2437-03       | Yes                       | July, 2012          |
| <i>Call Level Interface Guide and Reference, Volume 2</i>           | SC27-2438-03       | Yes                       | July, 2012          |
| <i>Command Reference</i>                                            | SC27-2439-03       | Yes                       | July, 2012          |
| <i>Data Movement Utilities Guide and Reference</i>                  | SC27-2440-01       | Yes                       | July, 2012          |
| <i>Data Recovery and High Availability Guide and Reference</i>      | SC27-2441-03       | Yes                       | July, 2012          |
| <i>Database Administration Concepts and Configuration Reference</i> | SC27-2442-03       | Yes                       | July, 2012          |
| <i>Database Monitoring Guide and Reference</i>                      | SC27-2458-03       | Yes                       | July, 2012          |
| <i>Database Security Guide</i>                                      | SC27-2443-02       | Yes                       | July, 2012          |
| <i>DB2 Text Search Guide</i>                                        | SC27-2459-03       | Yes                       | July, 2012          |
| <i>Developing ADO.NET and OLE DB Applications</i>                   | SC27-2444-02       | Yes                       | July, 2012          |
| <i>Developing Embedded SQL Applications</i>                         | SC27-2445-02       | Yes                       | July, 2012          |
| <i>Developing Java Applications</i>                                 | SC27-2446-03       | Yes                       | July, 2012          |
| <i>Developing Perl, PHP, Python, and Ruby on Rails Applications</i> | SC27-2447-02       | No                        | July, 2012          |
| <i>Developing User-defined Routines (SQL and External)</i>          | SC27-2448-02       | Yes                       | July, 2012          |
| <i>Getting Started with Database Application Development</i>        | GI11-9410-02       | Yes                       | July, 2012          |

Table 23. DB2 technical information (continued)

| <b>Name</b>                                                                             | <b>Form Number</b> | <b>Available in print</b> | <b>Last updated</b> |
|-----------------------------------------------------------------------------------------|--------------------|---------------------------|---------------------|
| <i>Getting Started with DB2 Installation and Administration on Linux and Windows</i>    | GI11-9411-00       | Yes                       | August, 2009        |
| <i>Globalization Guide</i>                                                              | SC27-2449-00       | Yes                       | August, 2009        |
| <i>Installing DB2 Servers</i>                                                           | GC27-2455-03       | Yes                       | July, 2012          |
| <i>Installing IBM Data Server Clients</i>                                               | GC27-2454-02       | No                        | July, 2012          |
| <i>Message Reference Volume 1</i>                                                       | SC27-2450-01       | No                        | August, 2009        |
| <i>Message Reference Volume 2</i>                                                       | SC27-2451-01       | No                        | August, 2009        |
| <i>Net Search Extender Administration and User's Guide</i>                              | SC27-2469-02       | No                        | September, 2010     |
| <i>Partitioning and Clustering Guide</i>                                                | SC27-2453-02       | Yes                       | July, 2012          |
| <i>pureXML Guide</i>                                                                    | SC27-2465-02       | Yes                       | July, 2012          |
| <i>Query Patroller Administration and User's Guide</i>                                  | SC27-2467-00       | No                        | August, 2009        |
| <i>Spatial Extender and Geodetic Data Management Feature User's Guide and Reference</i> | SC27-2468-02       | No                        | July, 2012          |
| <i>SQL Procedural Languages: Application Enablement and Support</i>                     | SC27-2470-03       | Yes                       | July, 2012          |
| <i>SQL Reference, Volume 1</i>                                                          | SC27-2456-03       | Yes                       | July, 2012          |
| <i>SQL Reference, Volume 2</i>                                                          | SC27-2457-03       | Yes                       | July, 2012          |
| <i>Troubleshooting and Tuning Database Performance</i>                                  | SC27-2461-03       | Yes                       | July, 2012          |
| <i>Upgrading to DB2 Version 9.7</i>                                                     | SC27-2452-03       | Yes                       | July, 2012          |
| <i>Visual Explain Tutorial</i>                                                          | SC27-2462-00       | No                        | August, 2009        |
| <i>What's New for DB2 Version 9.7</i>                                                   | SC27-2463-03       | Yes                       | July, 2012          |
| <i>Workload Manager Guide and Reference</i>                                             | SC27-2464-03       | Yes                       | July, 2012          |
| <i>XQuery Reference</i>                                                                 | SC27-2466-01       | No                        | November, 2009      |

