DB2 Version 9.5
for Linux, UNIX, and Windows

Version 9 Release 5

IBM

Text Search Guide
Updated December, 2010

DB2 Version 9.5
for Linux, UNIX, and Windows

Version 9 Release 5

IBM

Text Search Guide
Updated December, 2010

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.

Contents

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 x86 (32-bit) platforms
- Linux x86-64 (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 x86-64 (64-bit) platforms

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

1

- Synonym dictionary support

Chapter 2. DB2 Text Search overview

DB2 Text Search allows you to search text columns stored in a DB2 table. Starting with DB2 Version 9.5 Fix Pack 1, you can enable text search support to use the CONTAINS, SCORE, and xmlcolumn-contains built in 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 instance, but you can enable a database for only one of these text searching options.
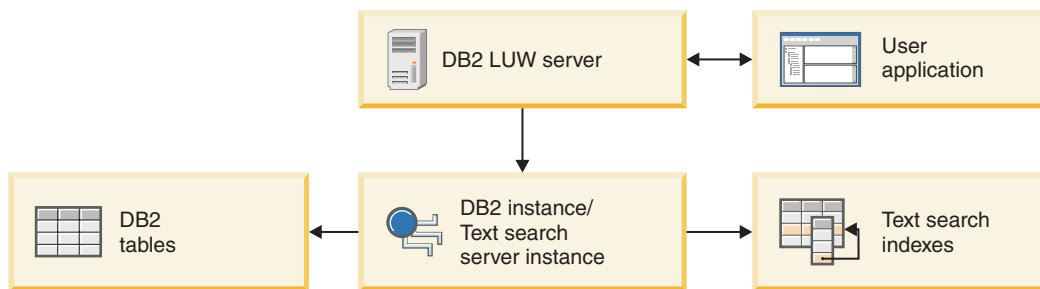
Figure 1. Deployment diagram for the DB2 Text Search server

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.

Related concepts

Appendix A, "DB2 Text Search and Net Search Extender comparison," on page 147
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.

Chapter 3, "Installing and configuring DB2 Text Search," on page 19
DB2 Text Search is an optionally installable component whose installation and
configuration are fully integrated with the installation of all DB2 server products.

"Text search solution planning" on page 10

"Searching with text search indexes" on page 59
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.

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 the same database.

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 searchwith XQuery if the index contains XML data.

As Figure 2 on page 5 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.
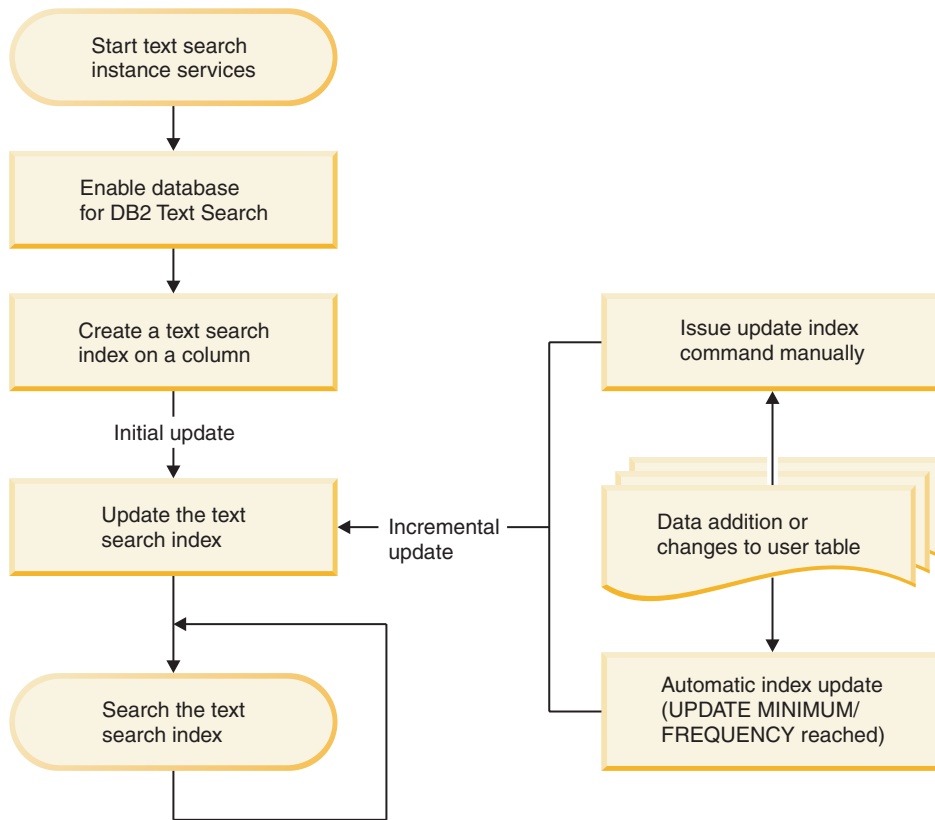
Figure 2. Setting up text search indexes for searching

