IBM Text Search for Db2 for z/OS
Version 1 Release 5.0.1
Version

Installation, Administration, and Reference



GC19-4070

Notes

Before using this information and the product it supports, be sure to read the general information under "Notices" at the end of this information.

Subsequent editions of this PDF will not be delivered in IBM Publications Center. Always download the latest edition from .

2020-06-17 edition

This edition applies to IBM Text Search for DB2 for z/OS, Version 1.5.0.1, part of Version 3.1 of DB2 Accessories Suite for z/OS, product number 5697-Q04, and to any subsequent releases to IBM Text Search for DB2 for z/OS until otherwise indicated in new editions. Make sure that you are using the correct edition for the level of the text search offering.

© Copyright International Business Machines Corporation 2007, 2019.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this information..... vii
Who should read this information..... vii
Db2 Utilities Suite for z/OS..... vii
Terminology and citations..... vii
Accessibility features for Db2 11 for z/OS..... viii
How to send your comments..... viii
How to read syntax diagrams..... viii

Chapter 1. IBM Text Search for Db2 for z/OS..... 1
Overview of the text search server..... 2
System requirements for enabling text search support..... 2
Db2 for z/OS migration considerations..... 3
System requirements for installing a text search server..... 3

Chapter 2. Key concepts for IBM Text Search for Db2 for z/OS..... 5
Text search index creation and updates..... 5
Asynchronous indexing and triggers..... 5
Data sharing and text search servers..... 5
Supported document formats..... 6
Document truncation..... 7
Supported data types..... 7
Text score and synonym support..... 7
Linguistic processing..... 8
Supported languages..... 8
Linguistic processing for Chinese, Japanese, and Korean documents..... 9

Chapter 3. User roles..... 11
Db2 subsystem installer..... 11
Db2 database administrator..... 11
Text search server administrator..... 11
User performing search queries..... 12

Chapter 4. Installing and configuring text search functions..... 13
Deploying IBM Text Search for Db2 for z/OS as Docker images..... 13
Installing IBM Text Search for Db2 for z/OS by running silent installation..... 14
Installing fixes on top of an existing text search server..... 17
Resolving an installation failure caused by Security-Enhanced Linux..... 18
Configuring a text search server..... 19
Migrating to a new version of IBM Text Search for Db2 for z/OS..... 20
Migrating when the text search server is stopped..... 20
Migrating when the text search server is running..... 21
Enabling text search support..... 23
Enabling text search support for Db2 11 for z/OS..... 23
Setting up the WLM environment..... 25
Populating the text search administration tables for Db2 for z/OS..... 27
Verifying the installation..... 28
Starting text search functions..... 30
Creating a text search index..... 31
Modifying the options of a text search index..... 31
Updating a text search index..... 32

Searching a text search index....33
Adding a text search server to the system....33

Chapter 5. Administration stored procedures for text search....35
SYSPROC.SYSTS_ALTER....35
SYSPROC.SYSTS_CREATE....41
SYSPROC.SYSTS_DROP....51
Dropping a table without previously calling SYSPROC.SYSTS_DROP....53
SYSPROC.SYSTS_RESTORE....53
SYSPROC.SYSTS_START....55
SYSPROC.SYSTS_STOP....56
SYSPROC.SYSTS_TAKEOVER....57
SYSPROC.SYSTS_UPDATE....59
Scheduling the SYSPROC.SYSTS_UPDATE stored procedure....63
Archive log consumption....63
Responding to an automatic takeover....64
Retrieving messages without truncation....64
Enabling trace for stored procedures....65
Overview of the event table....65
Checking the event table after calling SYSPROC.SYSTS_CREATE....66
Handling single document errors....67
Administering the event table....68

Chapter 6. User-defined functions and search argument syntax....69
CONTAINS....69
Optimizing usage of the equality operator....71
SCORE....72
Optimizing usage of comparison operators....74
SYSFUN.SYSTS_ENCRYPT....74
Search argument syntax....75
Special characters in searches....77
Simple query examples....79
Advanced search operators....79
Restrictions for search argument syntax....83
Example that uses the CONTAINS function and SCORE function....84
Using the RESULTLIMIT option in the CONTAINS and SCORE functions....84
XML search for IBM Text Search for Db2 for z/OS....85
XML search query grammar....90
Namespaces....92
Namespace declarations in a search....93
XML search scenario and query examples....94

Chapter 7. Administering a text search server....99
Starting a text search server....99
Stopping a text search server....99
Adding a synonym dictionary to a collection....99
Removing a synonym dictionary from a collection....100
Uninstalling a text search server....101
Command-line tools....102
Configuration Tool....102
Administration Tool....104
Synonym Tool....105

Chapter 8. Text search administration tables....107
SYSIBMTS.SYSTEXTDEFAULTS administration table....107
SYSIBMTS.SYSTEXTINDEXES administration table....107
SYSIBMTS.SYSTEXTCOLUMNS administration table....109

SYSIBMTS.SYSTEXTSERVERS administration table..... 110
SYSIBMTS.SYSTEXTCONNECTINFO administration table.....111
SYSIBMTS.SYSTEXTSTATUS administration table.....111
SYSIBMTS.SYSTEXTCONFIGURATION administration table..... 112
SYSIBMTS.SYSTEXTLOCKS administration table.....112
SYSIBMTS.SYSTEXTSERVERHISTORY administration table..... 112

Chapter 9. Troubleshooting.....115
Supporting concurrent index updates and search requests..... 115
Preventing a timeout abend..... 115
Troubleshooting invalid entries in administration tables.....115
Troubleshooting SQL code -430..... 116
Troubleshooting SQL code -20212.....117
Troubleshooting SQL code -20423 with message OF00801E.....118

Chapter 10. Messages and codes.....119
Text search server messages.....119

Information resources for Db2 11 for z/OS and related products..... 153

Notices.....155
Trademarks.....156
Privacy policy considerations.....156

Glossary.....159

Index..... 161

About this information

Who should read this information

This information is primarily intended for people who are responsible for creating and maintaining full-text indexes on character-based columns in Db2® tables.

Db2 Utilities Suite for z/OS

Important: In Db2 11, the Db2 Utilities Suite for z/OS® is available as an optional product. You must separately order and purchase a license to such utilities, and discussion of those utility functions in this publication is not intended to otherwise imply that you have a license to them.

Db2 11 utilities can use the DFSORT program regardless of whether you purchased a license for DFSORT on your system. For more information, see the following informational APARs:

- II14047
- II14213
- II13495

Db2 utilities can use IBM® Db2 Sort for z/OS (5655-W42) as an alternative to DFSORT for utility SORT and MERGE functions. Use of Db2 Sort for z/OS requires the purchase of a Db2 Sort for z/OS license. For more information about Db2 Sort for z/OS, see Db2 Sort for z/OS.

Related information

Terminology and citations

When referring to a Db2 product other than Db2 for z/OS, this information uses the product's full name to avoid ambiguity.

About the Db2 brand change: IBM has re-branded DB2® to Db2, and Db2 for z/OS is now the name of the offering previously know as "DB2 for z/OS". During and after implementation of this change, you might sometimes still see references to the original names, such as "DB2 for z/OS" and "DB2", in different IBM web pages and documents. However, assume that these refer to the same "Db2 for z/OS" product when the PID, Entitlement Entity, version, modification, and release information match. For more information, see Revised naming for IBM Db2 family products on IBM z/OS platform.

The following terms are used as indicated:

Db2
Represents either the Db2 licensed program or a particular Db2 subsystem.

Tivoli® OMEGAMON® XE for Db2 Performance Expert on z/OS
Refers to any of the following products:

- IBM Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS
- IBM Db2 Performance Monitor on z/OS
- IBM Db2 Performance Expert for Multiplatforms and Workgroups
- IBM Db2 Buffer Pool Analyzer for z/OS

C, C++, and C language
Represent the C or C++ programming language.

CICS®
Represents CICS Transaction Server for z/OS.

© Copyright IBM Corp. 2007, 2019

IMS
Represents the IMS Database Manager or IMS Transaction Manager.

MVS™
Represents the MVS element of the z/OS operating system, which is equivalent to the Base Control Program (BCP) component of the z/OS operating system.

RACF®
Represents the functions that are provided by the RACF component of the z/OS Security Server.

Accessibility features for Db2 11 for z/OS

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use information technology products successfully.

Accessibility features

The following list includes the major accessibility features in z/OS products, including Db2 11 for z/OS. These features support:

- Keyboard-only operation.
- Interfaces that are commonly used by screen readers and screen magnifiers.
- Customization of display attributes such as color, contrast, and font size

Tip: The IBM Knowledge Center (which includes information for Db2 for z/OS) and its related publications are accessibility-enabled for the IBM Home Page Reader. You can operate all features using the keyboard instead of the mouse.