Table 24. DB2 Connect-specific technical information

| Name                                                           | Form Number  | Available in print | Last updated |
|----------------------------------------------------------------|--------------|--------------------|--------------|
| <i>Installing and Configuring DB2 Connect Personal Edition</i> | SC27-2432-03 | Yes                | July, 2012   |
| <i>Installing and Configuring DB2 Connect Servers</i>          | SC27-2433-03 | Yes                | July, 2012   |
| <i>DB2 Connect User's Guide</i>                                | SC27-2434-02 | Yes                | July, 2012   |

Table 25. Information Integration technical information

| Name                                                                                          | Form Number  | Available in print | Last updated |
|-----------------------------------------------------------------------------------------------|--------------|--------------------|--------------|
| <i>Information Integration: Administration Guide for Federated Systems</i>                    | SC19-1020-02 | Yes                | August, 2009 |
| <i>Information Integration: ASNCLP Program Reference for Replication and Event Publishing</i> | SC19-1018-04 | Yes                | August, 2009 |
| <i>Information Integration: Configuration Guide for Federated Data Sources</i>                | SC19-1034-02 | No                 | August, 2009 |
| <i>Information Integration: SQL Replication Guide and Reference</i>                           | SC19-1030-02 | Yes                | August, 2009 |
| <i>Information Integration: Introduction to Replication and Event Publishing</i>              | GC19-1028-02 | Yes                | August, 2009 |

## Ordering printed DB2 books

### About this task

If you require printed DB2 books, you can buy them online in many but not all countries or regions. You can always order printed DB2 books from your local IBM representative. Keep in mind that some softcopy books on the *DB2 PDF Documentation DVD* are unavailable in print. For example, neither volume of the *DB2 Message Reference* is available as a printed book.

Printed versions of many of the DB2 books available on the DB2 PDF Documentation DVD can be ordered for a fee from IBM. Depending on where you are placing your order from, you may be able to order books online, from the IBM Publications Center. If online ordering is not available in your country or region, you can always order printed DB2 books from your local IBM representative. Note that not all books on the DB2 PDF Documentation DVD are available in print.

**Note:** The most up-to-date and complete DB2 documentation is maintained in the DB2 Information Center at <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7>.

To order printed DB2 books:

### Procedure

- To find out whether you can order printed DB2 books online in your country or region, check the IBM Publications Center at <http://www.ibm.com/shop/publications/order>. You must select a country, region, or language to access publication ordering information and then follow the ordering instructions for your location.
- To order printed DB2 books from your local IBM representative:
  1. Locate the contact information for your local representative from one of the following websites:
    - The IBM directory of world wide contacts at [www.ibm.com/planetwide](http://www.ibm.com/planetwide)
    - The IBM Publications website at <http://www.ibm.com/shop/publications/order>. You will need to select your country, region, or language to the access appropriate publications home page for your location. From this page, follow the "About this site" link.
  2. When you call, specify that you want to order a DB2 publication.
  3. Provide your representative with the titles and form numbers of the books that you want to order. For titles and form numbers, see “DB2 technical library in hardcopy or PDF format” on page 189.

---

## Displaying SQL state help from the command line processor

DB2 products return an SQLSTATE value for conditions that can be the result of an SQL statement. SQLSTATE help explains the meanings of SQL states and SQL state class codes.