Simple scenario

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 db2sampl 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"
```

Related concepts

"Text search index creation" on page 41
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.

"Text search index maintenance" on page 45
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.

"Searching with text search indexes" on page 59
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.

Related tasks

"Starting the DB2 Text Search instance service" on page 39
Before you can enable your database for text search, you must start the DB2 Text Search instance service.

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.

Note: The maximum recommended number of text search indexes is 100.

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

Text search index "productindex"		DB2 table "PRODUCT"	
		PID	**Name**
		100-01	Snow Shovel Base
		101-01	Snow Shovel Deluxe

Update index for text
(initial update)

Text search index "productindex"		DB2 table "PRODUCT" **PRODUCT table**	
		PID	**Name**
100-01	**Snow Shovel Base**	100-01	Snow Shovel Base
101-01	**Snow Shovel Deluxe**	101-01	Snow Shovel Deluxe
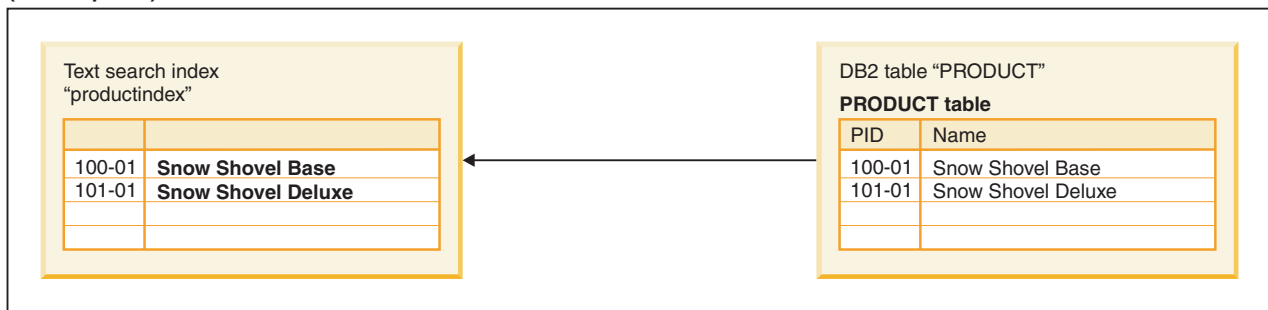
Update index for text
(incremental update)

Text search index "productindex"		Staging table "MYSCHEMA TSSTAGING.prodin"		DB2 table "PRODUCT" **PRODUCT table**	
	productindex	**PID**		**PID**	**Name**
100-01	Snow Shovel Base	101-01	mod	100-01	Snow Shovel Base
101-01	Snow Shovel **Extra**	103-01	chg	101-01	Snow Shovel Extra
103-01	**Snow Shovel Medium**			103-01	Snow Shovel Medium
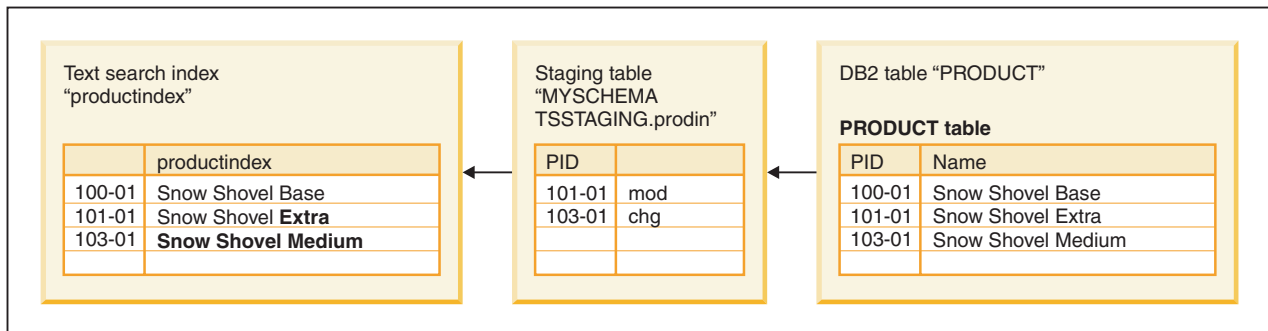
Figure 3. The two types of updates to a text search index

Related concepts

"Text search index creation" on page 41
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.

"Asynchronous index updates" on page 14
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.

Related tasks

"Updating a text search index" on page 46
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.

Linguistic processing for DB2 Text Search

DB2 Text Searchprovides 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 then 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.

Related reference

Appendix B, "Locales supported for DB2 Text Search," on page 151

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.

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.

Related reference

"SYSTS_ADMIN_CMD stored procedure - Run text search administration commands" in Administrative Routines and Views

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.
Ensure custom directories and default paths have proper privileges. Privileges need to be assigned manually to different roles. For example, a text search instance owner will need read/or write privileges, while a normal user may only require the read privilege.	Ensure that you check (and change, if necessary) the access permissions to the collection directories. For example if the extended security model is employed, then only DB2ADMNS should have permissions on the directory.

Table 1. Security-related issues and solutions (continued)

Issue	Solution
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 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 systems:

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

Related concepts

"Database administration authority (DBADM)" in Database Security Guide

Related tasks

"Starting the DB2 Text Search instance service" on page 39
Before you can enable your database for text search, you must start the DB2 Text Search instance service.

Related reference

"db2ts START FOR TEXT" on page 88

"db2ts STOP FOR TEXT" on page 89

"db2ts CLEANUP FOR TEXT" on page 90

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.

Related concepts

"Database administration authority (DBADM)" in Database Security Guide

Related tasks

"Enabling a database for DB2 Text Search" on page 39
You must enable each database that contains columns of text to be searched. You can enable a database forDB2 Text Search by using the db2ts ENABLE DATABASE FOR TEXT command or the SYSPROC.SYSTS_ENABLE stored procedure.

Related reference

"db2ts ENABLE DATABASE FOR TEXT" on page 91

"db2ts DISABLE DATABASE FOR TEXT" on page 92
This command somewhat reverses the changes (e.g., drops the text-search related tables and view) done by the command ENABLE DATABASE FOR TEXT.

"db2ts CLEAR COMMAND LOCKS" on page 94

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.

Related concepts

"Authorization, privileges, and object ownership" in SQL Reference, Volume 1

"Text search index maintenance" on page 45
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.

Related tasks

"Creating a text search index" on page 42
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.

Related reference

"db2ts CREATE INDEX" on page 96
Creates a text search index for a text column which allows the column data to be searched using text search functions.

"db2ts DROP INDEX" on page 101
Drops an existing text search index.

"db2ts ALTER INDEX" on page 102
This command changes the update characteristics of an index.

"db2ts UPDATE INDEX" on page 105
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.

"db2ts CLEAR EVENTS FOR INDEX" on page 107
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.

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)

Related concepts

"Searching with text search indexes" on page 59
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.

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 to a text column (for example, inserts, deletes, or
updates) 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.

Resource consumption

The text search server consumes system resources such as file descriptors 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 determine if this has
happened, check the server logs for an exception with the message string similar to
" too many open files."

If the text search server runs out of 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 hold good across login
   sessions.
3. Restart the text search server.

Related reference

"SYSIBMTS.TSSTAGING view" on page 139
The staging table stores the change operations on the user table that requires
synchronization with the text search index.

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

Note: The LONG VARCHAR and LONG VARGRAPHIC data types continue to be supported but are deprecated, not recommended, and might be removed in a future release.

Related tasks

"Creating a text search index on binary data types" on page 44
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.

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.

Related tasks

"Creating a text search index on unsupported data types" on page 44
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.

Related reference

"Supported data types" on page 14
The data types in the text columns that you want to index and search can be either binary or character.

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. .

Related reference
"SYSIBMTS.TSDEFAULTS view" on page 135
SYSIBMTS.TSDEFAULTS displays all the default values for all text search indexes in a database.

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 applies to the plain text document produced when transforming a rich text document to plain text.

If a text document is truncated during the parsing stage due to its size, you receive a warning that the text document has exceeded the maximum limit and the indexed document will be truncated. Text that is truncated will be indexed and searchable up to 10 000 000 characters in length. 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 deleting it from the DB2 table or by changing the value for the document to empty or null and then running db2ts update to remove the text from the index.

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.

Related reference

"SYSIBMTS.TSEVENT view" on page 138
The event view provides information about indexing status and error events.

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 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 platforms, and Figure 5 on page 20 describes the process on Linux and UNIX platforms.
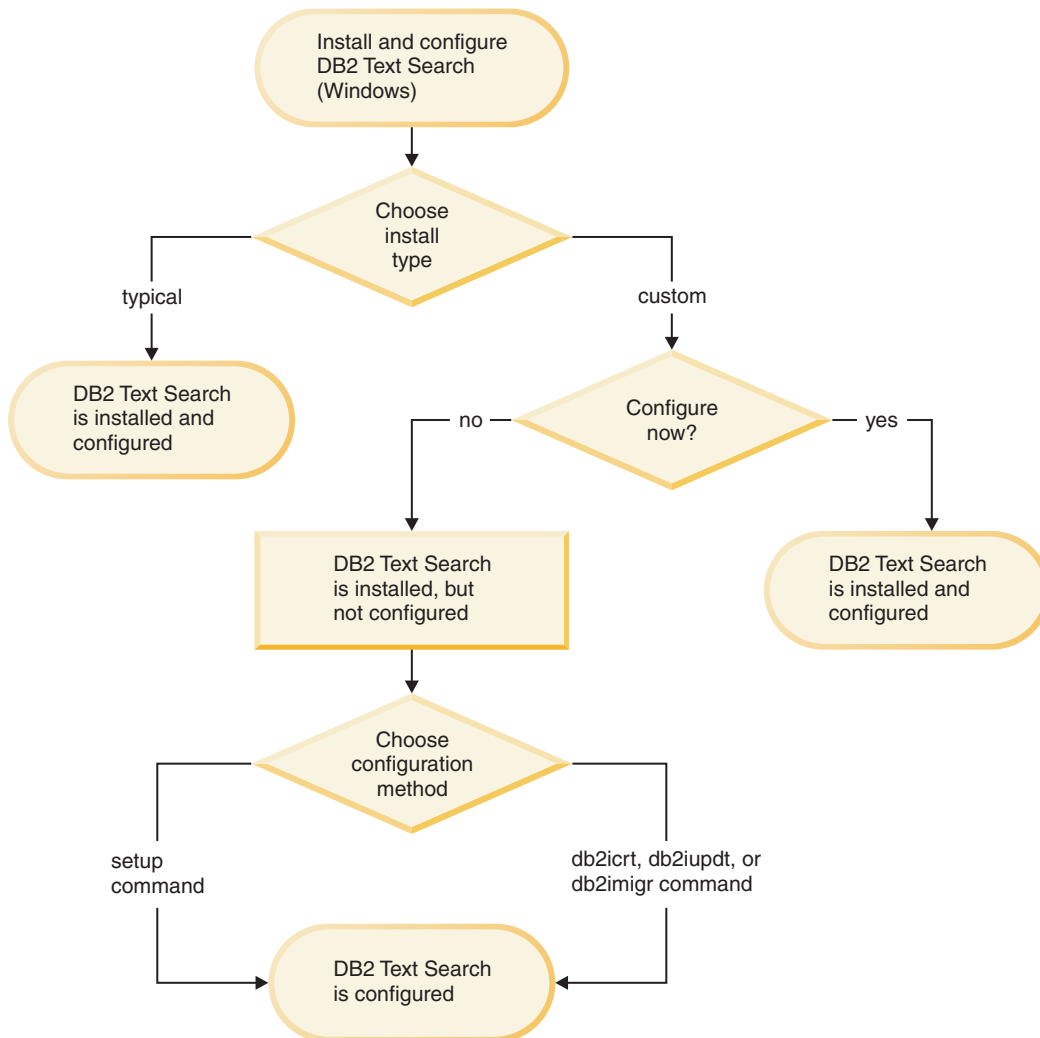
Figure 4. Installing and configuring on Windows platforms
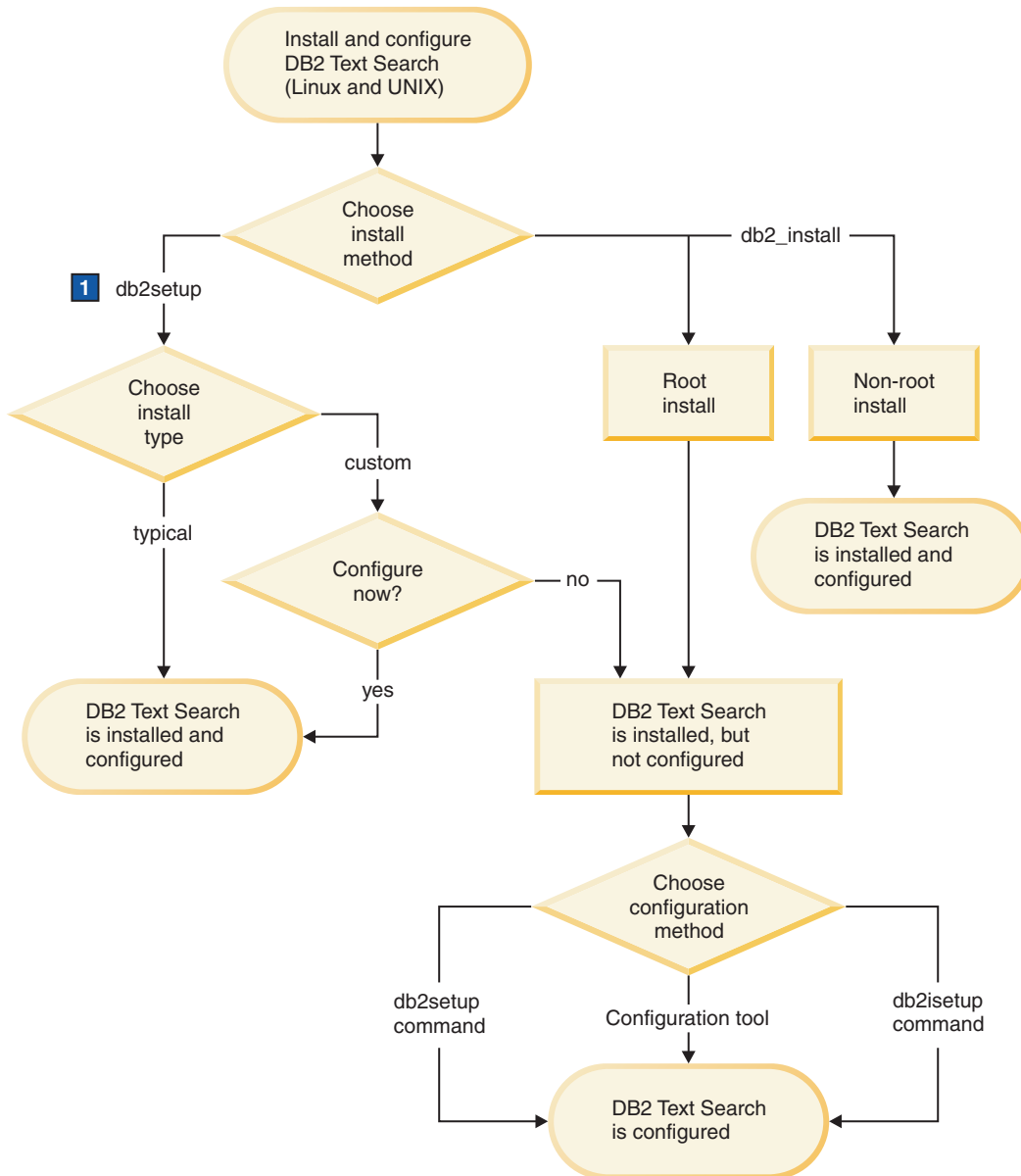
1. If you run db2setup as a non-root user, have your system administrator (who has SYSADM authority) run the **db2rfe** command afterwards to reserve the port number that you want in the services file.
Figure 5. Installing and configuring on Linux and UNIX platforms

> **Restrictions**
>
> DB2 Text Search does not support a Database Partitioning Feature (DPF) instance.
>
> **Related reference**
>
> "Configuration Tool for DB2 Text Search" on page 140
> 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.

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 typical installation or as a part of a custom installation of your DB2 product.

If you perform a typical installation of your DB2 product, DB2 Text Search is installed and configured automatically. You see the configuration panel only if the DB2 installer cannot automatically determine valid default configuration values.

If you perform a custom installation of your DB2 product, you can have DB2 Text Search automatically configured, or you can manually configure it later.

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

   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. Optional. 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.

Note: If you are upgrading from Version 9.5 GA to a fix pack, you will need to run the `setup.exe` file again. Then choose the "Modify" option and enable Server Support for DB2 Text Search from the list of components.

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 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 on Windows to register the DB2 server in 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.
- 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 web site. 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.

Restrictions
- The DB2 copy name and the instance name cannot start with a numeric value.
- 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 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, 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 the DB2 database product DVD, insert it into the drive. If enabled, the auto-run feature automatically starts the DB2 Setup Launchpad. If the auto-run does not work, use Windows Explorer to browse the DB2 database product DVD and double-click on 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 upgrade an existing DB2 copy, to add functionality to an existing DB2 copy, migrate an existing DB2 Version 8 or Version 9.1 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.

Your DB2 database product will be installed, by default, in the `Program_Files`\IBM\sqllib 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.
- Verify your installation.
- Perform the necessary post-installation tasks.

For information on errors encountered during installation, review the installation log file located in the `My Documents\DB2LOG\` directory. The log file uses the following format: DB2-*ProductAbrrev-DateTime*.log, for example, DB2-ESE-Tue Apr 04 17_04_45 2006.log.

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 16 GB.

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

Important notes when migrating from Version 9.1:

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

Related concepts

"DB2 object naming rules" in Data Servers, Databases, and Database Objects Guide

Related tasks

"Applying fix packs" in Quick Beginnings for DB2 Servers

"Verifying the installation of DB2 servers using First Steps (Linux and Windows)" in Getting Started with DB2 installation and administration on Linux and Windows

"Verifying the installation using the command line processor (CLP)" in Getting Started with DB2 installation and administration on Linux and Windows

"Extending the Active Directory Schema for LDAP directory services (Windows)" in Quick Beginnings for DB2 Servers

"Migrating a DB2 server (Windows)" in Migration Guide

"Setting up a partitioned database environment" in Partitioning and Clustering Guide

"Applying DB2licenses" in Quick Beginnings for DB2 Servers

"An overview of installing your DB2 server product (Windows)" in Quick Beginnings for DB2 Servers

Related reference

"UPDATE ADMIN CONFIGURATION " in Command Reference

"Installation requirements for DB2 servers and IBM data server clients (Windows)" in Getting Started with Database Application Development

"Required user accounts for installation of DB2 server products (Windows)" in Quick Beginnings for DB2 Servers

"Multiple DB2 copies overview" in Data Servers, Databases, and Database Objects Guide

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 *Quick Beginnings for DB2 Servers*
- Ensure that your system meets installation, memory, and disk requirements.
- You can install a DB2 database server using either root or non-root authority. For more information on non-root installation, see "Non-root installation overview (Linux and UNIX)" in *Quick Beginnings for 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.

Restrictions

- 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.

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 decompress and untar the product file.

   a. Decompress 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
   ```
   gnutar -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/disk1`) 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 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.

   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*/sqllib 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/V9.5

Linux /opt/ibm/db2/V9.5

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

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 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 16 GB.

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

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

Related concepts

"DB2 object naming rules" in Data Servers, Databases, and Database Objects Guide

"DB2installation methods" in Partitioning and Clustering Guide

"Non-root installation overview (Linux and UNIX)" in Quick Beginnings for IBM Data Server Clients

Related tasks

"Migrating a DB2 server (Linux and UNIX)" in Migration Guide

"Setting up a partitioned database environment" in Partitioning and Clustering Guide

Related reference

"UPDATE DATABASE CONFIGURATION " in Command Reference

"UPDATE ADMIN CONFIGURATION " in Command Reference

"Multiple DB2 copies overview" in Data Servers, Databases, and Database Objects Guide

Installing and configuring DB2 Text Search with a response file

You can install and configureDB2 Text Search as a part of a typical silent installation or custom silent installation of your DB2 product. Both types of installations use the setup or db2setup command with a response file.

If you perform a typical installation of your DB2 product, DB2 Text Search is installed and configured automatically.

If you perform a custom installation of your DB2 product, you must add a number of keywords to your response file to have DB2 Text Search installed and configured.

Note: If you are upgrading from Version 9.5 GA to a fix pack, you will need to enable the Text Search component in the response file and then run the installer in silent mode for the fix pack.