Keyboard navigation

For information about navigating the Db2 for z/OS ISPF panels using TSO/E or ISPF, refer to the *z/OS TSO/E Primer*, the *z/OS TSO/E User's Guide*, and the *z/OS ISPF User's Guide*. These guides describe how to navigate each interface, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

Related accessibility information

IBM and accessibility

See the *IBM Accessibility Center* at http://www.ibm.com/able for more information about the commitment that IBM has to accessibility.

How to send your comments

Your feedback helps IBM to provide quality information. Please send any comments that you have about this book or other Db2 for z/OS documentation.

Send your comments by email to db2zinfo@us.ibm.com and include the name of the product, the version number of the product, and the number of the book. If you are commenting on specific text, please list the location of the text (for example, a chapter and section title or a help topic title).

How to read syntax diagrams

Certain conventions apply to the syntax diagrams that are used in IBM documentation.

Apply the following rules when reading the syntax diagrams that are used in Db2 for z/OS documentation:

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

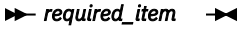
  The ►►—— symbol indicates the beginning of a statement.

  The ——► symbol indicates that the statement syntax is continued on the next line.

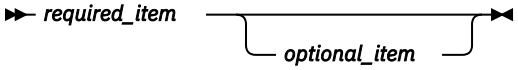
The ►—— symbol indicates that a statement is continued from the previous line.

The ——►◄ symbol indicates the end of a statement.

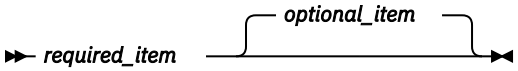
- Required items appear on the horizontal line (the main path).



- Optional items appear below the main path.

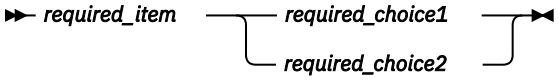


If an optional item appears above the main path, that item has no effect on the execution of the statement and is used only for readability.

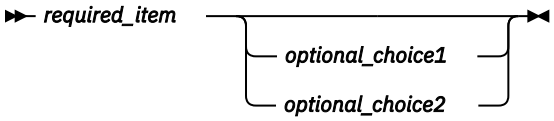


- If you can choose from two or more items, they appear vertically, in a stack.

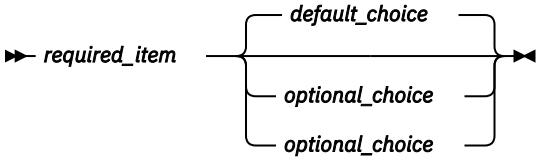
  If you *must* choose one of the items, one item of the stack appears on the main path.



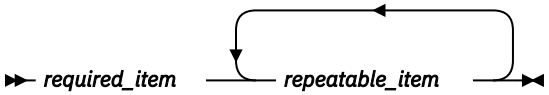
If choosing one of the items is optional, the entire stack appears below the main path.



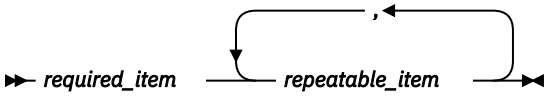
If one of the items is the default, it appears above the main path and the remaining choices are shown below.



- An arrow returning to the left, above the main line, indicates an item that can be repeated.



If the repeat arrow contains a comma, you must separate repeated items with a comma.

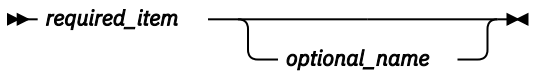


A repeat arrow above a stack indicates that you can repeat the items in the stack.

- Sometimes a diagram must be split into fragments. The syntax fragment is shown separately from the main syntax diagram, but the contents of the fragment should be read as if they are on the main path of the diagram.



fragment-name



- With the exception of XPath keywords, keywords appear in uppercase (for example, FROM). Keywords must be spelled exactly as shown. XPath keywords are defined as lowercase names, and must be spelled exactly as shown. Variables appear in all lowercase letters (for example, *column-name*). They represent user-supplied names or values.
- If punctuation marks, parentheses, arithmetic operators, or other such symbols are shown, you must enter them as part of the syntax.

Related information

Chapter 1. IBM Text Search for Db2 for z/OS

The IBM Text Search for Db2 for z/OS enables a Db2 for z/OS user to issue SQL statements to satisfy familiar text search queries on data that is stored in a Db2 database.

The text search feature for Db2 for z/OS is powered by a text search server. IBM Text Search for Db2 for z/OS provides text search capability for text columns that are stored in a Db2 table. This text search server offers excellent query performance and scalability by integrating optimization and state-of-the-art search technologies.

To use the text search feature, you enable support on a Db2 for z/OS subsystem. By enabling text search support, you can use the CONTAINS function and the SCORE function, which are built into the Db2 engine, to search text search indexes based on the search argument criteria that you specify.

You install a text search server on a supported Linux®, zLinux, or Windows server. Db2 for z/OS uses a TCP/IP connection to communicate with the text search server.

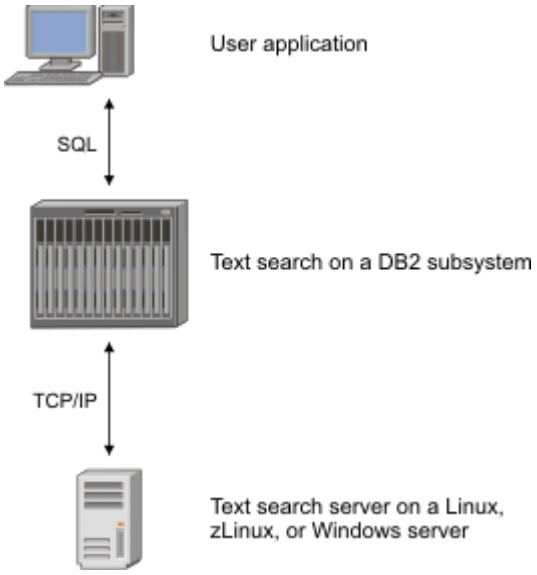


Figure 1. The communication method between the Db2 subsystem and the text search server

IBM Text Search for Db2 for z/OS supports multiple collections. A *collection* contains one text search index and the index specific options for parsing, indexing, and searching.

IBM Text Search for Db2 for z/OS does not have a graphical user interface. However, the text search server provides command-line tools that you can use for common tasks, such as configuring and administering the text search server, creating a synonym dictionary for a collection, and diagnosing problems.

A set of administration stored procedures is provided by Db2 for z/OS to start and stop text search support and to create, update, drop, restore, and take over text search indexes. For example, in the case of a planned outage of a text search server, you can use the stored procedure for restoring a text search index to copy that text search index to another text search server, where you can use the text search index during the outage. These stored procedures are provided with Db2 for z/OS and are defined during installation.

Although the text search index is maintained by a text search server on its local disk, the master copy of the text search index is stored in the Db2 tables. You can access the text search index from all Db2 subsystems in a data sharing group. All text search servers that are configured for a particular Db2 subsystem or data sharing group can access the master copy of the text search index.

© Copyright IBM Corp. 2007, 2019

Overview of the text search server

Db2 for z/OS uses IBM Text Search for Db2 for z/OS as an indexing and search engine for documents that are stored in a Db2 database.

IBM Text Search for Db2 for z/OS supports multiple collections. A *collection* contains one text search index and the index specific options for parsing, indexing, and searching.

IBM Text Search for Db2 for z/OS does not have a graphical user interface. However, the text search server provides command-line tools that you can use for common tasks, such as configuring and administering the text search server, creating a synonym dictionary for a collection, and diagnosing problems.

System requirements for enabling text search support

Before you enable text search support, ensure that your system meets all of the operating system and software requirements.

Operating system requirements

Your z/OS operating system must meet the following requirements:

- For Db2 11, your z/OS operating system must meet the requirements for z/OS Version 2 Release 1, or later.
- The APF-authorized Language Environment® facility must be enabled for use in the DBM1 address space. You must install part of the APF-authorized Language Environment in the LPA. This module is installed in SYS1.LPALIB. The remaining APF-authorized Language Environment parts are shipped in SCEERUN2, along with the 64-bit Language Environment runtime parts.
- For Db2 to use the APF-authorized Language Environment library, the Language Environment data sets SCEERUN2 and SCEERUN must be defined as authorized libraries, and the libraries must be part of the z/OS program search order for the DBM1 started task. To meet these requirements, you can do one of the following tasks:
  - Put SCEERUN2 and SCEERUN in the LNKLST.
  - Define SCEERUN2 and SCEERUN to be APF-authorized, and place them in the STEPLIB/JOBLIB concatenation for the DBM1 address space.
- To use the text search feature for Db2 for z/OS, an OMVS segment must be defined for the user ID of the DBM1 address space started task.

Software requirements

You must install or migrate to Db2 11 in new-function mode.

In addition, you must install the following components of Db2 Accessories Suite for z/OS, Version 4.1.0 or later:

- IBM International Components for Unicode (ICU) for Db2 for z/OS
- IBM Text Search for Db2 for z/OS

To download the most current version of the text search server, see IBM Text Search for Db2 for z/OS downloads.