### Procedure

To start SQL state help, open the command line processor and enter:

```
? sqlstate or ? class code
```

where *sqlstate* represents a valid five-digit SQL state and *class code* represents the first two digits of the SQL state.

For example, ? 08003 displays help for the 08003 SQL state, and ? 08 displays help for the 08 class code.

---

## Accessing different versions of the DB2 Information Center

### About this task

For DB2 Version 9.8 topics, the *DB2 Information Center* URL is <http://publib.boulder.ibm.com/infocenter/db2luw/v9r8/>.

For DB2 Version 9.7 topics, the *DB2 Information Center* URL is <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/>.

For DB2 Version 9.5 topics, the *DB2 Information Center* URL is <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/>.

For DB2 Version 9.1 topics, the *DB2 Information Center* URL is <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>.

For DB2 Version 8 topics, go to the *DB2 Information Center* URL at:  
<http://publib.boulder.ibm.com/infocenter/db2luw/v8/>.

---

## Displaying topics in your preferred language in the DB2 Information Center

### About this task

The DB2 Information Center attempts to display topics in the language specified in your browser preferences. If a topic has not been translated into your preferred language, the DB2 Information Center displays the topic in English.

### Procedure

- To display topics in your preferred language in the Internet Explorer browser:
  1. In Internet Explorer, click the **Tools** —> **Internet Options** —> **Languages...** button. The Language Preferences window opens.
  2. Ensure your preferred language is specified as the first entry in the list of languages.
    - To add a new language to the list, click the **Add...** button.

**Note:** Adding a language does not guarantee that the computer has the fonts required to display the topics in the preferred language.
    - To move a language to the top of the list, select the language and click the **Move Up** button until the language is first in the list of languages.
  3. Refresh the page to display the DB2 Information Center in your preferred language.
- To display topics in your preferred language in a Firefox or Mozilla browser:
  1. Select the button in the **Languages** section of the **Tools** —> **Options** —> **Advanced** dialog. The Languages panel is displayed in the Preferences window.
  2. Ensure your preferred language is specified as the first entry in the list of languages.
    - To add a new language to the list, click the **Add...** button to select a language from the Add Languages window.
    - To move a language to the top of the list, select the language and click the **Move Up** button until the language is first in the list of languages.
  3. Refresh the page to display the DB2 Information Center in your preferred language.

### Results

On some browser and operating system combinations, you must also change the regional settings of your operating system to the locale and language of your choice.

---

## Updating the DB2 Information Center installed on your computer or intranet server

A locally installed DB2 Information Center must be updated periodically.

## Before you begin

A DB2 Version 9.7 Information Center must already be installed. For details, see the “Installing the DB2 Information Center using the DB2 Setup wizard” topic in *Installing DB2 Servers*. All prerequisites and restrictions that applied to installing the Information Center also apply to updating the Information Center.

## About this task

An existing DB2 Information Center can be updated automatically or manually:

- Automatic updates - updates existing Information Center features and languages. An additional benefit of automatic updates is that the Information Center is unavailable for a minimal period of time during the update. In addition, automatic updates can be set to run as part of other batch jobs that run periodically.
- Manual updates - should be used when you want to add features or languages during the update process. For example, a local Information Center was originally installed with both English and French languages, and now you want to also install the German language; a manual update will install German, as well as, update the existing Information Center features and languages. However, a manual update requires you to manually stop, update, and restart the Information Center. The Information Center is unavailable during the entire update process.

This topic details the process for automatic updates. For manual update instructions, see the “Manually updating the DB2 Information Center installed on your computer or intranet server” topic.