To perform a custom installation:

1. Add the line

   ```
   COMP = TEXT_SEARCH
   ```

   to the response file that you are using to install your DB2 product.

2. Optional. 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.

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

   - For root installations and non-root installations:
     ```
     db2inst_name.TEXT_SEARCH_HTTP_PORT_NUMBER = port-number
     ```

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 (the default setting).

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

Note: 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.

Installing a DB2 product using a response file (Windows)

Before you begin the installation, ensure that:
- Your system meets all of the memory, hardware, and software requirements to install your DB2 product.

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

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 machine where the DB2 installation files reside.
   - *directory_sharename* represents the share name of the directory on the network drive or DVD drive where the DB2 installation files reside.
   - *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 db2nt and is located on the remote server codesrv, 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 on errors encountered during installation, review the installation

log file located in the `My Documents\DB2LOG\` directory. The log file uses the following format: DB2-ProductAbrrev-DateTime.log, for example, `DB2-ESE-Tue Apr 04 17_04_45 2007.log`.

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.

Related concepts

"DB2 Information Center installation options" in Quick Beginnings for DB2 Servers

Related tasks

"Installing the DB2 Information Center using the DB2 Setup wizard (Windows)" in Quick Beginnings for DB2 Servers

Related reference

"IBM Data Server Runtime Client installation command-line options" in Quick Beginnings for IBM Data Server Clients

"Command line options to install IBM Data Server Driver Package (Windows)" in Quick Beginnings for IBM Data Server Clients

Installing a DB2 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/V9.5`
- For Linux `/opt/ibm/db2/V9.5`

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.

Related tasks

"Installing the DB2 Information Center using the DB2 Setup wizard (Linux)" in Quick Beginnings for DB2 Servers

Related reference

"db2_install - Install DB2 database product " in Command Reference

"installFixPack - Update installed DB2 database products " in Command Reference

Installing DB2 Text Search using db2_install (Linux and UNIX)

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

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

If this is a non-root installation, a DB2 instance is created and DB2 Text Search will be automatically configured. 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.

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 *Quick Beginnings for DB2 Servers* documentation to review installation prerequisites and other important setup information.

- You can install a DB2 database product using either `root` or `non-root` user authority. However, you cannot use the doce_install to install the *DB2 Information Center* under `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 and UNIX 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 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 decompress and untar the product file.
   a. Decompress 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

```
gnutar -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/V9.5
  - for Linux operating systems: /opt/ibm/db2/V9.5

  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

  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.

Related concepts

"Non-root installation overview (Linux and UNIX)" in Quick Beginnings for IBM Data Server Clients

"Limitations of non-root installations" in Quick Beginnings for IBM Data Server Clients

Related tasks

"Manually configuring DB2 servers after installation" in Quick Beginnings for DB2 Servers

"Removing DB2 database products using the db2_deinstall or doce_deinstall commands (Linux and UNIX)" in Quick Beginnings for DB2 Servers

"Listing DB2 products installed on your system (Linux and UNIX)" in Troubleshooting Guide

Related reference

"db2_install - Install DB2 database product " in Command Reference

"doce_install - Install DB2 Information Center " in Command Reference

"Installation requirements for DB2 database products" in Quick Beginnings for DB2 Servers

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 reconfiguring DB2 Text Search, stop the text search instance services, as outlined in Starting and stopping the DB2 Text Search instance services.

Determining whether DB2 Text Search is configured
To 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.

Configuring 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.
- Rerun the GUI installation as described in Installing and configuring DB2 Text Search with the DB2 Setup Wizard .
- Use the Configuration Tool. 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 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.
- Rerun the GUI installation as described in Installing and configuring DB2 Text Search with the DB2 Setup Wizard.
- Issue the db2icrt, db2iupdt, or db2imigr command.

Reconfiguring DB2 Text Search

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

On Linux and UNIX operating systems, use one of the following methods to reconfigure DB2 Text Search:
- Rerun the silent installation as described in Installing and configuring DB2 Text Search with a response file.
- Use the Configuration Tool.
- Issue the db2isetup command with the **-r** option.

On Windows operating systems, use one of the following methods to reconfigure DB2 Text Search:
- Rerun the silent installation as described in Installing and configuring DB2 Text Search with a response file.
- Use the Configuration Tool.
- Run the db2icrt, db2iupdt, or db2imigr 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.

   - On Linux and UNIX operating systems, 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 the DB2 Text Search installation directory, and the **maxHeapSize** specifies the size of the heap that will be used.

   - 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
   -hostName localhost
   ```

   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*.

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

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

Chapter 4. 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.

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.

Related reference

"db2ts START FOR TEXT" on page 88

"db2ts STOP FOR TEXT" on page 89

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 forDB2 Text Search by using the db2ts ENABLE DATABASE FOR TEXT command or the SYSPROC.SYSTS_ENABLE stored procedure.

You must start the DB2 instance service.

You require SYSADM authority.

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', ?)

Related reference

"db2ts ENABLE DATABASE FOR TEXT" on page 91

"SYSTS_ENABLE procedure - Enable current database for text search" in Administrative Routines and Views

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.

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.

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

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.

Related reference

"SYSTS_DISABLE procedure - Disable current database for text search" in Administrative Routines and Views

"db2ts DISABLE DATABASE FOR TEXT" on page 92
This command somewhat reverses the changes (e.g., drops the text-search related tables and view) done by the command ENABLE DATABASE FOR TEXT.

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.

Related concepts

"Asynchronous index updates" on page 14
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.

Related reference

"SYSIBMTS.TSDEFAULTS view" on page 135
SYSIBMTS.TSDEFAULTS displays all the default values for all text search indexes in a database.

"SYSIBMTS.TSEVENT view" on page 138
The event view provides information about indexing status and error events.

"SYSIBMTS.TSSTAGING view" on page 139
The staging table stores the change operations on the user table that requires synchronization with the text search index.

"SYSIBMTS.TSINDEXES view" on page 137
You can view the db2ts CREATE INDEX command parameters and generated settings using SYSIBMTS.TSINDEXES.

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.

Creating a text search index requires CONTROL privilege on the table.

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

  Note: The LONG VARCHAR and LONG VARGRAPHIC data types continue to be supported but are deprecated, not recommended, and might be removed in a future release.
- 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:
  - 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

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', ?)
```

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
called 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.)

"Document size considerations" on page 16
DB2 Text Search has limits on the size of the document that can be indexed and on the number of characters within that document.

Related reference

"SYSTS_CREATE procedure - Create a text search index on a column" in Administrative Routines and Views

"db2ts CREATE INDEX" on page 96
Creates a text search index for a text column which allows the column data to be searched using text search functions.

"SYSTS_ADMIN_CMD stored procedure - Run text search administration commands" in Administrative Routines and Views

Appendix B, "Locales supported for DB2 Text Search," on page 151

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.

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 INDSCHEMA='schema-name'
and INDNAME='index-name'"
```

Related reference

"Supported data types" on page 14
The data types in the text columns that you want to index and search can be either binary or character.

"SYSIBMTS.TSINDEXES view" on page 137
You can view the db2ts CREATE INDEX command parameters and generated settings using SYSIBMTS.TSINDEXES.

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.

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

Note: The LONG VARCHAR and LONG VARGRAPHIC data types continue to be supported but are deprecated, not recommended, and might be removed in a future release.

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.

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.

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, which increases your per-document overhead.

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.

Related tasks

"Dropping a text search index" on page 49
When you no longer intend to perform text searches in a text column, you can drop the text search index.

"Viewing text search index status" on page 47
To get information on the current text search indexes within a database, you can query the administrative views or use the Administration Tool.

"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.

Related reference

"Administration Tool for DB2 Text Search" on page 142
The Administration Tool is a command-line tool that provides a convenient way to perform several administrative tasks related to DB2 Text Search.

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.

Updating a text search index requires the CONTROL privilege on the target table.

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. Because updates are resource intensive, ensure that no more than 10 index updates are running at the same time.

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 on the same index 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 instance, 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:

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>'))
```

Too 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"
```

Related reference
"db2ts UPDATE INDEX" on page 105
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.
"SYSTS_UPDATE procedure - Update the text search index" in Administrative Routines and Views
"SYSTS_ADMIN_CMD stored procedure - Run text search administration commands" in Administrative Routines and Views

Viewing text search index status

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

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
```

Related concepts

"Maintenance and troubleshooting for DB2 Text Search" on page 51
Maintenance tasks for DB2 Text Search include deleting orphaned text search
collections and clearing text search index events.

Related reference

"Administration Tool for DB2 Text Search" on page 142
The Administration Tool is a command-line tool that provides a convenient way to
perform several administrative tasks related to DB2 Text Search.

"SYSIBMTS.TSINDEXES view" on page 137
You can view the db2ts CREATE INDEX command parameters and generated
settings using SYSIBMTS.TSINDEXES.

Altering a text search index

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

The db2ts ALTER INDEX command requires the CONTROL privilege on the target
table.

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.

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.

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"
```

Related reference

"db2ts ALTER INDEX" on page 102

This command changes the update characteristics of an index.

"SYSTS_ALTER procedure - Change the update characteristics of an index" in Administrative Routines and Views

"SYSTS_ADMIN_CMD stored procedure - Run text search administration commands" in Administrative Routines and Views

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.

CONTROL privilege is required on the table which the text search index is associated with.

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 are 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 re-issued as before.

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

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.

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.

Related concepts

"Text search index maintenance" on page 45
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.

Related reference

"db2ts DROP INDEX" on page 101
Drops an existing text search index.

"SYSTS_DROP procedure - Drop a text search index" in Administrative Routines and Views

"SYSTS_ADMIN_CMD stored procedure - Run text search administration commands" in Administrative Routines and Views

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>
```

Related tasks

"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.

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.

- 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.

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.

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

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

Related concepts

"Synonym dictionaries for DB2 Text Search" on page 50
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.

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.

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.

Related concepts

"Synonym dictionaries for DB2 Text Search" on page 50
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.

Related tasks

"Adding a synonym dictionary for DB2 Text Search" on page 50
You can easily add a synonym dictionary to a text search index by using the Synonym Tool.

Maintenance and troubleshooting for DB2 Text Search

Maintenance tasks for DB2 Text Search include deleting orphaned text search collections and clearing text search index events.

There are three resources that can help you perform problem determination for DB2 Text Search: the event table, the db2trc command, and the Log Formatter Tool.

A complete list of all of the DB2 Text Search server messages can be found at www.ibm.com/support/docview.wss?uid=swg21298225.

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.

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/sqllib/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/sqllib/db2tss/config
-collectionName tigertail_DBCP1208_TS867530_0000
```

Related reference

"Administration Tool for DB2 Text Search" on page 142
The Administration Tool is a command-line tool that provides a convenient way to perform several administrative tasks related to DB2 Text Search.

Tracing to investigate text search errors

If you need to report an error to an IBM representative, you might be asked to switch on tracing so that information can be written to a file that can be used for locating the error.

You can use the DB2 tracing facility, db2trc, to capture information about the problems that might occur while you are performing text searches or administrative operations associated with text searches.

Because system performance is affected when tracing is switched on, use the trace facility only when directed by an IBM Support Center representative or by your technical support representative.

To turn tracing on and receive information specific to DB2 Text Search, run the db2trc command using a mask with a component code for CIE (155), as follows:

```
 db2trc on -m '*.*.CIE.*.*'
```

To help diagnose severe errors, it might also help to look in the db2diag.log file.

Steps for obtaining OmniFind text search trace for troubleshooting:

- Turn OmniFind trace on, run the following command:

  For Unix/Linux:

  $DB2PATH/db2tss/bin/adminTool.sh configureTrace -configPath $DB2PATH/db2tss/config -trace on;

  For Windows:

  %DB2PATH%\db2tss\bin\adminTool.bat configureTrace -configPath %DB2INSTPROF%\%DB2INSTANCE%\db2tss\config -trace on;

- Run the command that gives db2 text search error

- Dump trace information to a file, run the following command:

  For Unix/Linux:

  $DB2PATH/db2tss/bin/logformatter.sh -f $DB2PATH/db2tss/log/Trace.0.log > <filename>

  For Windows:

  %DB2PATH%\db2tss\bin\logformatter.bat -f %DB2INSTPROF%\%DB2INSTANCE%\db2tss\log\Trace.0.log > <filename>

- Turn OmniFind trace off, run the following command:

  For Unix/Linux:

  $DB2PATH/db2tss/bin/adminTool.sh configureTrace -configPath $DB2PATH/db2tss/config -trace off;

  For Windows:

  %DB2PATH%\db2tss\bin\adminTool.bat configureTrace -configPath %DB2INSTPROF%\%DB2INSTANCE%\db2tss\config -trace off;

Related reference

"db2trc - Trace " in Command Reference

"db2diag - db2diag logs analysis tool " in Command Reference

Clearing text search index events

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

Clearing the event view requires the CONTROL privilege.