For more information about enabling text search functions, see the Program Directory for Db2 Accessories Suite for z/OS. The Program Directory provides instructions for the SMP/E installation and includes the correct service levels for all of the required software.

Db2 for z/OS migration considerations

Db2 for z/OS does not provide a migration path to convert indexes from existing text search solutions.

After migrating from one Db2 for z/OS release to another, you can still use existing CONTAINS or SCORE user-defined functions by explicitly specifying their schema qualifiers.

Important: Before you migrate from Db2 10 to Db2 11, you must apply APAR PM84540 and complete the instructions that are provided in the ++HOLD action.

System requirements for installing a text search server

Before you install IBM Text Search for Db2 for z/OS, make sure that your system meets all of the hardware, software, and operating system requirements.

Hardware requirements

You can install a text search server on a supported Linux, zLinux, or Windows server. Db2 for z/OS uses a TCP/IP connection to communicate with the text search server.

Make sure that your server meets the following minimum hardware requirements.

For Intel Linux or Windows servers:

- Two dual-core 2.66 GHz processors
- 4 GB RAM

For zLinux servers:

- Two processors
- 8 GB RAM

You need enough disk space to hold four times the original text size multiplied by the number of text search indexes. Each text column in a Db2 table has a corresponding text search index on the text search server. For example, if a text search server supports five indexes each with 1 million rows of 1 KB text size, the required disk space needed is about 20 GB.

Designate that this hardware is used exclusively by the text search server.

Operating system requirements for IBM Text Search for Db2 for z/OS

You can install a text search server for IBM Text Search for Db2 for z/OS on the following operating systems:

- 64-bit Red Hat Enterprise Linux, Version 5 with the latest update
- 64-bit SUSE Linux Enterprise 10
- 64-bit SUSE zLinux SLES10x
- 64-bit Red Hat zLinux, Version 5 with the latest update
- 64-bit Windows Server with the latest service pack
- 32-bit Red Hat Enterprise Linux, Version 5 with the latest update
- 32-bit SUSE Linux Enterprise 10
- 32-bit Windows Server 2008 with the latest service pack

Note: If you install more than one text search server, for fail over support you must install all of the servers on the same type of hardware, and make sure that each server uses the same operating system.

Operating system requirements for deploying text search server Docker images

You can also deploy a text search server for IBM Text Search for Db2 for z/OS as a Docker image on the following operating system platforms (64-bit only):

- SLES 12.4

- Redhat 7.4
- zLinux Redhat 7.6
- Windows 10

Docker Engine must be installed on the server where you plan to install IBM Text Search for Db2 for z/OS.

Related information

Chapter 2. Key concepts for IBM Text Search for Db2 for z/OS

Understanding the key concepts about text search functions, such as what document types and languages are supported, will help you to leverage the benefits.

Text search index creation and updates

Text search index creation is the process of defining and declaring the properties of the text search index. *Text search index update* is the process of adding new data from a Db2 table to the text search index on the text search server and propagating any changes of the data.

For each text search index that you create, a new collection is created on the text search server. After initial creation, the text search index contains no data. You add data to the text search index by running a program that calls the SYSPROC.SYSTS_UPDATE stored procedure. The first update process adds all of the text documents from the text column to the text search index. This process is known as the *initial update*. The subsequent updates are incremental.

Asynchronous indexing and triggers

Because updating a text search index is an extensive operation, the text search index that is maintained on the text search server is not updated synchronously when the Db2 table is updated.

Instead, changes to the Db2 table column are captured by triggers to a local log table. (Sometimes a log table might be referred to as a staging table.) These triggers automatically store information about new, changed, and deleted documents in a log table. One log table is associated with each text search index. Applying the contents of the log table to its corresponding text search index is called *incremental update*.

You can update the text search index that is on the text search server by invoking the SYSPROC.SYSTS_UPDATE stored procedure. You must periodically update the text search index in order for changes to be reflected in queries.

Data sharing and text search servers

The text search feature is fully supported in a Db2 for z/OS data sharing Parallel Sysplex® environment.

All of the administration data is stored in Db2, so the data is available to all data sharing group members. If you use data sharing, you can run the administration stored procedures on any subsystem in the Db2 data sharing group.

You can define multiple text search servers. By defining more than one text search server, you prevent one server from becoming a single point of failure.

Each text index is assigned to a server that works with the latest copy of the text index from Db2. The administration stored procedures that contact a text search server can detect if a server is no longer available for creating, updating, and searching indexes. If a server is not available, a console message is written. You can run the SYSPROC.SYSTS_TAKEOVER stored procedure to verify that a server is not available. When a server isn't available, the SYSPROC.SYSTS_TAKEOVER stored procedure selects a different server and restores the index to that server.

If the SYSPROC.SYSTS_UPDATE stored procedure is running and detects that a server is not available, this stored procedure automatically checks for the availability of another server and restores the index to that server. The next call to the SYSPROC.SYSTS_UPDATE stored procedure will perform the index update on the new server.

You can run the SYSPROC.SYSTS_RESTORE stored procedure to copy an index to another text search server. By using this stored procedure, the original text index can be searched while the stored procedure is running (for example, if you are preparing for a planned outage of a text search server).

© Copyright IBM Corp. 2007, 2019

Supported document formats

The text column data can be plain text, an HTML document, an XML document, or any document that is recognized by the search engine.

A text search server parses documents to extract the relevant parts and make those parts searchable. For example, tags and metadata in an HTML document are not indexed. Parsing of the following document formats is supported:

- TEXT: Flat text
- HTML: Hypertext Markup Language
- XML: Extensible Markup Language
- INSO: The search engine uses filters to detect the format of text documents. The following table lists the file types that are supported.

Table 1. Supported document types

Document types	Typical file extensions
Lotus® 1-2-3 and Lotus 1-2-3 for SmartSuite®	123
Microsoft Word 97 and later	DOC
Hypertext Markup Language	HTML
JustSystems Ichitaro	JTD, JTT, JFW, JVW
Lotus Word Pro®	LWP
Microsoft Project	MPP
Portable Document Format	PDF
Microsoft PowerPoint	PPT
Lotus Freelance	PRZ
Quattro Pro	QPW
Microsoft Rich Text Format	RTF
StarOffice Calc and OpenOffice Calc	SXC
StarOffice Impress and OpenOffice Impress	SXI
StarOffice Writer and OpenOffice Writer	SXW
Microsoft WordPad (File types can vary.)	TXT
Microsoft Visio	VSD
Microsoft Write	WRI
WordStar and WordStar 2000	WS
Microsoft Excel	XLS

All of the documents in an indexed text column must be of the same format (TEXT, HTML, XML or INSO).

XML data

XML structure in the XML data is indexed in the text search server after parsing the data through an XML parser. Then, you can use the supported XPath query syntax to retrieve the results.

Document truncation

A text search server limits the number of Unicode characters that can be indexed 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 that are allowed for each text document is 10 million. For a rich text document, this limit is applied after the document is transformed to plain text.

If a text document is truncated during the parsing stage, you receive a warning that some documents were not processed correctly or completely, and the document is partially indexed. Details about the warning are written to the event table that is created for the text search index. Text that is in the document after the limit is reached is not indexed and cannot be searched.

Supported data types

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

The following data types are binary data:

- BINARY
- VARBINARY
- BLOB

In addition, the text search feature for Db2 for z/OS handles the following data types similarly to binary data:

- CHAR FOR BIT DATA
- VARCHAR FOR BIT DATA

The following data types are character data:

- CHAR FOR SBCS DATA or FOR MIXED DATA
- VARCHAR FOR SBCS DATA or FOR MIXED DATA
- CLOB
- DBCLOB
- GRAPHIC
- VARGRAPHIC

If the data is binary data, you can specify the coded character set identifier (CCSID) of the data that is to be used to build the text search index. For character data, Db2 knows the encoding; therefore, if you explicitly specify a CCSID, that specification is ignored.

All of the CCSIDs that are supported for conversion to UTF-8 by z/OS Unicode Conversion Services are supported by the text search feature for Db2 for z/OS. To display active conversions to UTF-8 on your system, use command `D UNI,CONV,TOID=1208`.

Related concepts

z/OS Support for Unicode: Using Unicode Services

Text score and synonym support

A text search server supports the use of synonyms to improve the results of a query. In addition, text score might be returned as part of the query results.

Text score

Db2 might request that a text score be returned as part of the query results. A text score is a value between 0 (zero) and 1 up to 3 decimal points (for example, 0.000 to 1.000). A text score denotes how closely a result matches the query relative to all of the other documents in the text search index.

A text score is composed from various factors. These factors include the general importance of the search terms (based on the frequency of the terms in each document and offset by the frequency of the terms across all documents) and the proximity of occurrences of the search terms. By default, a text score is not returned.