## Procedure

To automatically update the DB2 Information Center installed on your computer or intranet server:

1. On Linux operating systems,
  - a. Navigate to the path where the Information Center is installed. By default, the DB2 Information Center is installed in the `/opt/ibm/db2ic/V9.7` directory.
  - b. Navigate from the installation directory to the `doc/bin` directory.
  - c. Run the `update-ic` script:  
`update-ic`
2. On Windows operating systems,
  - a. Open a command window.
  - b. Navigate to the path where the Information Center is installed. By default, the DB2 Information Center is installed in the `<Program Files>\IBM\DB2 Information Center\Version 9.7` directory, where `<Program Files>` represents the location of the Program Files directory.
  - c. Navigate from the installation directory to the `doc\bin` directory.
  - d. Run the `update-ic.bat` file:  
`update-ic.bat`

## Results

The DB2 Information Center restarts automatically. If updates were available, the Information Center displays the new and updated topics. If Information Center

updates were not available, a message is added to the log. The log file is located in `doc\eclipse\configuration` directory. The log file name is a randomly generated number. For example, `1239053440785.log`.

---

## Manually updating the DB2 Information Center installed on your computer or intranet server

If you have installed the DB2 Information Center locally, you can obtain and install documentation updates from IBM.

### About this task

Updating your locally-installed *DB2 Information Center* manually requires that you:

1. Stop the *DB2 Information Center* on your computer, and restart the Information Center in stand-alone mode. Running the Information Center in stand-alone mode prevents other users on your network from accessing the Information Center, and allows you to apply updates. The Workstation version of the DB2 Information Center always runs in stand-alone mode. .
2. Use the Update feature to see what updates are available. If there are updates that you must install, you can use the Update feature to obtain and install them

**Note:** If your environment requires installing the *DB2 Information Center* updates on a machine that is not connected to the internet, mirror the update site to a local file system using a machine that is connected to the internet and has the *DB2 Information Center* installed. If many users on your network will be installing the documentation updates, you can reduce the time required for individuals to perform the updates by also mirroring the update site locally and creating a proxy for the update site.

If update packages are available, use the Update feature to get the packages. However, the Update feature is only available in stand-alone mode.

3. Stop the stand-alone Information Center, and restart the *DB2 Information Center* on your computer.

**Note:** On Windows 2008, Windows Vista (and higher), the commands listed later in this section must be run as an administrator. To open a command prompt or graphical tool with full administrator privileges, right-click the shortcut and then select **Run as administrator**.

### Procedure

To update the *DB2 Information Center* installed on your computer or intranet server:

1. Stop the *DB2 Information Center*.
  - On Windows, click **Start > Control Panel > Administrative Tools > Services**. Then right-click **DB2 Information Center** service and select **Stop**.
  - On Linux, enter the following command:  
`/etc/init.d/db2icdv97 stop`
2. Start the Information Center in stand-alone mode.
  - On Windows:
    - a. Open a command window.
    - b. Navigate to the path where the Information Center is installed. By default, the *DB2 Information Center* is installed in the

*Program\_Files\IBM\DB2 Information Center\Version 9.7* directory, where *Program\_Files* represents the location of the Program Files directory.

c. Navigate from the installation directory to the `doc\bin` directory.

d. Run the `help_start.bat` file:

```
help_start.bat
```

• On Linux:

a. Navigate to the path where the Information Center is installed. By default, the *DB2 Information Center* is installed in the `/opt/ibm/db2ic/V9.7` directory.

b. Navigate from the installation directory to the `doc/bin` directory.

c. Run the `help_start` script:

```
help_start
```

The systems default Web browser opens to display the stand-alone Information Center.

3. Click the **Update** button (🔄). (JavaScript must be enabled in your browser.)

On the right panel of the Information Center, click **Find Updates**. A list of updates for existing documentation displays.

4. To initiate the installation process, check the selections you want to install, then click **Install Updates**.

5. After the installation process has completed, click **Finish**.

6. Stop the stand-alone Information Center:

• On Windows, navigate to the installation directory's `doc\bin` directory, and run the `help_end.bat` file:

```
help_end.bat
```

**Note:** The `help_end` batch file contains the commands required to safely stop the processes that were started with the `help_start` batch file. Do not use `Ctrl-C` or any other method to stop `help_start.bat`.