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.

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')
    ```

Related reference

"db2ts CLEAR EVENTS FOR INDEX" on page 107
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.

"SYSTS_CLEAR_EVENTS procedure - Delete indexing events from an index's event table" in Administrative Routines and Views

"SYSIBMTS.TSEVENT view" on page 138
The event view provides information about indexing status and error events.

Troubleshooting hints and tips

Use the information in the following table to help troubleshoot problems.

Table 2. Issues and solutions

Issue	Solution
If you drop and re-create a text index while a DB2 connection session is active, a plan for a previously used query might be cached and reused, thus producing incorrect results. If the DB2 Text Search engine encounters this issue, IQQG0037W will be returned, indicating that an incorrect collection name was used for the search.	Use the FLUSH PACKAGE CACHE DYNAMIC statement to remove the execution plans for the SQL statements from the statement cache.
If you drop your DB2 instance using the db2idrop, db2iupdt, or db2imigr command, the entire `sqllib` directory in the instance is removed, including the `sqllib/db2tss/config` directory. This will make it hard to reuse text search instance databases.	Back up the `sqllib/db2tss/config` directory before issuing the db2idrop, db2iupdt, or db2imigr command. Also, use the **printAll** option of the configTool command to save a record of the configuration parameters, which should help you to configure the new instance.
You receive the message CIE00341 "Update index errors" after issuing a db2ts update command.	Check the db2diag.log and the event view named in the error for details.

Table 2. Issues and solutions (continued)

Issue	Solution
You find a message in the event table or the db2diag.log: "IQQD0085E The requested operation can't be done. The server is running in safe mode due to System has an IO error."	Use the logformatter tool to inspect the textserver log file. For more details see, Log Formatter Tool for DB2 Text Search . Usually, the error IQQD0085E is caused by a disk full situation flagged by IQQG0020E java.io.IOException: No space left on device" A disk full error might cause the text index metadata to be inconsistent. Use the Administration Tool to check whether there are any orphaned collections and to remove them, as outlined in Deleting orphaned text search . Note that you might not be able to reuse the text index name that was being used when you encountered the disk full error.
You find the message "IQQD0072E An error occurred while communicating with the server localhost at port number *<nnn>* . Unable to send the request to the server or the client timed out while receiving a response from the server." in the event table or db2diag.log.	This error may be caused by 1. The text server process was interrupted 2. A TCP/IP port issue 3. A disk full scenario. See above for using the logformatter tool to investigate this case.
You receive the message CIE00301 "Insufficient authority" or CIE00377 "Only instance owner *<instance-owner>* can use that command" after issuing a db2ts START FOR TEXT command.	For Windows users, have a user with sufficient authorization issue db2ts START FOR TEXT. For Linux and UNIX, have the instance owner issue this command.

Table 2. Issues and solutions (continued)

Issue	Solution

You receive the message CIE00311 telling you that an internal file cannot be opened. This message can indicate a missing configuration directory, or indicate that a file may have been lost or corrupted, for example, due to a disk full error on the filesystem where db2tss/config is located, or because of a problem during the backup of the db2tss/config directory.	Check to see if the instance has a 'config' directory for text search.
	
	• If the 'config' directory is missing ensure that DB2 Text Search was installed and configured.
	
	• If the 'config' directory is in another location, add a symbolic link to it (UNIX).
	
	If this error is being caused by a missing or corrupted file:
	
	• If the file is 'ciedem.dat' you can regenerate it with the following steps:
	
	  1. Issue the db2ts STOP FOR TEXT command.
	
	  2. Locate the ciedem.dat file. It should be in <instance home>/sqllib/db2tss/ config (UNIX) or <instance profile path>\instancename\db2tss\config (Windows)(You can find the instance profile path using the following: db2set DB2INSTPROF)
	
	  3. If the file exists, back it up and then remove it.
	
	  4. Issue the db2ts START FOR TEXT command. (The internal file should be regenerated as a 0 byte file)
	
	  5. Issue the db2ts CLEANUP FOR TEXT command
	
	• If the file is something other than 'ciedem.dat, contact IBM Support.

If you run the db2iupdt command with the /p option to change the instance profile path, the DB2 Text Search configuration path and the log path need to be manually updated after the instance is updated.	After db2iupdt completes, update the **configPath** and **logPath** parameters using the configTool as follows. From the DB2 command window for your current DB2 copy, and with the instance you just updated as the current selected instance, run: configTool configureParams -logPath *newlogpath* -configPath *newconfigpath*

Chapter 5. Searches

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, we have to ensure the rows from a text-indexed table have not been moved to other nodes when we apply those functions. The most reliable approach is to require the text search function occurs right next base table access.

Related reference

"CONTAINS function" on page 77
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.

"SCORE function" on page 80
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.

Appendix B, "Locales supported for DB2 Text Search," on page 151

"xmlcolumn-contains function" on page 82
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.

Text search query planning

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

- **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: The optional marker "?*search-term*" does not impact performance.

- **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.

- **Word-form derivation, or stemming, can have exceptions**
  Some search terms are not considered to be derived from their base forms, so some documents that appear to be relevant documents might not be returned. For example, a search for the term winning does not return documents

containing `win`, `wins`, and `won`. In this case, you should consider including the base form, by searching for `winning OR win`.

- **The score of a document is dynamic**
  Updates to a document can cause the score of a term used to search that document to change.
- **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')...
  ```

- **Multiple instances of RESULTLIMIT within a query require the same search argument to produce predictable results**
  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 on XML namespaces requires a workaround**
  You can index XML documents that contain namespace bindings without generating errors, but the namespace information is removed from each tag. As a result, text searches on XML documents with namespace bindings can lead to undesired results.

  However, there is a workaround to this limitation for queries that use DB2 XQuery. The DB2 Text Search engine is not namespace aware, but you can use the DB2 XQuery support for namespaces to do namespace filtering for the unwanted documents returned from a text search.

  Consider the following example in which the default database environment variable is set to SAMPLE and a text search index called prod_desc_idx is created on the PRODUCT table:

  ```
  db2ts "ENABLE DATABASE FOR TEXT"
  db2ts "CREATE INDEX prod_desc_idx FOR TEXT ON product(description)"
  ```

  Now, a new row with the namespace http://posample.org/wheelshovel is added to the PRODUCT table, which already has two XML documents with the namespace http://posample.org:

  ```
  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>'))
  ```

The text search index is then updated, as follows:

```
db2ts "UPDATE INDEX prod_desc_idx FOR TEXT"
```

The following XQuery expression, which specifies the default element as http://posample.org, returns all documents that have the matching XPath /product/description/details that contains the word `ergonomic`:

```
xquery declare default element namespace "http://posample.org";
db2-fn:xmlcolumn-contains('PRODUCT.DESCRIPTION', '@xpath:
''/product/description/details [. contains ("ergonomic")]''')
```

Three documents are returned, two of which are expected because they have the namespace http://posample.org and one of which is unexpected because it has the namespace http://posample.org/wheelshovel.

The following XQuery expression uses the path expression /product/.. to use the DB2 XQuery support for XML search and namespaces to filter the documents returned by DB2 Text Search engine so that only documents with the namespace http://posample.org are returned:

```
xquery declare default element namespace "http://posample.org";
db2-fn:xmlcolumn-contains('PRODUCT.DESCRIPTION', '@xpath:
''/product/description/details [. contains ("ergonomic")]''')/product/..
```

Note: SQL queries can use DB2 XQuery to force namespace filtering. Given the previous example, the corresponding expression using an SQL query is as follows:

```
xquery declare default element namespace "http://posample.org";
db2-fn:sqlquery("select description from product where
contains(description, '@xpath:''/product/description/details
[. contains (""ergonomic"")]''') = 1")
```

The workaround is as follows:

```
xquery declare default element namespace "http://posample.org";
db2-fn:sqlquery("select description from product where
contains(description, '@xpath:''/product/description/details
[. contains (""ergonomic"")]''') = 1")/product/..
```

Similarly, to access a specific element in the document (as opposed to just having the matching document returned, as in the previous query), the following query can be used:

```
xquery declare default element namespace "http://posample.org";
db2-fn:xmlcolumn-contains('PRODUCT.DESCRIPTION', '@xpath:
''/product/description/details [. contains ("ergonomic")]''')
/product/description[price > 20]/name
```

Note: This workaround is limited and might not work as expected if, for example, multiple product elements exist within a document.

Related reference

"Text search argument syntax" on page 66
A search argument comprises one or more terms and optional search parameters, separated by white space, that you specify to search in text documents.

"Search syntax for XML documents" on page 72
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.

"CONTAINS function" on page 77
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.

"SCORE function" on page 80
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.

"xmlcolumn-contains function" on page 82
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.

Searching in text search indexes

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

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

- **Searching 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"
  ```

- **Searching 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.DESCRIPTION','"snow shovel"')
  return <result> {\$info/description/name, \$info/description/price} </result>"
  ```

  Note:

  - For both SQL and XQuery type searches, search terms or phrases can contain any characters, however only the following characters are actively used in the search [A-Za-z0-9][A-Za-z0-9]*_-.

  - The underscore character is only indexed if it is not at the end of a word.

  - The dash character is indexed as a standalone word if it is at the beginning or end of a word. It can be searched, with the proper escape clause. If however, the dash is in the middle of a word, it is considered a white space and not indexed.

– 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 platforms.

Related concepts

"Searching XML documents using DB2 Text Search" on page 70
DB2 Text Search supports using XML search for searching XML documents.

Related reference

"CONTAINS function" on page 77
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.

"xmlcolumn-contains function" on page 82
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.

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.

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.

Note: Do not interpret scores returned for XML search queries in the same manner as you interpret scores that SCORE returns from plain text documents. Non-structured queries use only the text terms for scoring.

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
```

Related reference

"SCORE function" on page 80
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.

Enhancing performance during text searches

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

- Use the EXPLAIN statement to check the processing plan of the DB2 optimizer when searching within SQL.
- 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.

Related reference

"EXPLAIN " in SQL Reference, Volume 2

"RUNSTATS " in Command Reference

"CONTAINS function" on page 77
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.

"SCORE function" on page 80
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.

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 the following special characters 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 3. Boolean operators for search queries

Operator	Example	Query results
AND	King AND Lear 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 4. Occurrence modifiers

Modifier	Example	Query result
NOT -	Hamlet NOT Othello Hamlet -Othello	Returns documents that contain Hamlet but not Othello.

Table 4. Occurrence modifiers (continued)

Modifier	Example	Query result
+	Lear + King	Returns documents that contain the terms Lear and King. Documents containing Lear and monarch aren't returned.

Table 5. Other modifiers

Modifier	Example	Query result
term1 or *phrase1*^*number* *term2* or *phrase2*	Hamlet^2 Othello 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* k*ng *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.
?	mea? be?n "?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*" "*exact term*" "*phrase with wildcard*"	"King Lear" "king" "John * Kennedy" "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. 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 *initial* Kennedy in the fourth example.
( )	(Hamlet OR Othello) AND plays	Returns documents that contain the following terms: • The term Hamlet or Othello • The term plays
\	\(1\+1\)\: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 6. Valid XML search queries

Query	Description
/	The root node; any document
/sentences	Any document with a top-level tag of sentences

Table 6. Valid XML search queries (continued)

Query	Description
//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
/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.

Related reference

"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.

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 below.

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
> below.

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 a 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
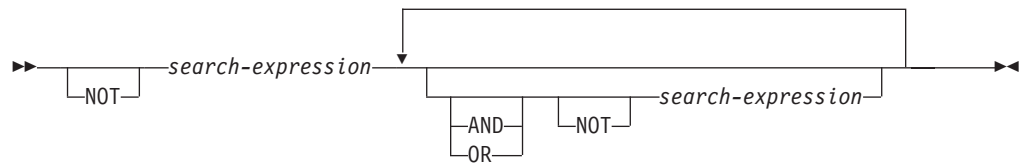
search-expression
>   A DB2 Text Search XML search query. DB2 Text Search uses a search expression to search node or attribute values in an XML document.

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 numerical arguments. Numerical 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──┬─contains─┬──(──literal──)──────────────────►◄
                     └─excludes─┘
```

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')]'
```

Related reference

"Text search argument syntax" on page 66
A search argument comprises one or more terms and optional search parameters,
separated by white space, that you specify to search in text documents.