Synonym support

You can use synonyms to improve the results for a query. Using synonyms might increase the number of query results by causing more documents to match a query. However, using synonyms might also decrease the precision of a query and make it more difficult to find a small number of documents that match the exact search criteria.

By default, synonyms are not used for a query. To use synonyms for a query, create a synonym dictionary, and add the synonym dictionary to a collection by using the Synonym Tool.

Related reference

"Synonym Tool" on page 105
Use the Synonym Tool to add a synonym dictionary to a collection to improve the quality of search results, or to remove synonyms from a collection.

Linguistic processing

A text search server provides dictionary packs to support the linguistic processing of documents and queries that are not in English.

As an alternative to *dictionary-based word segmentation*, which uses a language-specific dictionary to identify words in the sequence of characters in the document, the search engine uses n-gram segmentation to support the linguistic processing of Chinese, Japanese, and Korean. *N-gram segmentation* is a method of analysis that considers overlapping sequences of a given number of characters as a single word rather than using blank space to delimit words.

If a text document is in one of the supported languages, linguistic processing is carried out when the input is parsed into tokens (which means when the text is separated into individual words).

For unsupported languages, the document is parsed by using blank space or n-gram segmentation, depending on the script of each character. If the script is used by a language that usually does not use blank space as a word delimiter, such as Han, Hiragana, Hangul, then the document is parsed by n-gram segmentation. Otherwise, the document is parsed by using blank space. *Lemmatization*, a process that identifies the root form and different grammatical forms of a word, is not performed on unsupported languages.

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

Supported languages

You can specify that text documents be processed by using a specific language. The text search feature supports the linguistic processing of text documents by 26 different language codes.

You can specify the language for the indexed text data in the SYSPROC.SYSTS_CREATE administration stored procedure. If you set the value to AUTO, the text search server tries to determine the language.

Automatic language detection is more accurate for longer documents. For very short documents that consist of just a few words, automatic language detection is not recommended. The default language for linguistic processing is English (en_US).

The following table shows the five-character language codes for the supported languages.

Table 2. The five-character language codes for the supported languages

Language code	Language
ar_AA	Arabic

Table 2. The five-character language codes for the supported languages (continued)

Language code	Language
cs_CZ	Czech
da_DK	Danish
de_CH	German (Switzerland)
de_DE	German (Germany)
el_GR	Greek
en_AU	English (Australia)
en_GB	English (United Kingdom)
en_US	English (United States)
es_ES	Spanish (Spain)
fi_FI	Finnish
fr_CA	French (Canada)
fr_FR	French (France)
it_IT	Italian
ja_JP	Japanese
ko_KR	Korean
nb_NO	Norwegian Bokmal
nl_NL	Dutch
nn_NO	Norwegian Nynorsk
pl_PL	Polish
pt_BR	Brazilian Portuguese
pt_PT	Portuguese (Portugal)
ru_RU	Russian
sv_SE	Swedish
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Linguistic processing for Chinese, Japanese, and Korean documents

You can process documents that are in Chinese, Japanese, or Korean by 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.

A text search server provides n-gram segmentation to support the linguistic processing of Chinese, Japanese, and Korean, and by default, comes with a pre-configured index.

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.

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 is `election for governor`, which is four characters and is represented as E F G H. In this case, 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.

When n-gram segmentation is used, you will see more results, but the results might be less precise. 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 for this result is that `City of Tokyo` and `Kyoto` share two of the same Japanese characters.

Note: Wildcard card characters are ignored when processing Chinese, Japanese, and Korean documents.

Chapter 3. User roles

Using the text search feature for Db2 for z/OS requires several user roles, and sometimes different authorizations are required for the various users.

Typical users include:

- Db2 subsystem installers
- Db2 database administrators
- Text search server administrators
- Users who perform text search queries

Db2 subsystem installer

A user with the role of Db2 subsystem installer runs the installation JCL that creates the Db2-supplied stored procedures and tables and that enables text search support.

To bind the stored procedure packages, the user ID for this role must have table privileges (SELECT, INSERT, UPDATE, and DELETE privileges) on the text search administration tables and SELECT privileges on the Db2 catalog tables.

Db2 database administrator

A Db2 for z/OS database administrator can start and stop text search support on a Db2 subsystem, and create, drop, and change indexes.

This role corresponds to the Db2 DBADM authorization level, which allows the manipulation of all database objects.

The main tasks of a database administrator include:

- Starting Db2 text search functions
- Stopping Db2 text search functions
- Creating tables
- Deleting tables
- Creating an index, and defining and altering its characteristics
- Updating a full-text index
- Dropping an index
- Granting the required privileges to text search users
- Clearing the event table periodically
- Maintaining the text search server information in the SYSIBMTS.SYSTEXTSERVERS administration table
- Maintaining the Db2 for z/OS connection information in the SYSIBMTS.SYSTEXTCONNECTINFO administration table

Text search server administrator

The text search server administrator installs one or more text search servers.

You can install a text search server on a Linux, zLinux, or Windows server. Db2 uses a TCP/IP connection to communicate with the text search server.

Related tasks

“Installing IBM Text Search for Db2 for z/OS by running silent installation” on page 14

© Copyright IBM Corp. 2007, 2019

The text search server administrator installs one or more text search servers. IBM Text Search for Db2 for z/OS uses a silent installation that takes input values from a response file.

User performing search queries

The user who performs search queries uses the CONTAINS and SCORE functions in an SQL query against a Db2 column that has a text search index on it.

Depending on the access rights that users are granted on the table that the index is created on, the SQL query is permitted or rejected by Db2. Specific authorization for the text search feature is not required.

The user who performs search queries can include the following functionality in SQL queries:

- Limit the text search based on the criteria that is specified by the search argument
- Return a score indicating how well a document ranks with respect to the search argument

Chapter 4. Installing and configuring text search functions

Before you can start using text search functions, you need to enable and configure text search support on a Db2 subsystem, and install and configure a text search server.

About this task

You can complete some of the steps for installing and configuring text search functions in a different order than is specified in this information. For example, you can choose to populate the SYSIBMTS.SYSTEXTCONNECTINFO administration table with the Db2 for z/OS connection information before populating the server information columns in the SYSIBMTS.SYSTEXTSERVERS administration table.

Procedure

To install and configure text search functions:

1. Install a text search server on either a Linux, zLinux, or Windows server.

   You can install the server by using the installation files, or on certain platforms by deploying a Docker image.
2. Configure the text search server by using the Configuration Tool.
3. Start text search functions by running a program that calls the SYSPROC.SYSTS_START stored procedure.
4. On your Db2 subsystem, install all of the APARs and PTFs for enabling text search support.
5. Enable text search support for Db2 for z/OS by completing the configuration steps.
6. Set up the WLM environment for running the administration stored procedures and the user-defined function.
7. Populate the text search administration tables.
8. Verify the installation.

Deploying IBM Text Search for Db2 for z/OS as Docker images

You can deploy Docker images to install IBM Text Search for Db2 for z/OS servers.

Before you begin

Docker Engine must be installed on the server where you plan to deploy the text search server for IBM Text Search for Db2 for z/OS.

About this task

You can install a text search server with the Docker images on the following platforms (64-bit only):

- SLES 12.4
- Redhat 7.4
- zLinux Redhat 7.6
- Windows 10

Tip: Ensure that the text search server automatically starts each time that the operating system reboots by installing the text search server as a service. In service mode, the silent installation includes the scripts for starting and stopping the text search server as part of the operating system's start up and shutdown processes. To install a text search server as a service, the user ID installing the product must be a root user on the Linux or zLinux server, or an administrator on the Windows server. If a user ID

© Copyright IBM Corp. 2007, 2019

without the correct access installs a text search server, you can use the start and stop scripts each time that you want to start or stop the server.

Procedure

1. Download the Docker image from IBM Text Search for Db2 for z/OS downloads.
   The following table shows the file names of the Docker images for the supported server platforms.

   Table 3. IBM Text Search for Db2 for z/OS Docker images by platform

Server platform (64-bit only)	File name
zLinux Rehat 7.6	`tsserver_v1_linux.tar`
Linux, including SLES 12.4 and Redhat 7.4	`tsserver_v1_zlinux.tar`
Windows 10	`tsserver_v1_windows.tar`

2. Open a command window, go to the download folder, and run the following commands.

```
# docker load -i  tsserver_v1_platform.tar
# docker run -d tsserver:v1
# docker ps
```

   The output contains the `CONTAINER ID` value.

```
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS              PORTS               NAMES
6fa66d5e8fdb        tsserver:v1         "/bin/sh -c 'bin/s..."           11 days
ago         Up 11 days          9000/tcp
```

3. Run the following commands to get the configuration information, whether *container-id* is the `CONTAINER ID`.

```
 # docker logs container-id
```

   The result is similar to the following output.

```
WARNING: The number of files per process value was not set to unlimited or to 65536 as
required. The startup script changed this value to 65536.
IQQG0337I The server started successfully. The server is listening on port 9000.
The authentication token is printed below. This token is used to communicate with the
server. Store the token if applicable.
EPhhaz0=
The encryption key is printed below. This key is used to encrypt the password during text
index backup and restore operations. Store the key if applicable.
EPhhaz2txoqZbGLjTb7Vfg==
```

Related tasks

“Configuring a text search server” on page 19
After installing a text search server, use the Configuration Tool to customize some of the default properties.