• On Linux, navigate to the installation directory's `doc/bin` directory, and run the `help_end` script:

```
help_end
```

**Note:** The `help_end` script contains the commands required to safely stop the processes that were started with the `help_start` script. Do not use any other method to stop the `help_start` script.

7. Restart the *DB2 Information Center*.

• On Windows, click **Start > Control Panel > Administrative Tools > Services**. Then right-click **DB2 Information Center** service and select **Start**.

• On Linux, enter the following command:

```
/etc/init.d/db2icdv97 start
```

## Results

The updated *DB2 Information Center* displays the new and updated topics.

---

## DB2 tutorials

The DB2 tutorials help you learn about various aspects of DB2 products. Lessons provide step-by-step instructions.

## Before you begin

You can view the XHTML version of the tutorial from the Information Center at <http://publib.boulder.ibm.com/infocenter/db2help/>.

Some lessons use sample data or code. See the tutorial for a description of any prerequisites for its specific tasks.

## DB2 tutorials

To view the tutorial, click the title.

### “pureXML” in *pureXML Guide*

Set up a DB2 database to store XML data and to perform basic operations with the native XML data store.

### “Visual Explain” in *Visual Explain Tutorial*

Analyze, optimize, and tune SQL statements for better performance using Visual Explain.

---

## DB2 troubleshooting information

A wide variety of troubleshooting and problem determination information is available to assist you with using DB2 database products.

### DB2 documentation

Troubleshooting information can be found in the *Troubleshooting and Tuning Database Performance* or the Database fundamentals section of the *DB2 Information Center*. The troubleshooting information contains topics that can help you isolate and identify problems with DB2 diagnostic tools and utilities. There are also solutions to some of the most common problems and advice on how to solve problems you might encounter with your DB2 database products.

### IBM Support Portal

See the IBM Support Portal if you are experiencing problems and want help finding possible causes and solutions. The Technical Support site has links to the latest DB2 publications, TechNotes, Authorized Program Analysis Reports (APARs or bug fixes), fix packs, and other resources. You can search through this knowledge base to find possible solutions to your problems.

Access the IBM Support Portal at [http://www.ibm.com/support/entry/portal/Overview/Software/Information\\_Management/DB2\\_for\\_Linux,\\_UNIX\\_and\\_Windows](http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/DB2_for_Linux,_UNIX_and_Windows).

---

## Terms and Conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

**Personal use:** You may reproduce these Publications for your personal, non commercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these Publications, or any portion thereof, without the express consent of IBM.

**Commercial use:** You may reproduce, distribute and display these Publications solely within your enterprise provided that all proprietary notices are preserved.

You may not make derivative works of these Publications, or reproduce, distribute or display these Publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the Publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.



---

## Appendix E. Notices

This information was developed for products and services offered in the U.S.A. Information about non-IBM products is based on information available at the time of first publication of this document and is subject to change.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information about the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
1623-14, Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 Japan

**The following paragraph does not apply to the United Kingdom or any other country/region where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements, changes, or both in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those

websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licenses of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information that has been exchanged, should contact:

IBM Canada Limited  
U59/3600  
3600 Steeles Avenue East  
Markham, Ontario L3R 9Z7  
CANADA

Such information may be available, subject to appropriate terms and conditions, including, in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems, and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious, and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating

platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (*your company name*) (*year*). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. *\_enter the year or years\_*. All rights reserved.

## Trademarks

IBM, the IBM logo, and [ibm.com](http://ibm.com)<sup>®</sup> are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

The following terms are trademarks or registered trademarks of other companies

- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle, its affiliates, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.