Related information

"Abbreviated syntax for path expressions" in XQuery Reference

Chapter 6. Reference

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────┤
   │                  ┌─OFF─┐ │
   └─SYNONYM──=───────┴─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 cannot exceed 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 FIRSTNME, 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──────────────────────)─►◄
                                          └─,─string-constant─┘
                                                         (1)
```

Notes:

1   *string-constant* must conform to the rules for search-argument-options.

search-argument-options:

```
        (1)
┌──────────────────────────────────────┐
▼                                       │
├─┬──────────────────────────┬──────────┤
  ├─QUERYLANGUAGE──=─locale──┤
  ├─RESULTLIMIT──=─value─────┤
  │                 ┌─OFF─┐
  └─SYNONYM──=──┴─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 cannot exceed 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
```

Related tasks

"Adding a synonym dictionary for DB2 Text Search" on page 50
You can easily add a synonym dictionary to a text search index by using the Synonym Tool.

"Searching text search indexes using SCORE" on page 65
You can use the SCORE function to find out the extent to which a document matches a search argument.

Related reference

"Text search argument syntax" on page 66
A search argument comprises one or more terms and optional search parameters, separated by white space, that you specify to search in text documents.

Appendix B, "Locales supported for DB2 Text Search," on page 151

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:

```
          (1)
  ┌─────────────────────────────┐
──┼──┬──────────────────────────┼──────────────────────────────────
     │  ─QUERYLANGUAGE=locale─   │
     ├──RESULTLIMIT=value────────┤
     │              ─OFF─        │
     └──SYNONYM=────┬─OFF─┬──────┘
                    └─ON──┘
```
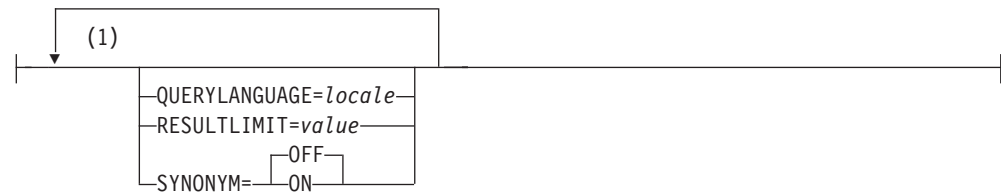
Notes:

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.DESCRIPTION", "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.DESCRIPTION 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.DESCRIPTION', '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.DESCRIPTION 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.DESCRIPTION', '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.DESCRIPTION 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.DESCRIPTION',
  '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'')')
```

Related concepts

"Text search index creation" on page 41
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.

Related tasks

"Adding a synonym dictionary for DB2 Text Search" on page 50
You can easily add a synonym dictionary to a text search index by using the Synonym Tool.

Related reference

"Text search argument syntax" on page 66
A search argument comprises one or more terms and optional search parameters, separated by white space, that you specify to search in text documents.

"Search syntax for XML documents" on page 72
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.

"CONTAINS function" on page 77
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.

Appendix B, "Locales supported for DB2 Text Search," on page 151

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

To issue the command successfully, the user must be the DB2 instance owner for the current instance.

Required connection

This command must be issued from the DB2 database server. This command is not supported from an IBM Data Server Client.

Command syntax

```
►►──START FOR TEXT──────────────────────────────────────────────────►◄
```

Command parameters

None

Related tasks

"Issuing text search commands" on page 9
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.

"Starting the DB2 Text Search instance service" on page 39
Before you can enable your database for text search, you must start the DB2 Text Search instance service.

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

To issue the command successfully, the user must be the DB2 instance owner for the current instance.

Required connection

This command must be issued from the DB2 database server. This command is not supported from an IBM Data Server Client.

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

Related tasks

"Stopping the DB2 Text Search instance service" on page 40
When you stop the DB2 Text Search instance service, any currently running db2ts commands will be completed, but you cannot issue any new ones.

"Issuing text search commands" on page 9
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.

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 for the current instance.

Required connection

This command must be issued from the DB2 database server. This command is not supported from an IBM Data Server Client.

Command syntax

```
►►──CLEANUP FOR TEXT──────────────────────────────────────────►◄
```

Command parameters

None

Related concepts

"Maintenance and troubleshooting for DB2 Text Search" on page 51
Maintenance tasks for DB2 Text Search include deleting orphaned text search
collections and clearing text search index events.

Related tasks

"Issuing text search commands" on page 9
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.

db2ts ENABLE DATABASE FOR TEXT

Enables DB2 Text Search for the current database. This command must be issued
successfully before it will be possible to create text search indexes on columns in
tables within the database.

For execution, the command needs to be prefixed with db2ts at the command line.

Authorization

A connection to the database must be established by an authorization ID with
system administrator authorization (*sysadm*).

Required connection

Database

Command syntax

►►──ENABLE DATABASE FOR TEXT──┤ connection options ├────────────────────►◄

connection options:

├──┬───┬──┤
   └─CONNECT TO──*database-name*──┬──────────────────────────────┬─┘
                                  └─USER──*username*──USING──*password*─┘

Command parameters

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

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.

Related tasks

"Enabling a database for DB2 Text Search" on page 39
You must enable each database that contains columns of text to be searched. You can enable a database forDB2 Text Search by using the db2ts ENABLE DATABASE FOR TEXT command or the SYSPROC.SYSTS_ENABLE stored procedure.

"Issuing text search commands" on page 9
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.

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 *username* for the database connection must have *dbadm* or *sysadm* authorization.

Required connection

Database

Command syntax

```
►►──DISABLE DATABASE FOR TEXT──┬──────────┬──│ connection options │──────────►◄
                               └─FORCE─┘
```

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 db2diag.log. 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

Related tasks

"Disabling a database for DB2 Text Search" on page 40
Disable a database when you no longer intend to perform text searches in that database.

"Issuing text search commands" on page 9
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.

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* or *sysadm* 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 (according to DB2 catalog views).

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.

Related tasks

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.

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

CONTROL privilege is required on the table in which the text search index will be
created by the authorization ID that established the current database connection.
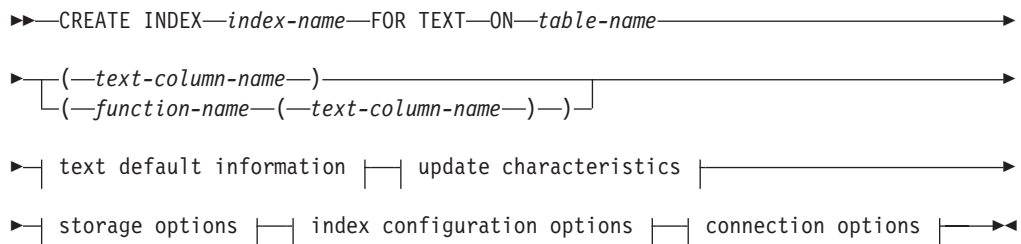
Required connection

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:

```
├──┬──────────────────────┬──┬─────────────────────┬──┬──────────────────┬──┤
   └─CODEPAGE──code-page──┘  └─LANGUAGE──locale─────┘  └─FORMAT──format───┘
```

update characteristics:

```
├──┬───────────────────────────────────────────────────┬────────────────────►
   └─UPDATE FREQUENCY──┬─NONE────────────────────┬──────┘
                       └─┤ update frequency ├─────┘

►─┤ incremental update characteristics ├────────────────────────────────────┤
```

update frequency:

```
├─D─(─┬──────*──────┬─)─H─(─┬──────*──────┬─)─M─(─┬─integer3─┬─)─────────────┤
       │     ┌─,─┐    │       │     ┌─,─┐    │       └────,────┘
       └──▼─integer1─┘       └──▼─integer2─┘
```

incremental update characteristics:

```
├─┬─────────────────────────────┬─────────────────────────────────────────┤
  └─UPDATE MINIMUM─minchanges─┘
```

storage options:

```
├─┬──────────────────────────────────────────┬──────────────────────────────▶
  └─COLLECTION DIRECTORY─directory─┘

▶─┬──────────────────────────────────────────────┬──────────────────────────┤
  └─ADMINISTRATION TABLES IN─tablespace-name─┘
```

index configuration options:

```
├─┬────────────────────────────────────────────┬──────────────────────────┤
  │                      ┌──────,──────┐         │
  └─INDEX CONFIGURATION──(─▼─(option-value)─┬─)─┘
```

connection options:

```
├─┬───────────────────────────────────────────────────────────┬───────────┤
  └─CONNECT TO─database-name─┬───────────────────────────────────┬─┘
                             └─USER─username─USING─password─┘
```
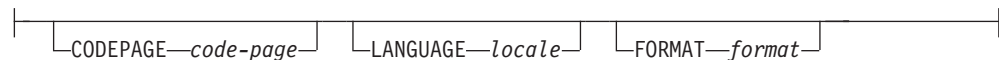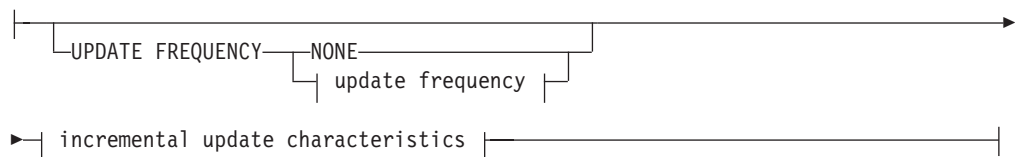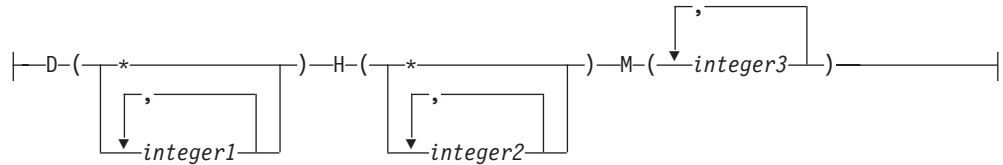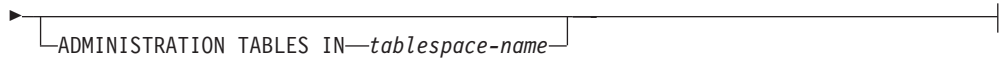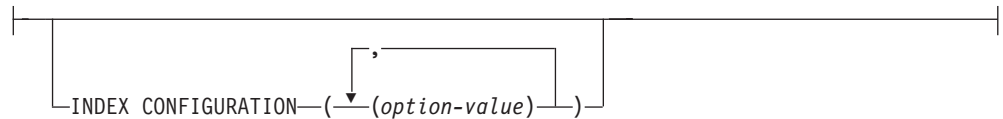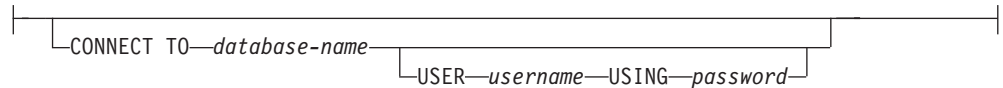
Command parameters

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.

> **Note:** The LONG VARCHAR and LONG VARGRAPHIC data types continue to be supported but are deprecated, not recommended, and might be removed in a future release.

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, and HTML. 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.

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 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. These values must be enclosed in single quotes.

Note: A single quote character within a string value must be represented by two consecutive single quotes. The following values are supported:

Table 7. Specifications for option-value

Option	Allowed values (Default)	Meaning
COMMENT	String value shorter 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.

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 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.
- 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

Related tasks

"Creating a text search index" on page 42
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.

"Issuing text search commands" on page 9
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.

db2ts DROP INDEX

Drops an existing text search index.

For execution, the command needs to be prefixed with db2ts at the command line.

Authorization

One of the following:
- *dbadm* or *sysadm*: This authorization is used to drop a text index as part of the DISABLE DATABASE FOR TEXT command with the FORCE option.
- CONTROL privileges on the table for which the text search index was created must be held by the user with authorization ID that was used to establish the database connection.

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.

Related tasks

"Dropping a text search index" on page 49
When you no longer intend to perform text searches in a text column, you can drop the text search index.

"Issuing text search commands" on page 9
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.

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

CONTROL privileges on the table for which the text search index was created must be held by the user with authorization ID that was used to establish the database connection.

Required connection

Database

Command syntax

```
►►──ALTER INDEX──index-name──FOR TEXT──┤ update characteristics ├────────────►

►──┤ connection options ├───────────────────────────────────────────────────►◄
```

update characteristics:

```
├─────┬──────────────────────────────────────────────────┬──────────────────►
      └─UPDATE FREQUENCY──┬─NONE──────────────────┬─┘
                          └─┤ update frequency ├──┘

►──┬─┤ incremental update characteristics ├─────────────────────┬────────────┤
```

update frequency:

```
                                                            ┌─,──────────┐
                                                            ▼            │
├──D─(──┬─*──────────┬──)──H─(──┬─*──────────┬──)──M─(────integer3──)─────────┤
        │ ┌─,──────┐ │          │ ┌─,──────┐ │
        │ ▼        │ │          │ ▼        │ │
        └───integer1─┘          └───integer2─┘
```

incremental update characteristics:

```
├─────┬────────────────────────────────┬────────────────────────────────────┤
      └─UPDATE MINIMUM──minchanges──────┘
```

connection options:

```
├─────┬──────────────────────────────────────────────────────────┬──────────┤
      └─CONNECT TO──database-name──┬──────────────────────────────┬─┘
                                   └─USER──username──USING──password─┘
```

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.