“Resolving an installation failure caused by Security-Enhanced Linux” on page 18
Security-Enhanced Linux (SELinux) can cause the installation of a text search server on Linux to fail.

Installing IBM Text Search for Db2 for z/OS by running silent installation

The text search server administrator installs one or more text search servers. IBM Text Search for Db2 for z/OS uses a silent installation that takes input values from a response file.

Before you begin

Before running the silent installation, verify that the values in the response file are correct.

Tip: As an alternative to this procedure, you can deploy a text search server as a Docker image on certain platforms. For details and instructions, see .

About this task

You can install a text search server on a Linux, Windows, or zLinux server. For detailed requirements, see .

You can also install fixes without previously installing a text search server from the product DVD or product website. Fixes provide a complete installation of the product.

Tip: Ensure that the text search server automatically starts each time that the operating system reboots by installing the text search server as a service. In service mode, the silent installation includes the scripts for starting and stopping the text search server as part of the operating system's start up and shutdown processes. To install a text search server as a service, the user ID installing the product must be a root user on the Linux or zLinux server, or an administrator on the Windows server. If a user ID without the correct access installs a text search server, you can use the start and stop scripts each time that you want to start or stop the server.

Procedure

To install a text search server, complete the following steps:

1. Create an installation directory, which is referred to as USER_INSTALL_DIR in these instructions.

   The directory must be empty.

   For example, specify `/IBM/TextSearch_DB2` Linux or zLinux systems, or specify `C:\Program Files\IBM\TextSearch_DB2` For Windows servers. The USER_INSTALL_DIR directory must be empty.
2. Download the installation executable from the free download site to a directory other than the `USER_INSTALL_DIR` directory.

   The following table shows the names of the files for IBM Text Search for Db2 for z/OS. The compressed files contain an installation file, a response file, and the license directory.

Table 4. File names for IBM Text Search for Db2 for z/OS installation files packages

Server	Fix *<version>*[1] for the latest version	Product DVD
zLinux	`TS_fixR1L5_.1zlinux.zip`	`TextSearchServer_setupzlinux.zip`
64-bit Linux	TS_fixR1L5_linux64.zip	`TextSearchServer_setuplinux64.zip`
32-bit Linux	`TS_fixR1L5_linux32.zip`	`TextSearchServer_setuplinux32.zip`
64-bit Windows	`TS_fixR1L5_win64.zip`	`TextSearchServer_setupwin64.zip`
32-bit Windows	`TS_fixR1L5_win32.zip`	`TextSearchServer_setupwin32.zip`

3. Before you can extract the compressed files for Linux or zLinux, you must change the permission of the file.
   For example, you might run the following commands.

```
chmod 755 TS_fixR1L5.1_linux64.zip
unzip TS_fixR1L5.1_linux64.zip
```

4. Prepare an installation response file named `ibmts_response.txt` in the same directory where you copied the executable files. Edit the response file by using a text editor, such as VI for Linux and zLinux, or Microsoft Notepad for Windows.

   Example of a partial response file:

```
# Specify the following parameters when installing
# IBM Text Search for DB2 for z/OS for the first time:
#
# LICENSE_ACCEPTED=true
# USER_INSTALL_DIR
# IA_ECMTS_SERVER_PORT
# IA_TOKEN
```

If the USER_INSTALL_DIR directory that you specify does not exist, the installation program creates the directory.

For the IA_ECMTS_SERVER_PORT value, specify a port number that is not already used by other programs on this system or on the z/OS system. A suggested value is "9000".

Attention: If you supply a port number that is used by another process, the installation aborts.

5. Copy the license directory to the same directory where you copied the installation executable file.
6. Start the installation by entering one of the following commands when positioned on the USER_INSTALL_DIR directory.

Option	Description
On 32-bit Linux, enter the following command:	`TS_fixR1L5_linux32.bin -i silent -f ibmts_response.txt`
On 64-bit Linux, enter the following command:	`TS_fixR1L5_linux64.bin -i silent -f ibmts_response.txt`
On zLinux, enter the following command:	`TS_fixR1L5_zlinux.bin -i silent -f ibmts_response.txt`
On 32-bit Windows, enter the following command from a command prompt:	`TS_fixR1L5_win32.exe -i silent -f ibmts_response.txt`
On 64-bit Windows, enter the following command from a command prompt:	`TS_fixR1L5_win64.exe -i silent -f ibmts_response.txt`

To open a command prompt window, click **Start** > **Run**, enter 'cmd', and click **OK**. The command prompt window opens. Change the directory to USER_INSTALL_DIR.

7. Verify that the installation was successful by looking at the folders that were created in the USER_INSTALL_DIR directory.

   For example, you should see at least the following folders:

```
bin
config
InstallerJVM
Java60
lib
license
log
MSRedist
plugins
resource
Uninstall_IBMTextSearch
```

8. Start the text search server by running the startup script from the USER_INSTALL_DIR directory.

Option	Description
On Linux or zLinux, enter the following command:	`bin/startup.sh`
On Windows, enter the following command:	`bin\startup`

9. Print the authentication token by entering one of the following commands.

Option	Description
On Linux or zLinux, enter the following command:	`bin/configTool.sh printToken -configPath USER_INSTALL_DIR/config`
On Windows, enter the following command:	`bin\configTool printToken -configPath USER_INSTALL_DIR\config`

Note: If the value for *-configPath* contains blanks, you must enclose the value in quotation marks.

You must insert the case-sensitive information that is collected during the installation into a Db2 administration table later. This information is the authentication token, the encryption key, the port number, and the IP name or address of the text search server.

What to do next

Next, you can configure and customize the server properties by using the Configuration Tool.

Related tasks

"Configuring a text search server" on page 19

After installing a text search server, use the Configuration Tool to customize some of the default properties.

"Resolving an installation failure caused by Security-Enhanced Linux" on page 18

Security-Enhanced Linux (SELinux) can cause the installation of a text search server on Linux to fail.

Installing fixes on top of an existing text search server

You can install fixes on top of your existing installation of IBM Text Search for Db2 for z/OS.

Before you begin



Attention: Before you install a fix, read the attached release notes to determine the prerequisites or migration procedures that apply to the fix.

Procedure

To install fixes on top of your current installation of a text search server:

1. Change the directory to the USER_INSTALL_DIR directory of your current installation of the text search server.
2. Stop the text search server by running the shutdown script.

Option	Description
On Linux or zLinux, enter the following command:	`bin/shutdown.sh`
On Windows, enter the following command from a command prompt:	`bin\shutdown`

   To open a command prompt window, click **Start -> Run**, enter `'cmd'`, and click **OK**. The command prompt window opens. Change to a directory other than USER_INSTALL_DIR.
3. Copy the installation executable file to the directory that you specified.

   Make sure that the `ibmts_response.txt` file resides in this directory. The following table shows the names of the installation executable files.

Table 5. Names of the installation executable files for IBM Text Search for Db2 for z/OS

Server	Installation executable files
32-bit Linux	TS_fix*<version>*linux32.bin
64-bit Linux	TS_fix*<version>*linux64.bin

Table 5. Names of the installation executable files for IBM Text Search for Db2 for z/OS (continued)

Server	Installation executable files
zLinux	TS_fix*<version>*zlinux.bin
32-bit Windows	TS_fix*<version>*win32.exe
64-bit Windows	TS_fix*<version>*win64.exe

4. Copy the license directory to the same directory as the installation executable file.
5. Modify the `ibmts_response.txt` file by using a text editor, such as VI for Linux and zLinux, or Microsoft Notepad for Windows.

   Example of a partial response file:

```
# The following parameters are required when upgrading
# IBM Text Search for DB2 for z/OS:
#
# LICENSE_ACCEPTED=true
# USER_INSTALL_DIR
# IA_ECMTS_SERVER_PORT=9000
# IA_TOKEN=
```

   The value for the IA_ECMTS_SERVER_PORT parameter must be the same port number that you specified for your existing text search server. For the IA_TOKEN parameter, remove the default value of `defaulttokenstr`.
6. Start the installation of the fix by entering one of the following commands.