---

# Index

## A

Accessories Suite  
installing 50, 52  
ALTER INDEX Text Search command 114  
arguments 73

## B

books  
ordering 192

## C

capacity planning 12  
cataloging  
TCP/IP nodes 44  
CLEANUP FOR TEXT Text Search command 101  
CLEAR COMMAND LOCKS Text Search command 105  
CLEAR EVENTS FOR INDEX Text Search command 121  
commands  
ALTER INDEX 65  
calling from procedure 127  
CLEAR EVENTS FOR INDEX 65  
db2icrt 178  
db2idrop  
details 182  
db2iupdt 184  
db2iupgrade  
details 175  
db2ts  
summary 99  
db2ts ALTER INDEX 114  
db2ts CLEANUP FOR TEXT 101  
db2ts CLEAR COMMAND LOCKS 105  
db2ts CLEAR EVENTS FOR INDEX 121  
db2ts CREATE INDEX 107  
db2ts DISABLE DATABASE FOR TEXT 103  
db2ts DROP INDEX 112  
db2ts ENABLE DATABASE FOR TEXT 102  
db2ts HELP 123  
db2ts START FOR TEXT 100  
db2ts STOP FOR TEXT 100  
db2ts UPDATE INDEX 118  
DISABLE DATABASE FOR TEXT 56  
DROP INDEX 67  
ENABLE DATABASE FOR TEXT 49  
text search 99  
UPDATE INDEX 64  
configuration  
DB2 Text Search 54  
configuration tuning 13  
configuring 55  
CONTAINS function  
text search 89  
CREATE INDEX Text Search command 107  
create instance command 178

## D

DB2 Information Center  
languages 194  
updating 195, 196  
versions 193  
DB2 servers  
installing  
Windows 29  
DB2 Setup wizard  
installing  
DB2 servers (Linux) 32  
DB2 servers (UNIX) 32  
installing DB2 servers  
Linux 32  
UNIX 32  
installing DB2 servers on UNIX 32  
DB2 Text Search 12, 13, 14, 15, 16, 17, 18, 20, 21, 22, 73, 74, 75  
adding a synonym dictionary 69, 165  
administration commands 99  
administration tool 163  
administrative routines 127  
administrative views  
database level 155, 156  
event table 158  
index level 155, 156, 157, 158, 159  
log table 159  
staging table 159  
SYSIBMTS.TSCOLLECTIONNAMES 156  
SYSIBMTS.TSCONFIGURATION 157  
SYSIBMTS.TSDEFAULTS 155  
SYSIBMTS.TSEVENT 158  
SYSIBMTS.TSINDEXES 156  
SYSIBMTS.TSLOCKS 156  
SYSIBMTS.TSSTAGING 159  
ALTER INDEX command 114  
altering indexes 65  
asynchronous indexing 17  
authorizations  
database administrator 11  
instance owner 10  
roles 10  
user performing text search queries 11  
user table owner 11  
capacity planning  
TCP/IP port requirements 21  
changing update characteristics 65  
CLEAR COMMAND LOCKS command 105  
CLEAR EVENTS FOR INDEX command 121  
code pages supported 23  
command-line tools 8  
commands  
ALTER INDEX 114  
CLEANUP FOR TEXT 101  
CLEAR COMMAND LOCKS 105  
CLEAR EVENTS FOR INDEX 121  
CREATE INDEX 107  
DISABLE DATABASE FOR TEXT 103  
DROP INDEX 112  
ENABLE DATABASE FOR TEXT 102  
HELP 123  
START FOR TEXT 100