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.

Related tasks

"Altering a text search index" on page 48
You can alter the update properties of a text search index.

"Issuing text search commands" on page 9
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.

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

CONTROL privileges on the table for which the text search index was created must be held by the user with authorization ID that was used to establish the database connection.

Required connection

Database

Command syntax

```
►►──UPDATE INDEX──index-name──FOR TEXT───────────────────────────────►
                                        └─USING UPDATE MINIMUM─┘

►─┤ connection options ├────────────────────────────────────────────►◄
```

connection options:

```
├──────────────────────────────────────────────────────────────────┤
  └─CONNECT TO──database-name─────────────────────────────┘
                            └─USER──username──USING──password─┘
```

Command parameters

UPDATE INDEX *index-name*
:   Specifies the name of the text search index to be updated. The index name must adhere to the naming restrictions for DB2 indexes.

USING UPDATE MINIMUM
:   Specifies that the UPDATE MINIMUM *minchange* settings, specified in the CREATE INDEX command used to create the index, should be used, and starts an incremental update if the specified minimum number of changes have occurred. By default the update is started unconditionally.

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.

Related tasks

"Issuing text search commands" on page 9
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.

"Updating a text search index" on page 46
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.

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

CONTROL privileges on the table, for which the text search index was created, must be held by the user with authorization ID that was used to establish the database connection
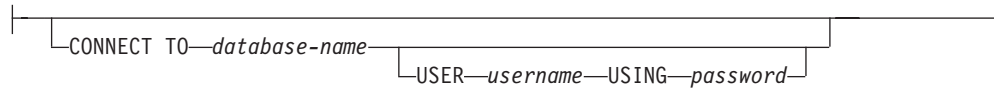
Required connection

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:

```
[A-Za-z][A-Za-z0-9@#$_]*
```

or

```
"[A-Za-z ][A-Za-z0-9@#$_ ]*"
```

When regular updates are scheduled (see UPDATE FREQUENCY options in CREATE INDEX or ALTER INDEX commands), the event table should be regularly checked. To cleanup the DB2 Text Search event table for a text search index, use CLEAR EVENTS FOR INDEX command after you have checked the reason for the event and removed the source of the error.

Be sure to make changes to all rows referenced in the event table. By changing the rows in the user table, you ensure that the next UPDATE INDEX attempt can be made to successfully re-index the once erroneous documents.

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:
- CLEAR EVENTS FOR INDEX
- UPDATE INDEX
- ALTER INDEX
- DROP INDEX

- DISABLE DATABASE FOR TEXT
- STOP FOR TEXT

Changes to the database: The event table is cleared.

Related tasks

"Clearing text search index events" on page 54
If you no longer need the messages in the event view of an index, you can clear (delete) them.

"Issuing text search commands" on page 9
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.

db2ts HELP

HELP displays the list of available DB2 Text Search commands, or the syntax of an individual command.

Use the db2ts HELP command to get help on specific error messages as well.

For execution, the command needs to be prefixed with db2ts at the command line.

Authorization

None.

Command syntax

```
►►─┬─HELP─┬─┬───────────────────┬─────────────────────────────►◄
   └─?────┘ ├─command──────────┤
            ├─sqlcode──────────┤
            ├─sqlstate─────────┤
            └─error-identifier─┘
```

Command parameters

HELP | ?
> Provides help information for a command or a reason code.

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 ? CIE00323
```

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]

Related tasks

"Issuing text search commands" on page 9
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.

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 desired 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 is an example of 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 file.

Related reference

"db2ts DROP INDEX" on page 101
Drops an existing text search index.

"db2ts ALTER INDEX" on page 102
This command changes the update characteristics of an index.

"db2ts UPDATE INDEX" on page 105
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.

"db2ts CLEAR EVENTS FOR INDEX" on page 107
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.

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

EXECUTE privilege on the SYSTS_ENABLE procedure.

The authorization-name for the database connection must have SYSADM authority.

Examples

Example 1: Enable the database for text search and return any error messages in English.

```
CALL SYSPROC.SYSTS_ENABLE('en_US', ?)
```

The following is an example of 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 is an example of 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

EXECUTE privilege on the SYSTS_DISABLE procedure.

The authorization-name for the database connection must have DBADM or SYSADM 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 is an example of 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 prior to 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 is an example of 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 db2diag.log. 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 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 ├─────┤
   ├──┤ update characteristics ├───────┤
   ├──┤ storage options ├──────────────┤
   └──┤ index configuration options ├──┘
```

text default information:
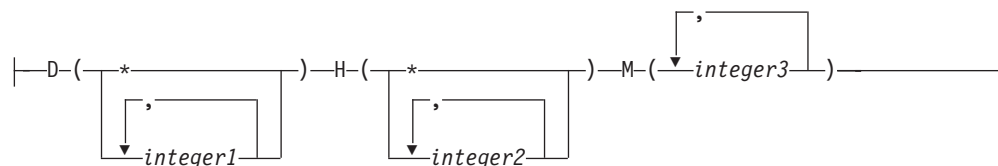
```
├──┬──────────────────────────┬──┬──────────────────────┬──┬──────────────────┬──┤
   └─CODEPAGE──code-page─┘     └─LANGUAGE──language─┘     └─FORMAT──format─┘
```

update characteristics:

```
├──┬────────────────────────────────────────────────────┬──────────────────────►
   └─UPDATE FREQUENCY──┬─NONE──────────────────┬──┘
                       └─│ update frequency │──┘

►──│ incremental update characteristics │──────────────────────────────────────┤
```

update frequency:

```
├──D─(──┬─*────────────┬──)──H─(──┬─*────────────┬──)──M─(──┬─integer3─┬──)──────┤
        │   ┌──,───┐   │          │   ┌──,───┐   │          └────,─────┘
        └──▼─integer1─┘┘          └──▼─integer2─┘┘
```

incremental update characteristics:

```
├──┬────────────────────────────────┬──────────────────────────────────────────┤
   └─UPDATE MINIMUM──minchanges─┘
```

storage options:

```
├──┬──────────────────────────────────────┬────────────────────────────────────►
   └─COLLECTION DIRECTORY──directory─┘

►──┬────────────────────────────────────────────────┬──────────────────────────┤
   └─ADMINISTRATION TABLES IN──tablespace-name─┘
```

index configuration options:

```
├──┬──────────────────────────────────────────────────┬────────────────────────┤
   │                       ┌──,───┐                    │
   └─INDEX CONFIGURATION──(──▼─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.

>>>**Note:** The LONG VARCHAR and LONG VARGRAPHIC data types continue to be supported but are deprecated, not recommended, and might be removed in a future release.
>>>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 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 *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, and HTML. 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.

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 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. These values must be enclosed in single quotes.

> **Note:** A single quote character within a string value must be represented by two consecutive single quotes. The following values are supported:

Table 8. Specifications for option-value

Option	Allowed values (Default)	Meaning
COMMENT	String value shorter 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.

> **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

EXECUTE privilege on the SYSTS_CREATE procedure.

The authorization-name for the database connection must have CONTROL privilege on the table in which the text search index is to be created.

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 is an example of 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 is an example of 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.

  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.
– 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 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 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

EXECUTE privilege on the SYSTS_DROP procedure.

The authorization-name for the database connection must have one of the following:
- CONTROL privileges on the table for which the text search index was created.
- DBADM or SYSADM authority. This authorization is used to drop a text index as part of the DISABLE DATABASE FOR TEXT command with the FORCE option.

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 is an example of 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 is an example of 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.

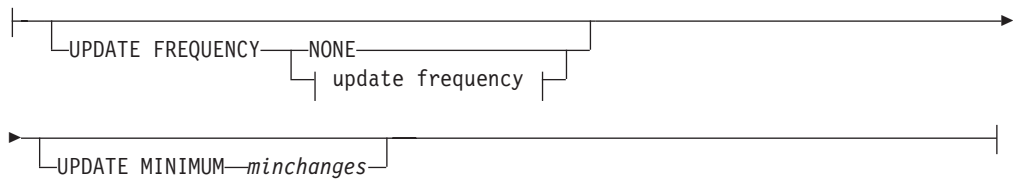# SYSTS_ALTER procedure - Change the update characteristics of an index

This procedure changes the update characteristics of an index.

The procedure issues an ALTER INDEX text search administration command on the database server.
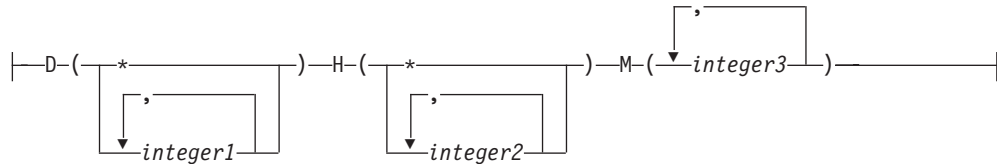
Syntax

```
►►─SYSTS_ALTER─(─index_schema─,─index_name─,─┤ update characteristics ├──────►

►─,─message_locale─,─message─)──────────────────────────────────────►◄
```

update characteristics:

```
├──┬────────────────────────────────────────────────────┬──▶
   └─UPDATE FREQUENCY──┬─NONE────────────────────┬───────┘
                       └─┤ update frequency ├─────┘

▶──┬────────────────────────────────────────────────┬──┤
   └─UPDATE MINIMUM──minchanges─┘
```

update frequency:

```
                                                        ┌─,─┐
├──D─(─┬─*──────────┬─)──H─(─┬─*──────────┬─)──M─(─▼─integer3─┴─)────┤
       │ ┌─,─┐      │        │ ┌─,─┐      │
       └─▼─integer1─┘        └─▼─integer2─┘
```

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.

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

EXECUTE privilege on the SYSTS_ALTER procedure.

The authorization-name for the database connection must have the CONTROL privilege on the table the text search index was created for.

Examples

Example 1: In the following example, the update-characteristics of a text search index is 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 is an example of 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 is an example of 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.

  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.

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.
> * 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

EXECUTE privilege on the SYSTS_UPDATE procedure.

The authorization-name for the database connection must have CONTROL privilege on the table in which the text search index was created.

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 is an example of 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 is an example of 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

EXECUTE privilege on the SYSTS_CLEAR_EVENTS procedure.

The authorization-name for the database connection must have CONTROL privileges on the table for which the text search index was created.

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 is an example of 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 is an example of 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

EXECUTE privilege on the SYSTS_CLEAR_COMMANDLOCKS procedure.

The authorization-name for the database connection must have DBADM or SYSADM authority when *index_name* is not specified as an argument.

The authorization-name for the database connection must have CONTROL privilege on the table for which the text search index was created (according to DB2 catalog views) when *index_name* is specified as an argument.

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 is an example of 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 is an example of 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 wish 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.

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_*nnnnnn*
  - SYSIBMTS.TSSTAGING_*nnnnnn*

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 9. SYSIBMTS.TSDEFAULTS view

Column name	Data type	Nullable?	Description
DEFAULTNAME	VARCHAR (30)	NO	Database default parameters for text search

Table 9. SYSIBMTS.TSDEFAULTS view (continued)

Column name	Data type	Nullable?	Description
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 10. 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 11. 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 12. SYSIBMTS.TSINDEXES view

Column name	Data type	Nullable?	Description
INDSCHEMA	VARCHAR(128)	NO	Schema name for the text search index.
INDNAME	VARCHAR(128)	NO	Unqualified name of the text search index.
TABSCHEMA	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 overhead.
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.

Table 12. SYSIBMTS.TSINDEXES view (continued)

Column name	Data type	Nullable?	Description
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 CREATE INDEX 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 INDSCHEMA=schema-name
        and INDNAME=ind-name and PARAMETER ='parameter'"
```

Table 13. 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..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 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 14. 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: • 1 = Informational • 4 = Parts of the document were indexed but there was a warning, as indicated by the message • 8 = The document was not indexed, as indicated by the message • 0= Otherwise
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
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 STAGINGVIEWSCHEMA 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 STAGINGVIEWSCHEMA, STAGINGVIEWNAME from SYSIBMTS.TSINDEXES
where INDSCHEMA=schema-name and INDNAME=index-name"
```

Table 15. 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.
PK*nn*	Data type of the last primary key column of the base table	YES	Last primary key column of the base table.

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─┘
```
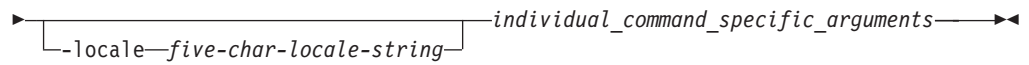
Parameters

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.