Option	Description
On 32-bit Linux, enter the following command:	`TS_fix<version>linux32.bin -i silent -f ibmts_response.txt`
On 64-bit Linux, enter the following command:	`TS_fix<version>linux64.bin -i silent -f ibmts_response.txt`
On zLinux, enter the following command:	`TS_fix<version>zlinux.bin -i silent -f ibmts_response.txt`
On 32-bit Windows, enter the following command from a command prompt:	`TS_fix<version>win32.exe -i silent -f ibmts_response.txt`
On 64-bit Windows, enter the following command from a command prompt:	`TS_fix<version>win64.exe -i silent -f ibmts_response.txt`

Related tasks

"Resolving an installation failure caused by Security-Enhanced Linux" on page 18
Security-Enhanced Linux (SELinux) can cause the installation of a text search server on Linux to fail.

Resolving an installation failure caused by Security-Enhanced Linux

Security-Enhanced Linux (SELinux) can cause the installation of a text search server on Linux to fail.

About this task

SELinux is a Linux feature that provides support for access control security policies. If you plan to install a text search server on Linux, you need to temporarily disable SELinux. You can set SELinux to permissive mode, install the text search server, and then switch SELinux back to enforcing mode.

Procedure

To resolve an installation failure that is caused by SELinux:

1. Check what mode your Linux system is using by issuing the following command:

   ```
   cat/selinux/enforce
   ```

   This command returns "0" for permissive mode and "1" for enforcing mode.
2. To set the system to permissive mode, issue the following command:

   ```
   echo 0 >/selinux/enforce
   ```

   To issue this command, you must be logged in as root, and in the sysadm_r role. For example:

   ```
   newrole -r sysadm_r
   ```
3. After successfully installing a text search server on Linux, you can switch the system back to enforcing mode by issuing the following command:

   ```
   echo 1 >/selinux/enforce
   ```

Configuring a text search server

After installing a text search server, use the Configuration Tool to customize some of the default properties.

Before you begin

Before configuring the properties for the text search server, make sure that the text search server is shut down. The Db2 database administrator can use the SYSPROC.SYSTS_STOP stored procedure to stop text search functions until configuration is complete. However, when the server is running, you can run the options to print the current authentication token, the server port, and the current properties of the system.

About this task

You can configure some of the system-level properties and the security properties.

System configuration:

The following list shows the command options for the system-level properties:

- **`configureHTTPListener`**: Configures the server port and host name of the text search server
- **`configureParams`**: Configures the parameters for the collection
- **`generateToken`**: Generates the authentication token and encryption key
- **`printToken`**: Prints the authentication token and encryption key
- **`printAdminHTTPPort`**: Prints the administration HTTP port number
- **`printAll`**: Prints all of the current values for the properties that you can configure
- **`help`**: Displays the help message for the Configuration Tool

Security configuration:

Every API request from Db2 to a text search server is authenticated by using an authentication token. This authentication token is generated during installation of the text search server. The maximum length of the token string is 32 bytes. The Db2 database administrator must make note of and remember the authentication token so that he can copy the token to the SYSIBMTS.SYSTEXTSERVERS administration table.

An encryption key is a string of 32 bytes that is used to encrypt user IDs and password information, which the text search server needs to connect to Db2. The encryption key is generated and displayed during installation of the text search server. The Db2 database administrator must copy this encryption key to

the SYSIBMTS.SYSTEXTSERVERS administration table, where it can be used by the SYSFUN.SYSTS_ENCRYPT user-defined function.

If the authentication token and encryption key are absent, the text search server creates them by using a random seed. You can obtain the current authentication token and encryption key values by running the Configuration Tool and printing the values.

Procedure

Run the Configuration Tool by entering the appropriate command, as follows:

Option	Description
On a Linux or zLinux server	`configTool.sh configuration_command -configPath value [-locale value] -command_specific_arguments`
On a Windows server	`configTool.bat configuration_command -configPath value [-locale value] -command_specific_arguments`

Example

For example, to print the current authentication token on a Linux server, use the following command:

```
configTool.sh printToken -configPath /opt/ibm/search/config
```

Related reference

"Configuration Tool" on page 102
Use the Configuration Tool to customize configuration settings after you install a text search server.

Migrating to a new version of IBM Text Search for Db2 for z/OS

You can migrate an existing installation of IBM Text Search for Db2 for z/OS to a newer version of the text search server.

About this task

The steps that you follow to migrate an existing text search server to a new release depend on whether you stop the text search server, or whether you want the text search server to continue running during migration.

Migrating when the text search server is stopped

You can migrate to a new version of IBM Text Search for Db2 for z/OS while the existing installation of the text search server is stopped.

Procedure

To migrate when an existing text search server is stopped:

1. Call the SYSPROC.SYSTS_STOP stored procedure to stop Db2 text search functions.
2. Stop the existing text search servers by using the shutdown script.
   - On a Linux or zLinux server, run the following script: `<INSTALL_HOME>/bin/shutdown.sh`
   - On a Windows server, run the following script: `<INSTALL_HOME>/bin/shutdown.bat`
3. Install IBM Text Search for Db2 for z/OS over the existing installation of IBM Text Search for Db2 for z/OS.

4. Restart the text search servers.
5. For data sharing environments, prevent Db2 from accepting SQL CALL statements from the text search stored procedures by issuing the following command from a Db2 subsystem console:

   ```
   STOP PROCEDURE (<proc>) ACTION(REJECT) SCOPE(GROUP)
   ```

   where *<proc>* is one of the following stored procedures:

   - SYSPROC.SYSTS_ALTER
   - SYSPROC.SYSTS_CREATE
   - SYSPROC.SYSTS_DROP
   - SYSPROC.SYSTS_RESTORE
   - SYSPROC.SYSTS_STOP
   - SYSPROC.SYSTS_TAKEOVER
   - SYSPROC.SYSTS_UPDATE

   You must issue this command for all of these stored procedures. Optionally, you can complete this step for non-data sharing environments to ensure that another user does not interfere with the migration process.
6. Refresh Db2 to ensure that the Db2 subsystem uses the new text search modules for future searches.
7. Rebind the stored procedures.

   Note: You can skip this step if you already performed it when completing the ++HOLD actions for the PTF for APAR PM55239 and APAR PM84540. Otherwise, before migrating from DB2 10 to Db2 11, apply APAR PM55239 and APAR PM84540 and follow the ++HOLD instructions.
8. Call the SYSPROC.SYSTS_START stored procedure to start Db2 text search functions.
9. For data sharing environments, reactivate the definitions of the stored procedures that were stopped by issuing the following command:

   ```
   START PROCEDURE (<proc>) SCOPE(GROUP)
   ```

   where *<proc>* is the name of the stored procedure.

   Optionally, complete this step for non-data sharing environments if you stopped the stored procedures.

Related tasks

“Starting a text search server” on page 99
You can start a text search server manually by using the startup script that is provided.

“Stopping a text search server” on page 99
You can stop a text search server manually by using the shutdown script that is provided.

Related reference

“SYSPROC.SYSTS_STOP ” on page 56
Invoke the SYSPROC.SYSTS_STOP stored procedure to stop Db2 text search functions.

“SYSPROC.SYSTS_START” on page 55
Invoke the SYSPROC.SYSTS_START stored procedure to start Db2 text search functions.

Migrating when the text search server is running

You can migrate to a new version of IBM Text Search for Db2 for z/OS while an existing installation of IBM Text Search for Db2 for z/OS is still running.

Procedure

To migrate when an existing text search server is running:

1. Prevent Db2 from accepting SQL CALL statements from the text search stored procedures by issuing the following command from a Db2 subsystem console:

```
STOP PROCEDURE (<proc>) ACTION(REJECT) SCOPE(GROUP)
```

where *<proc>* is one of the following stored procedures:

- SYSPROC.SYSTS_ALTER
- SYSPROC.SYSTS_CREATE
- SYSPROC.SYSTS_DROP
- SYSPROC.SYSTS_STOP
- SYSPROC.SYSTS_TAKEOVER
- SYSPROC.SYSTS_UPDATE

You must issue this command for all of these stored procedures.

2. Rebind the stored procedures.

   Note: You can skip this step if you already performed it when completing the ++HOLD actions for the PTF for APAR PM55239 and APAR PM84540. Otherwise, before migrating from DB2 10 to Db2 11, apply APAR PM55239 and APAR PM84540 and follow the ++HOLD instructions.

3. Identify the IBM text search servers that you have installed by referring to the SYSIBMTS.SYSTEXTSERVERS administration table. For each text search server that is installed, complete the following steps:

   a) Install the new IBM Text Search for Db2 for z/OS in a new installation directory by using a different response file for each text search server.

   b) Start the new text search servers by using the startup script that is provided.