- DB2 Text Search *(continued)*
  - commands *(continued)*
    - STOP FOR TEXT 100
    - UPDATE INDEX 118
  - Configuration Tool 42, 161
  - configuring
    - Configuration Tool 42, 161
    - methods 41
    - overview 27
    - response file 35
    - rich text document support 54
    - Setup Wizard 28
    - XML searches 82
  - CONTAINS function 71, 89
  - CREATE INDEX command 107
  - data types
    - converting unsupported 23
    - supported 23
  - DISABLE DATABASE FOR TEXT command 103
  - disabling databases for 56
  - disabling rich text support 59
  - disk consumption 18
  - document formats
    - converting unsupported 23
    - supported 23
  - document truncation 24
  - DROP INDEX command 112
  - dropping indexes 67
  - ENABLE DATABASE FOR TEXT command 102
  - enabling databases for 49
  - enhancing performance during indexing 63
  - event tables
    - overview 60
    - removing messages 65
  - functions 71
  - HELP command 123
  - improving search performance 77
  - indexes
    - binary data types 62
    - changing 6
    - creating 6, 60
    - creating (binary data types) 62
    - creating (unsupported data types) 62
    - enhancing performance 63
    - maintaining 63
    - populating 64
    - searching 75
    - updating 6, 64
  - installing
    - DB2 Accessories Suite filter libraries 50, 52
    - db2\_install command 38
    - overview 27
    - response file 35
    - Setup Wizard 28
  - issuing commands 9
  - languages supported 23
  - linguistic processing 7
  - Log Formatter Tool 166
  - log tables 60
  - migrating
    - Net Search Extender to DB2 Text Search 171
  - Net Search Extender comparison 169
  - non-root upgrade 46
  - overview 1, 3, 10, 169
  - planning queries 73
  - problem determination 166
  - reconfiguring 41, 42
- DB2 Text Search *(continued)*
  - removing synonym dictionary 69, 165
  - rich text document support 9, 50, 52, 54
  - roles
    - database administrator 11
    - instance owner 10
    - user performing searches 11
    - user table owner 11
  - scenario 3
  - SCORE function 71, 74, 91
  - search argument syntax 77
  - search functions 71
  - searching indexes 71
  - searching with the SCORE function 76
  - security considerations 10
  - SQL 75
  - SQL search functions 89
  - START FOR TEXT command 100
  - starting 49
  - STOP FOR TEXT command 100
  - stopping 56
  - synonym dictionaries
    - overview 68
    - removing 165
  - Synonym Tool 165
  - triggers 17, 60
  - uninstalling DB2 Accessories Suite 53
  - unsupported data types 62
  - UPDATE INDEX command 118
  - upgrade 45
  - user roles 10
  - viewing index status 66
  - XML columns 94
  - XML documents 80, 84
  - XML search functions 89
  - xmlcolumn-contains function 71
  - XQuery 75
  - db2\_install command
    - installing DB2 product 38
  - db2icrt command
    - details 178
  - db2idrop command
    - details 182
  - db2iupdt command
    - details 184
  - db2iupgrade command
    - details 175
  - db2ts commands
    - ALTER INDEX 114
    - CLEANUP FOR TEXT 101
    - CLEAR COMMAND LOCKS 105
    - CLEAR EVENTS FOR INDEX 121
    - CREATE INDEX 107
    - DISABLE DATABASE FOR TEXT 103
    - DROP INDEX 112
    - ENABLE DATABASE FOR TEXT 102
    - HELP 123
    - START FOR TEXT 100
    - STOP FOR TEXT 100
    - UPDATE INDEX 118
  - DISABLE DATABASE FOR TEXT Text Search command 103
  - disk space
    - consumption with DB2 Text Search 18
  - doce\_install command
    - installing products 38
  - documentation
    - overview 189

documentation (*continued*)  
  PDF files 189  
  printed 189  
  terms and conditions of use 198  
DROP INDEX Text Search command 112

## E

ENABLE DATABASE FOR TEXT Text Search command 102  
enabling 55

## F

File descriptors 22  
filter libraries 50, 52  
functions  
  stored procedures  
    SYSTS\_ALTER 141  
    SYSTS\_CLEAR\_COMMANDLOCKS 151  
    SYSTS\_CLEAR\_EVENTS 149  
    SYSTS\_DROP 139  
    SYSTS\_ENABLE 128  
    SYSTS\_UPDATE 146  
  XQuery  
    xmlcolumn-contains 94