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/sqllib/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. The default is en_US.

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.

Tool syntax

```
►►──adminTool──┬─status──-collectionName─collection-name───────────┬──►
               ├─delete──-collectionName─collection-name───────────┤
               ├─version───────────────────────────────────────────┤
               ├─configureTrace──-trace──┬─on──┬────────────────────┤
               │                         └─off─┘                    │
               ├─configureTrace────────────────────────────────────┤
               └─printTrace────────────────────────────────────────┘

►──configPath─absolute-path-to-config-folder──────────────────────────►

►──┬──────────────────────────────────┬──┬──────┬─────────────────────►
   └─-locale─five-char-locale-string──┘  └─help─┘

►──individual_command_specific_argument───────────────────────────────►◄
```
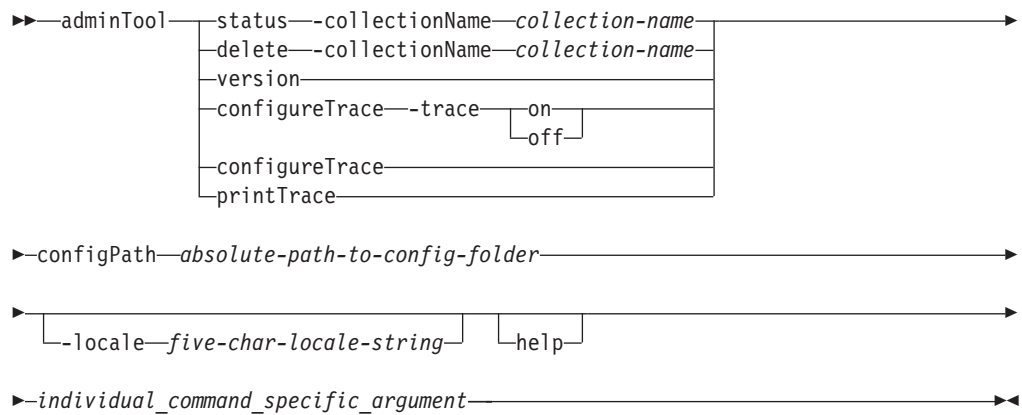
Parameters

delete
> Deletes the text search collection specified by -collectionName.
>
> -collectionName *collection-name*
> Specifies the text search collection to be deleted.

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
> Returns the server version.

configureTrace
> Turns tracing on or off.
>
> -trace
> - on
> - off

printTrace
> Prints the current trace setting.

-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.

Usage

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.

Examples

To print the status of all of the collections using the French locale, issue the following command:

```
adminTool status -configPath c:\Program Files\IBM\Search\config -locale fr_FR
```

To turn on tracing, issue the following command:

```
adminTool configureTrace -configPath /opt/ibm/search/config -trace on
```

Related tasks

"Deleting orphaned text search collections" on page 52
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.

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 16. Administration and environments

Feature or function	DB2 Text Search	DB2 Net Search Extender
The command-line interface	Supported	Supported
The Control Center	Unsupported	Supported
Stored procedures for administration	Supported	Unsupported
External storage of text documents	Unsupported	Supported
Partitioned database environments	Unsupported	Supported on AIX platforms
Error handling	SQL codes, product-specific codes	Product-specific codes
Linguistic processing	In all supported languages	In English only

Table 17. 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 (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.)	Supported	Supported

Table 17. Searching (continued)

Feature or function	DB2 Text Search	DB2 Net Search Extender
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
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 18. 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

Table 18. Indexing and data (continued)

Feature or function	DB2 Text Search	DB2 Net Search Extender
XML namespaces	Not directly supported (must use workaround)	Limited support
Stop word processing	Unsupported	Supported
Document formats	Text, HTML, XML	Text, HTML, XML, General Purpose Format (GPP), proprietary formats (using INSO)

Related concepts

"Net Search Extender key concepts" in Net Search Extender Administration and User's Guide

Chapter 2, "DB2 Text Search overview," on page 3
DB2 Text Search allows you to search text columns stored in a DB2 table. Starting with DB2 Version 9.5 Fix Pack 1, you can enable text search support to use the CONTAINS, SCORE, and xmlcolumn-contains built in functions, which are built into the DB2 engine to search text search indexes based on the search arguments that you specify.

Migration to DB2 Text Search

Although DB2 Text Search provides a rich set of functions and features, it is in its initial stage of integration with DB2 Database for Linux, UNIX, and Windows. Therefore, follow these steps before migrating to DB2 Text Search from Net Search Extender (NSE):

1. Install both NSE and DB2 Text Search with Version 9.5.
2. Continue to use NSE on your production database.
3. Create a test database with the production data.
4. Start the DB2 Text Search instance services.
5. Enable the test database for DB2 Text Search.
6. Map the text search index update requests and search queries for the NSE database to the test database.
7. Study the difference in functionality and performance between NSE and DB2 Text Search.

Note: Migration from NSE to DB2 Text Search cannot be incremental; you must migrate the entire database. This restriction arises because you can enable a database for either NSE or DB2 Text Search, but not both.

If you are satisfied that DB2 Text Search suits your needs better than NSE, disable NSE on your production database. After NSE is disabled, enable your production database for DB2 Text Search, and re-create the text search indexes on it.

Related reference

"DISABLE DATABASE command" in Net Search Extender Administration and User's Guide

Appendix B. Locales supported for DB2 Text Search

The following table lists the locales that DB2 Text Search supports for document processing.

Table 19. 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

Related concepts

"Linguistic processing for DB2 Text Search" on page 8
DB2 Text Searchprovides 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.

Related reference

"SYSIBMTS.TSDEFAULTS view" on page 135
SYSIBMTS.TSDEFAULTS displays all the default values for all text search indexes
in a database.

Appendix C. DB2 commands

db2imigr - Migrate instance

On Linux and UNIX operating systems, migrates an instance from a previous version of DB2 database system (supported for migration to the current version of DB2 database system) to the current version of the DB2 copy from where you are running the db2imigr command. On Windows operating systems, migrates an instance from a past release to the current release.

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.

Note: This command is not available for a non-root installation of DB2 on Linux and UNIX operating systems.

On Windows operating systems, execute the db2imigr command from the DB2 copy that you want to migrate the instance to. 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 migration. This utility is located in the **DB2PATH**\bin directory, where **DB2PATH** is the location where the DB2 copy is installed.
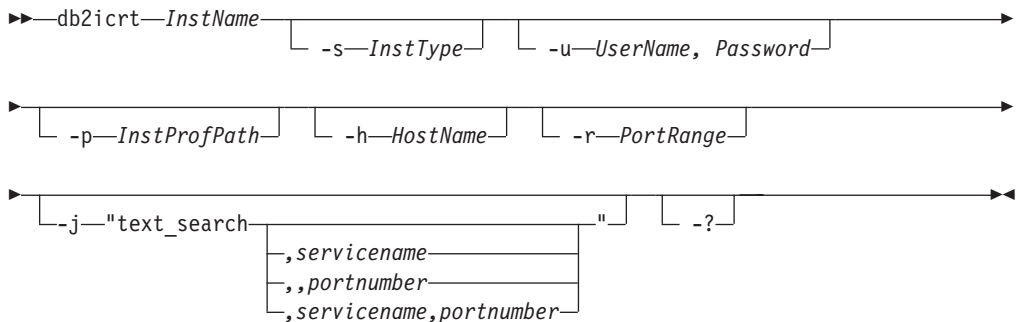
Authorization

Root user authority on Linux and UNIX operating systems or Local Administrator on Windows operating systems.

Command syntax

For Linux and UNIX operating systems

```
►►─db2imigr─┬────┬─┬──────────────┬─┬─────────────┬──InstName─────────►◄
            └─-d─┘ └─-a──AuthType─┘ └─-u──FencedID─┘
```

For Windows operating systems

```
►►─db2imigr──InstName──/u:──username,password─┬───────────────────────────┬─►
                                              └─/p:──instance-profile-path─┘

►─┬─────┬─┬───────────────┬────────────────────────────────────────────────►
  └─/q─┘ └─/a:──authType─┘

►─┬──────────────────────────────────────────────┬─┬─────┬─────────────────►◄
  └─/j──"text_search─┬────────────────────────┬──"┘ └─/?─┘
                     ├─,servicename───────────┤
                     ├─,,portnumber───────────┤
                     └─,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.

-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 migrating 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 migrating a partitioned instance.

/p:*instance-profile-path*
> Specifies the new instance profile path for the migrated instance.

/q   Issues the db2imigr command in quiet mode.

/a:*authType*
> Specifies the authentication type (SERVER, CLIENT, or SERVER_ENCRYPT) for the instance.

/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 it is already reserved in the services file.
>>
>> ```
>> /j "text_search,servicename"
>> ```
>
> *,portnumber*
>> Reserves the port number with the default service name. Valid ports must be within the 1024 - 65535 range.
>>
>> ```
>> /j "text_search,,portnumber"
>> ```

/?   Displays usage information for the db2imigr command.

Usage notes

The db2imigr command calls the db2ckmig command to verify that the databases are ready for migration. The `migration.log` log file is specified for db2ckmig, and the `/tmp/db2imigr.procID` log file is created for db2imigr. The migration will not continue if the db2ckmig command returns any errors.

On partitioned database environments, you can run the db2ckmig command in all database partitions before you issue the db2imigr command. The db2ckmig command only returns errors for the database partition where you issue the db2imigr command. If you do not check whether all database partitions are ready for migration, subsequent database migrations could fail even if the instance migration was successful. See db2ckmig for details.

For Linux and UNIX operating systems
- The db2imigr command removes any symbolic links that exist in /usr/lib and /usr/include in the version you are migrating from. If you have applications that load libdb2 directly from /usr/lib rather than using the operating system's library environment variable to find it, your applications might fail to execute properly after you have run db2imigr.
- If you use the db2imigr command to migrate 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 migrated over to the new installation location. The DB2 Instance Profile Variables specific to the instance to be migrated will be carried over after the instance is migrated.
- 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 db2imigr command when you sourced the DB2 instance environment is not supported.

Related concepts
"Migration for DB2 servers" in Migration Guide
Related tasks
"Migrating instances" in Migration Guide
Related reference
"db2ckmig - Database pre-migration tool " in Command Reference
"dasmigr - Migrate the DB2 administration server " in Command Reference

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.
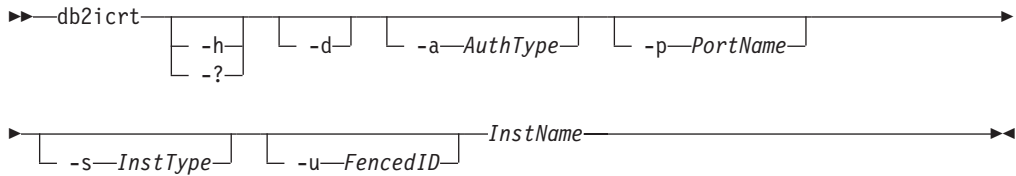
Command syntax

For Linux and UNIX operating systems

```
►►──db2icrt─────────────────────────────────────────────────────────────►
                ┌─-h─┐      ┌─-d─┐   ┌─-a──AuthType─┐   ┌─-p──PortName─┐
                └─-?─┘

►──────────────────────────────InstName──────────────────────────────────►◄
      └─-s──InstType─┘  └─-u──FencedID─┘
```

For Windows operating systems

```
►►──db2icrt──InstName──────────────────────────────────────────────────────►
                     └─-s──InstType─┘  └─-u──UserName, Password─┘

►──────────────────────────────────────────────────────────────────────────►
      └─-p──InstProfPath─┘  └─-h──HostName─┘  └─-r──PortRange─┘

►──────────────────────────────────────────────────────────────────────────►◄
      └─-j──"text_search────────────────────"─┘   └─-?─┘
                          ├─,servicename────────┤
                          ├─,,portnumber────────┤
                          └─,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.

> **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      endport/tcp
```

-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 it is already reserved in the services file.

`-j "text_search,servicename"`

,portnumber

Reserves the port number with the default service name. Valid ports must be within the 1024 - 65535 range.

`-j "text_search,,portnumber"`

-? 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.
- On Linux and UNIX operating systems, the **-s** option is intended for situations in which you want to create an instance that does not use the full functionality 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 db2mscs 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.
- It is possible to create multiple instances on Windows operating systems.
- 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.

Related concepts

"User, user ID and group naming rules" in Data Servers, Databases, and Database Objects Guide

Related reference

"db2isetup - Start instance creation interface " in Command Reference

"db2iclus - Microsoft cluster server " in Command Reference

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──InstName──┬────┬──┬────┬──┬──-h──┬──────────────────────────►◄
                        └─-d─┘  └─-f─┘  └──-?──┘
```

For Windows operating systems

```
►►──db2idrop──InstName──┬────┬──┬────┬───────────────────────────────────►◄
                        └─-f─┘  └─-h─┘
```

Command parameters

For Linux and UNIX operating systems

InstName
> Specifies the name of the instance.