      - On a Linux or zLinux server, run the following script: `USER_INSTALL_DIR/bin/startup.sh`
      - On a Windows server, run the following script: `USER_INSTALL_DIR/bin/startup.bat`

   c) For each text search server, add the server name (SERVERNAME), port (SERVERPORT), authentication token (SERVERAUTHTOKEN), and master key (SERVERMASTERKEY) to the SYSIBMTS.SYSTEXTSERVERS administration table.

   d) Update the encrypted password value for the DB2ENCRYPTEDPW column of the SYSIBMTS.SYSTEXTSERVERS administration table by issuing the following command:

   ```
   UPDATE SYSIBMTS.SYSTEXTSERVERS SET DB2ENCRYPTEDPW =
   SYSFUN.SYSTS_ENCRYPT (<myDB2UIDpw>,SERVERMASTERKEY);
   ```

   where *<myDB2UIDpw>* is the password that you specify.

   e) In the SYSIBMTS.SYSTEXTSERVERS administration table, make note of the server ID for each of the new text search servers.

4. Call the SYSPROC.SYSTS_START stored procedure to start text search functions.

5. Restore the text search indexes to the new text search server by calling the SYSPROC.SYSTS_RESTORE stored procedure.

   You can find the information about all of the text search indexes in the SYSIBMTS.SYSTEXTINDEXES administration table. Restore the text search indexes to the new IBM Text Search for Db2 for z/OS server by issuing the following command:

   ```
   CALL SYSTS_RESTORE (<indexSchema>,<indexName>,<serverID>)
   ```

6. After all of the text search indexes are restored to the new IBM Text Search for Db2 for z/OS servers, remove the old server entries from the SYSIBMTS.SYSTEXTSERVERS administration table.

7. Reactivate the definitions of the stored procedures that were stopped by issuing the following command:

   ```
   START PROCEDURE (<proc>) SCOPE(GROUP)
   ```

   where *<proc>* is the name of the stored procedure.

Related tasks

"Starting a text search server" on page 99
You can start a text search server manually by using the startup script that is provided.

Related reference

"SYSPROC.SYSTS_STOP " on page 56
Invoke the SYSPROC.SYSTS_STOP stored procedure to stop Db2 text search functions.

"SYSPROC.SYSTS_START" on page 55
Invoke the SYSPROC.SYSTS_START stored procedure to start Db2 text search functions.

"SYSPROC.SYSTS_RESTORE " on page 53
Invoke the SYSPROC.SYSTS_RESTORE stored procedure to restore a text search index from a Db2 administration table to a text search server. After this stored procedure completes successfully, the new server is used for searching and index maintenance.

"SYSIBMTS.SYSTEXTSERVERS administration table" on page 110
The SYSIBMTS.SYSTEXTSERVERS administration table stores information about where the text search servers are installed.

Enabling text search support

Follow these steps to enable text search support for Db2 for z/OS.

Enabling text search support for Db2 11 for z/OS

The Db2 for z/OS subsystem installer is typically responsible for enabling and customizing text search support for Db2 for z/OS. You must create the text search user-defined function and administration stored procedures, bind the packages, and populate the configuration tables.

Before you begin

Before you enable text search support, make sure that all of the following requirements are met:

- All of the required Db2 PTFs must be applied.
- Db2 for z/OS must run in new-function mode.
- One or more text search servers must be installed and configured.

Procedure

To enable text search support for Db2 for z/OS:

1. In the Db2 for z/OS installation panel DSNTIPR1, specify YES in the ADVANCED CONFIGURATION OPTIONS field.
2. In the DSNTIPRA panel, enter option 8 for "IBM Text Search routines."
3. In the DSNTIPRJ panel, specify the WLM environment name for the routine, one or more authorization IDs that are to be granted EXECUTE access on the routine, and optionally, the ID that should own the package when it is bound.

   Note: If you are migrating from Db2 10 to Db2 11 and had text search support enabled on Db2 10, specify the WLM environments that you used and the authorization IDs that have been granted access to use the text search routines.
4. Finish running the installation CLIST to customize the Db2 for z/OS installation and migration jobs.
5. Submit job prefix.NEW.SDSNSAMP(DSNTIJRT) to create the databases and routines for the text search feature.
6. Set up the WLM environments for running the text search user-defined function and administration stored procedures.
7. Populate the SYSIBMTS.SYSTEXTSERVERS table and the SYSIBMTS.SYSTEXTCONNECTINFO table.

Example

The following example shows generated output from job prefix.NEW.SDSNSAMP(DSNTIJRT) for text search routines. In this example, the package owner for the routine was not specified.

```
************************************************************************
** IBM Text Search routines
************************************************************************
  SYSPROC.SYSTS_ALTER
    WLMENV(DSNWLM_GENERAL)
    GRANTTO(PUBLIC)

  SYSPROC.SYSTS_CREATE
    WLMENV(DSNWLM_GENERAL)
    GRANTTO(PUBLIC)

  SYSPROC.SYSTS_DROP
    WLMENV(DSNWLM_GENERAL)
    GRANTTO(PUBLIC)

  SYSFUN.SYSTS_ENCRYPT
    WLMENV(DSNWLM_JAVA)
    GRANTTO(PUBLIC)

  SYSPROC.SYSTS_RESTORE
    WLMENV(DSNWLM_GENERAL)
    GRANTTO(PUBLIC)

  SYSPROC.SYSTS_START
    WLMENV(DSNWLM_GENERAL)
    GRANTTO(PUBLIC)

  SYSPROC.SYSTS_STOP
    WLMENV(DSNWLM_GENERAL)
    GRANTTO(PUBLIC)

  SYSPROC.SYSTS_TAKEOVER
    WLMENV(DSNWLM_GENERAL)
    GRANTTO(PUBLIC)

  SYSPROC.SYSTS_UPDATE
    WLMENV(DSNWLM_GENERAL)
    GRANTTO(PUBLIC)
```

Issue the following VARY MVS command from the console to refresh the WLM environment for the administration stored procedures:

```
VARY WLM,APPLENV=!wlmenv!,REFRESH
```

where `!wlmenv!` is the name of the WLM environment for running the administration stored procedures.

Related tasks

“Populating the text search administration tables for Db2 for z/OS” on page 27
As part of enabling text search support for Db2 for z/OS, you must populate the SYSIBMTS.SYSTEXTSERVERS administration table and the SYSIBMTS.SYSTEXTCONNECTINFO administration table.

“Setting up the WLM environment” on page 25
You must set up the WLM environment for running the administration stored procedures and the user-defined function. The administration stored procedures and the user-defined function must run in separate WLM environments.

Related reference

“SYSPROC.SYSTS_CREATE” on page 41
Invoke the SYSPROC.SYSTS_CREATE stored procedure to enable a text column for text search indexing, which allows the text search index to be used in SQL queries that contain the CONTAINS function or the SCORE function.

“Text search administration tables” on page 107

When you enable text search support for Db2 for z/OS, the database SYSIBMTA is created with various database objects in the schema SYSIBMTS, along with specific administration tables that facilitate text search support.

Setting up the WLM environment

You must set up the WLM environment for running the administration stored procedures and the user-defined function. The administration stored procedures and the user-defined function must run in separate WLM environments.

About this task

Db2 for z/OS provides WLM environments that are suitable for running the administration stored procedures and the user-defined function. You can use the following Db2-supplied WLM environments:

- DSNWLM_GENERAL: For running most of the text search routines
- DSNWLM_JAVA: For running the SYSFUN.SYSTS_ENCRYPT user-defined function

Alternatively, you can set up WLM environments specifically for the text search routines.

Setting up the WLM environment for administration stored procedures
The stored procedures for the text search feature need to run in a WLM-established stored procedure address space.

Procedure

To set up the WLM environment:

1. Follow the instructions in Setting up a WLM application environment for stored procedures.

   The WLM-established stored procedure address space must have the following characteristics:

   - NUMTCB: 10 or more
   - APF requirement: None
   - Run options: None
   - Special DDs: None
   - Other: None
2. If necessary, create a JCL startup procedure for the address space of this WLM environment.

   If you need to create a JCL startup procedure for the address space of this WLM environment, you can model it on the DSNWLM sample startup procedure that was created by the Db2 installation job DSNTIJMV.

   a) Change the procedure name from DSNWLM to the procedure name that you specified when you set up the WLM environment.

   b) Change the value of APPLENV to the name of the WLM environment that you set up for the text search stored procedures.

   c) Change the value of DB2SSN to your Db2 subsystem name.

      Each member of a Db2 data sharing environment needs access to the startup procedure for the address space of this WLM environment.

Related concepts

“Administration stored procedures for text search” on page 35

A set of administration stored procedures are provided by Db2 for z/OS to start and stop text search functions, and to create, alter, drop, update, take over, and restore text search indexes.

Setting up the runtime environment for Java routines
The SYSFUN.SYSTS_ENCRYPT user-defined function requires that the WLM environment be set up for processing Java™ routines.

About this task

If you want to reuse an existing WLM environment for Java routines, the WLM environment must comply with all of the requirements that are specified in this task.