## H

heap memory consumption 13  
help  
  configuring language 194  
  SQL statements 193  
HELP command  
  Text Search 123

## I

Index location 20  
Index optimization 18  
Index planning and optimization 16  
Index source characteristics 17  
Index specific parameters for updates 20  
indexing threads 14  
installation  
  DB2 Text Search 38  
  db2\_install command 38  
  doce\_install command 38  
  manual  
    db2\_install 38  
    doce\_install 38

## L

linguistic processing  
  using n-gram segmentation 7  
Linux  
  installing  
    DB2 servers 32  
    DB2 Setup wizard 32  
  response file 37

## M

maintaining a text search index  
  backing up a text search index 67  
  restoring a text search index 67  
multiple predicates 74

## N

Net Search Extender (NSE)  
  comparison with Text Search 169  
  notices 201

## O

ordering DB2 books 192  
orphaned DB2 Text Search collections  
  deleting 57  
  identifying 57

## P

problem determination  
  information available 198  
  tutorials 198  
procedures  
  SYSTS\_ADMIN\_CMD 127

## Q

Queue memory size 15

## R

remove instance command 182  
response files  
  installation  
    Linux 37  
    UNIX 37  
  running setup 36  
RESULTLIMIT function 75  
rich text 55  
  installing DB2 Accessories Suite 50, 52  
  overview 9  
  setting up DB2 Text Search 54

## S

SCORE function  
  searching text search indexes 91  
services file  
  updating for TCP/IP communications 44  
SQL statements  
  help  
    displaying 193  
START FOR TEXT Text Search command 100  
STOP FOR TEXT Text Search command 100  
synonym dictionaries  
  adding 69  
  overview 68  
  removing 69, 165  
SYSTS\_ADMIN\_CMD procedure 127  
SYSTS\_ALTER stored procedure 141  
SYSTS\_CLEAR\_COMMANDLOCKS stored procedure 151  
SYSTS\_CLEAR\_EVENTS stored procedure 149

- SYSTS\_CREATE procedure 132
- SYSTS\_DISABLE procedure 130
- SYSTS\_DROP stored procedure 139
- SYSTS\_ENABLE stored procedure 128
- SYSTS\_UPDATE stored procedure 146

## T

- TCP/IP
  - updating services file 44
- terms and conditions
  - publications 198
- text search 55
- Text Search 13
- troubleshooting
  - online information 198
  - tutorials 198
- tuning 13
- tutorials
  - list 197
  - problem determination 198
  - troubleshooting 198
  - Visual Explain 197

## U

- UNIX
  - installing
    - DB2 Setup wizard 32
    - using the DB2 Setup wizard 32
    - response file installation 37
- UPDATE INDEX Text Search command 118
- update instances command 184
- updates
  - DB2 Information Center 195, 196
  - DB2 Text Search indexes 64
- upgrade instance command 175

## V

- views for DB2 Text Search
  - database-level information
    - overview 155
    - SYSIBMTS.TSDEFAULTS 155
    - SYSIBMTS.TSLOCKS 156
  - index-level information
    - overview 155
    - SYSIBMTS.TSCOLLECTIONNAMES 156
    - SYSIBMTS.TSCONFIGURATION 157
    - SYSIBMTS.TSEVENT 158
    - SYSIBMTS.TSINDEXES 156
    - SYSIBMTS.TSSTAGING 159

## W

- Windows
  - installing
    - DB2 servers (with DB2 Setup wizard) 29
    - response file method 36

## X

- XML
  - DB2 Text Search
    - EBNF grammar 80

- XML (*continued*)
  - DB2 Text Search (*continued*)
    - search syntax 84
- XML columns
  - text search 94
- xmlcolumn-contains function 94
- XQuery functions
  - xmlcolumn-contains 94





Printed in USA

SC27-2459-03



Spine information:

DB2 for Linux, UNIX, and Windows

**Version 9 Release 7**

**DB2 Text Search Guide**