-d   Enters debug mode, for use by DB2 Service.

-f   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

- In a partitioned database environment, if more than one database partition
  belongs to the instance that is being dropped, the db2idrop command has to be
  run on each database partition so that the DB2 registry on each database
  partition is updated.
- Dropping an instance will not drop your databases, so you will not lose data.
  You can drop your instance, recreate it and then catalog the databases to make
  them available. You can use the db2cfexp and db2cfimp commands to export
  and import instance profiles.

- 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*/sqllib/db2tss/config and on Windows: *instance_profile_path**instance_name*\db2tss\config) 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 Windows operating systems, if an instance is clustered with Microsoft Cluster Service (MSCS), then you can uncluster that instance by issuing the db2mscs or db2iclus command prior to dropping the instance.

Related reference

"db2icrt - Create instance " in Command Reference

"db2_deinstall - Uninstall DB2 database products or features " in Command Reference

db2iupdt - Update instances

On Linux and UNIX operating systems, this command updates a specified DB2 instance. On Windows operating systems, it updates the instance release level. It can also be used to move an instance from one DB2 copy to another.

On Linux and UNIX operating systems, the db2iupdt command can be issued against instances of the same version that are associated with the same, or a different DB2 database installation directory. In all cases, it will update the instance so that it runs against the code located in the same directory as where you issued the db2iupdt command. You should issue this command:

- whenever you install a new DB2 database product or feature to the installation directory related to the DB2 instance, except after the installation of a fix pack.
- if you want to bring a DB2 instance from one installation path to the current one for the same version of DB2 database system.

Note: This command is not available for a non-root installation of DB2 database systems on Linux and UNIX operating systems.

On Linux and UNIX operating systems, it is located in the *DB2DIR*/`instance` directory, where *DB2DIR* is the location where the current version of the DB2 database product is installed.

After a fix pack is installed on Linux and UNIX operating systems, the db2iupdt command is executed automatically.

On Windows operating systems, the instance is moved to the DB2 copy you execute db2iupdt from. 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 update. Use the db2imigr command instead to change from a major release to another. This utility is located in the **DB2PATH**\`sqllib`\`bin` directory, where **DB2PATH** is the location where the current version of the DB2 database product is installed.

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──┘
              └─-?─┘

►──┬──────────────────┬──┬──InstName──┬──────────────────────────────────►◄
   └─-u──FencedID──┘     └─-e───────┘
```

For Windows operating systems

```
►►──db2iupdt──InstName──/u:──username,password──┬───────────────────────────────┬──►
                                                └─/p:──instance-profile-path──┘

►──┬──────────────────────────┬──┬────────────────┬──┬────┬──┬────┬──────────────►
   └─/r:──baseport,endport──┘    └─/h:──hostname──┘  └─/s─┘  └─/q─┘

►──┬──────────────────┬──┬──/j──"text_search────────────────────────────"──┬──►
   └─/a:──authType──┘     │                   ┌─,servicename──────────┐     │
                          │                   ├─,,portnumber──────────┤     │
                          │                   └─,servicename,portnumber─┘   │

►──┬──────┬──────────────────────────────────────────────────────────────────►◄
   └─/?─┘
```

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 cannot 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    endport/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"

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 it is already reserved in the services file.
>>
>> /j "text_search,*servicename*"
>
> *,portnumber*
>> Reserves the port number with the default service name. Valid ports must be within the 1024 - 65535 range.
>>
>> /j "text_search,,*portnumber*"

/?   Displays usage information for the db2iupdt command.

Examples (UNIX and Linux)

- An instance, db2inst2, is related to the installation path *DB2DIR1*. You have another installation of the DB2 database product on the same system at *DB2DIR2* for the same version of the DB2 database product as that installed on *DB2DIR1*. To update the instance to use the installed DB2 database product from *DB2DIR1* to *DB2DIR2*, issue the following command:

  DB2DIR2/instance/db2iupdt db2inst2

  If the DB2 database product installed at *DB2DIR2* is at level lower than that at *DB2DIR1*, issue:

  DB2DIR2/instance/db2iupdt -D db2inst2

Usage notes

For UNIX, Linux and Windows operating systems

- If you use the db2iupdt command to update a DB2 instance from another installation location to the current installation location, the DB2 Global Profile Variables defined in an old DB2 database 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

- 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.

Related tasks

"Updating the instance configuration (Linux , UNIX)" in Data Servers, Databases, and Database Objects Guide

"Updating the instance configuration (Windows)" in Data Servers, Databases, and Database Objects Guide

Related reference

"db2ilist - List instances " in Command Reference

"db2isetup - Start instance creation interface " in Command Reference

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 hard-copy 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®.

You can access additional DB2 technical information such as technotes, white papers, and IBM Redbooks® publications online at 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. 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 email 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.

If you would like to help IBM make the IBM Information Management products easier to use, take the Consumability Survey: http://www.ibm.com/software/data/info/consumability-survey/.

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. English and translated DB2 Version 9.5 manuals in PDF format can be downloaded from www.ibm.com/support/docview.wss?uid=swg27009727 and www.ibm.com/support/docview.wss?uid=swg27009728 respectively.

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 in the following table.

Note: The *DB2 Information Center* is updated more frequently than either the PDF or the hard-copy books.

Table 20. DB2 technical information

Name	Form Number	Available in print	Last updated
Administrative API Reference	SC23-5842-03	Yes	December, 2010
Administrative Routines and Views	SC23-5843-03	No	December, 2010
Call Level Interface Guide and Reference, Volume 1	SC23-5844-03	Yes	December, 2010
Call Level Interface Guide and Reference, Volume 2	SC23-5845-03	Yes	December, 2010
Command Reference	SC23-5846-03	Yes	December, 2010
Data Movement Utilities Guide and Reference	SC23-5847-03	Yes	December, 2010
Data Recovery and High Availability Guide and Reference	SC23-5848-03	Yes	December, 2010
Data Servers, Databases, and Database Objects Guide	SC23-5849-03	Yes	December, 2010
Database Security Guide	SC23-5850-03	Yes	December, 2010
Developing ADO.NET and OLE DB Applications	SC23-5851-02	Yes	April, 2009
Developing Embedded SQL Applications	SC23-5852-02	Yes	April, 2009
Developing Java Applications	SC23-5853-03	Yes	December, 2010
Developing Perl and PHP Applications	SC23-5854-02	No	April, 2009
Developing User-defined Routines (SQL and External)	SC23-5855-03	Yes	December, 2010

Table 20. DB2 technical information (continued)

Name	Form Number	Available in print	Last updated
Getting Started with Database Application Development	GC23-5856-03	Yes	December, 2010
Getting Started with DB2 installation and administration on Linux and Windows	GC23-5857-03	Yes	December, 2010
Internationalization Guide	SC23-5858-02	Yes	April, 2009
Message Reference, Volume 1	GI11-7855-00	No	
Message Reference, Volume 2	GI11-7856-00	No	
Migration Guide	GC23-5859-03	Yes	December, 2010
Net Search Extender Administration and User's Guide	SC23-8509-02	Yes	April, 2009
Partitioning and Clustering Guide	SC23-5860-03	Yes	December, 2010
Query Patroller Administration and User's Guide	SC23-8507-01	Yes	April, 2009
Quick Beginnings for IBM Data Server Clients	GC23-5863-03	No	December, 2010
Quick Beginnings for DB2 Servers	GC23-5864-03	Yes	December, 2010
Spatial Extender and Geodetic Data Management Feature User's Guide and Reference	SC23-8508-02	Yes	April, 2009
SQL Reference, Volume 1	SC23-5861-03	Yes	December, 2010
SQL Reference, Volume 2	SC23-5862-03	Yes	December, 2010
System Monitor Guide and Reference	SC23-5865-03	Yes	December, 2010
Text Search Guide	SC23-5866-02	Yes	December, 2010
Troubleshooting Guide	GI11-7857-03	No	December, 2010
Tuning Database Performance	SC23-5867-03	Yes	December, 2010
Visual Explain Tutorial	SC23-5868-00	No	
What's New	SC23-5869-03	Yes	December, 2010
Workload Manager Guide and Reference	SC23-5870-03	Yes	December, 2010
pureXML Guide	SC23-5871-03	Yes	December, 2010
XQuery Reference	SC23-5872-02	No	April, 2009

Table 21. DB2 Connect-specific technical information

Name	Form Number	Available in print	Last updated
Quick Beginnings for DB2 Connect Personal Edition	GC23-5839-03	Yes	December, 2010
Quick Beginnings for DB2 Connect Servers	GC23-5840-03	Yes	December, 2010
DB2 Connect User's Guide	SC23-5841-03	Yes	December, 2010

Table 22. Information Integration technical information

Name	Form Number	Available in print	Last updated
Information Integration: Administration Guide for Federated Systems	SC19-1020-01	Yes	
Information Integration: ASNCLP Program Reference for Replication and Event Publishing	SC19-1018-02	Yes	
Information Integration: Configuration Guide for Federated Data Sources	SC19-1034-01	No	
Information Integration: SQL Replication Guide and Reference	SC19-1030-01	Yes	
Information Integration: Introduction to Replication and Event Publishing	GC19-1028-01	Yes	

Ordering printed DB2 books

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/ v9r5.

To order printed DB2 books:
- 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 Web sites:
     – The IBM directory of world wide contacts at www.ibm.com/planetwide
     – The IBM Publications Web site 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 166.

Displaying SQL state help from the command line processor

DB2 returns an SQLSTATE value for conditions that could be the result of an SQL statement. SQLSTATE help explains the meanings of SQL states and SQL state class codes.

To invoke 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

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

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.

- 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. Clear the browser cache and then 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. Clear the browser cache and then refresh the page to display the DB2 Information Center in your preferred language.

On some browser and operating system combinations, you might have to 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

If you have installed the DB2 Information Center locally, you can obtain and install documentation updates from IBM.

Updating your locally-installed *DB2 Information Center* 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. Non-Administrative and Non-Root *DB2 Information Centers* always run in stand-alone mode. .
2. Use the update feature to see what updates are available. If there are updates that you would like to 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, you have to 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 Vista, the commands listed below must be run as an administrator. To launch a command prompt or graphical tool with full administrator privileges, right-click on the shortcut and then select **Run as administrator**.

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 on **DB2 Information Center** service and select **Stop**.
   - On Linux, enter the following command:
     ```
     /etc/init.d/db2icdv95 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 9.5` 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/V9.5` 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 launches to display the stand-alone Information Center.

3. Click the **Update** button (⬛). On the right hand 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 terminate the processes that were started with the `help_start` batch file. Do not use `Ctrl-C` or any other method to terminate `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 terminate the processes that were started with the `help_start` script. Do not use any other method to terminate the `help_start` script.

7. Restart the *DB2 Information Center*.

   - On Windows, click **Start** → **Control Panel** → **Administrative Tools** → **Services**. Then right-click on **DB2 Information Center** service and select **Start**.
   - On Linux, enter the following command:

     `/etc/init.d/db2icdv95 start`

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 on 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 in using DB2 database products.

DB2 documentation
> Troubleshooting information can be found in the DB2 Troubleshooting Guide or the Database fundamentals section of the DB2 Information Center. There you will find information on how to isolate and identify

problems using DB2 diagnostic tools and utilities, solutions to some of the most common problems, and other advice on how to solve problems you might encounter with your DB2 database products.

DB2 Technical Support Web site

Refer to the DB2 Technical Support Web site 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 DB2 Technical Support Web site at http://www.ibm.com/software/data/db2/support/db2_9/

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.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on 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 give 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 (DBCS) information, contact the IBM Intellectual Property Department in your country/region or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
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 and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This document may provide links or references to non-IBM Web sites and resources. IBM makes no representations, warranties, or other commitments whatsoever about any non-IBM Web sites or third-party resources that may be referenced, accessible from, or linked from this document. A link to a non-IBM Web site does not mean that IBM endorses the content or use of such Web site or

its owner. In addition, IBM is not a party to or responsible for any transactions you may enter into with third parties, even if you learn of such parties (or use a link to such parties) from an IBM site. Accordingly, you acknowledge and agree that IBM is not responsible for the availability of such external sites or resources, and is not responsible or liable for any content, services, products, or other materials on or available from those sites or resources. Any software provided by third parties is subject to the terms and conditions of the license that accompanies that software.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees 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 may contain 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.

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 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.

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 of Sun Microsystems, Inc. in the United States, other countries, 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

W

X

IBM ®

Printed in USA

Spine information:

DB2 Version 9.5 for Linux, UNIX, and Windows

Version 9 Release 5

Text Search Guide

IBM