Procedure

To set up a new WLM environment:

1. Install Java 2 Technology Edition, Version 5 or later.

   You can install the 31-bit or the 64-bit version. This software might already be installed on your system. The location of the installation is typically under the HFS directory, such as `/usr/lpp/java/5.0 or /usr/lpp/java/5.0_64`.
2. Install the IBM Db2 Driver for JDBC and SQLJ for the version of Db2 for z/OS that you are using.

   The required JDBC driver files might already be installed with Db2 on your system. The location of the installation is typically under the HFS directory, such as `/usr/lpp/db2b10/jdbc`.

   Note: If you migrated your Db2 subsystem from a previous release of Db2, the previous release of the JDBC and SQLJ driver will not work anymore. You must migrate to the latest version of the Db2 Driver for JDBC and SQLJ.
3. Create a JCL startup procedure for this WLM environment's address space to run the SYSFUN.SYSTS_ENCRYPT user-defined function.

   In addition to the required DD statements for Java routines, this function requires that the JSPDEBUG DD must be removed or commented out for security reasons. This requirement is because the SYSFUN.SYSTS_ENCRYPT function takes a password as an input parameter, and this password might be logged in the debugging output if JSPDEBUG DD is enabled.

   The following example shows the JCL that you can use to run the SYSFUN.SYSTS_ENCRYPT function:

   ```
   //procName PROC  DB2SSN=ssnm,NUMTCB=1,APPLENV=wlmEnvName
   //TCBNUM1  EXEC  PGM=DSNX9WLM,TIME=1440,
   //             PARM='&DB2SSN,&NUMTCB,&APPLENV',
   //             REGION=0M
   //STEPLIB   DD DISP=SHR,DSN=ceepre.SCEERUN   <- LE runtime lib
   //          DD DISP=SHR,DSN=dsnpre.SDSNEXIT  <- DB2 exit lib
   //          DD DISP=SHR,DSN=dsnpre.SDSNLOAD  <- DB2 runtime lib
   //          DD DISP=SHR,DSN=dsnpre.SDSNLOD2  <- JDBC/SQLJ DLL lib
   //JAVAENV   DD DISP=SHR,DSN=prefix.JSPENV    <- JSP runtime opts
   //*JSPDEBUG  DD SYSOUT=A                     <- debugging output
   //CEEDUMP   DD SYSOUT=A  .                   <- LE dump output
   //SYSPRINT  DD SYSOUT=A                      <- diagnostics
   ```

   where:

   - *procName* specifies the procedure name that is associated with the WLM environment.
   - *wlmEnvName* specifies the name of the WLM environment.
   - *ssnm* specifies the name of the Db2 subsystem.
   - *NUMTCB* specifies the startup time. Set this value to 1 for optimal startup time performance.
   - *ceepre* specifies the prefix of the IBM Language Environment (LE) runtime library.
   - *dsnpre* specifies the prefix of your Db2 for z/OS runtime libraries.
   - *prefix.JSPENV* specifies the data set that contains the LE runtime options for running the SYSFUN.SYSTS_ENCRYPT function.
4. Create a JAVAENV runtime environment data set with the DSN that you specified in the JCL procedure.

This data set must contain at least the following two LE environment variables:

- *JAVA_HOME* specifies the installation directory of the Java 2 Technology Edition, Version 5 or later.
- *JCC_HOME* specifies the installation directory of the Db2 Driver for JDBC and SQLJ.

Depending on your system, the *DB2_BASE* variable might be required. *DB2_BASE* specifies the Db2 HFS program files. You must specify this variable only if the files are installed in a directory other than `/usr/lpp/db2b10/base` on your system.

The following example shows a typical JAVAENV data set, when customized appropriately for the SYSFUN.SYSTS_ENCRYPT function:

```
ENVAR("JAVA_HOME=/usr/lpp/java/J5.0",
"JCC_HOME=/usr/lpp/db2b10/jdbc"),
XPLINK(ON)
```

Note: If you migrated your Db2 subsystem from a previous release of Db2, you must remove the *DB2_HOME* variable from any existing JAVAENV data sets and add the new *JCC_HOME* variable.

5. Define the WLM environment for Java routines by specifying the appropriate values on the WLM setup panels.

Related tasks

Setting up the environment for Java routines (Application Programming for Java)

Populating the text search administration tables for Db2 for z/OS

As part of enabling text search support for Db2 for z/OS, you must populate the SYSIBMTS.SYSTEXTSERVERS administration table and the SYSIBMTS.SYSTEXTCONNECTINFO administration table.

About this task

The database administrator must populate the server information columns in the SYSIBMTS.SYSTEXTSERVERS table with information about where one or more text search servers are installed.

Then, the Db2 database administrator must populate the SYSIBMTS.SYSTEXTCONNECTINFO administration table with the Db2 for z/OS connection information. This connection information is needed so that a text search server can connect to the Db2 subsystem and maintain the persistent copy of the Db2 full-text indexes.

Procedure

Create and process the necessary SQL statements by using the information that you collected during installation of the text search server (the authentication token, the encryption key, the port number, and the IP name or address of the text search server).

Use the DDF DOMAIN and TCPPORT values of your Db2 subsystem or data sharing group for DB2HOSTNAME and DB2SERVICEPORT. For the Db2 user ID (DB2UID), you can use any ID that is allowed to connect to Db2 for z/OS. This ID is used by the text search server to update full-text indexes in Db2. When you run the user-defined function SYSFUN.SYSTS_ENCRYPT, only the encrypted password is stored in Db2.

Example

You can use the following sample SQL INSERT statement to populate the tables:

```
//*********************************************************************
//* THE FOLLOWING EXAMPLE SQL STATEMENTS ARE FOR:
//* - REGISTERING INFORMATION ABOUT WHERE A TEXT SEARCH SERVER FOR DB2
//*   FOR Z/OS IS INSTALLED
//* - USING THE SYSFUN.SYSTS_ENCRYPT FUNCTION TO ENCRYPT A DB2 PASSWORD
//*   THAT THE TEXT SEARCH SERVER NEEDS TO CONNECT TO DB2
//* - ADDING INFORMATION SO THAT A TEXT SEARCH SERVER CAN CONNECT TO
```

```
//*   THE DB2 FOR Z/OS SUBSYSTEM
//*********************************************************************
//*  //          DD  *
//*    INSERT INTO SYSIBMTS.SYSTEXTSERVERS
//*      (SERVERNAME,SERVERPORT,SERVERAUTHTOKEN,SERVERMASTERKEY)
//*      VALUES('9.30.176.75',10000,'1aPgk6Q='
//*            ,'1aPgk6TkD/FVt8BuCyO8Bg==');
//*
//*    UPDATE SYSIBMTS.SYSTEXTSERVERS SET DB2ENCRYPTEDPW =
//*           SYSFUN.SYSTS_ENCRYPT('myDB2UIDpw',SERVERMASTERKEY);
//*
//*    INSERT INTO SYSIBMTS.SYSTEXTCONNECTINFO
//*      (DB2HOSTNAME,DB2SERVICEPORT,DB2UID)
//*      VALUES('db2host.svl.ibm.com','446','DB2UID');
//*
```

Related tasks

"Adding a text search server to the system" on page 33
After initially populating the SYSIBMTS.SYSTEXTSERVERS table, the Db2 database administrator can add another text search server by updating the connectivity information in the administration table.

Related reference

INSERT (DB2 SQL)

"SYSFUN.SYSTS_ENCRYPT" on page 74
Use the SYSFUN.SYSTS_ENCRYPT user-defined function that is supplied with Db2 for z/OS to encrypt the passed value according to the text search server encryption method by using the input key.

"SYSIBMTS.SYSTEXTCONNECTINFO administration table" on page 111
The SYSIBMTS.SYSTEXTCONNECTINFO administration table stores information about the Db2 for z/OS connection information so that a text search server can connect to the Db2 subsystem.

"SYSIBMTS.SYSTEXTSERVERS administration table" on page 110
The SYSIBMTS.SYSTEXTSERVERS administration table stores information about where the text search servers are installed.

Verifying the installation

Db2 provides a sample application that you can use to verify installation of the text search feature. The sample application consists of job DSNTEJ6O and two programs, DSN8DTS1 and DSN8DTS2.

The program DSN8DTS1 is a C-language source module that shows how to use the Db2-supplied stored procedures SYSPROC.SYSTS_START, SYSPROC.SYSTS_CREATE, and SYSPROC.SYSTS_UPDATE to create and initialize a text search index. Program DSN8DTS1 accepts the following three input parameters:

- The name of the schema for the text search index that is to be created. For example, DSN8910.
- The name of the text search index that is to be created. For example, IAA3.
- The name of the table and column on which to create the text search index. For example, DSN8910.TEXT_SEARCH(AA3).

The program DSN8DTS2 is a C-language source module that shows how to use the Db2-supplied stored procedures SYSPROC.SYSTS_START and SYSPROC.SYSTS_DROP to drop a text search index. Program DSN8DTS2 accepts the following two input parameters:

- The name of the schema for the text search index that is to be dropped. For example, DSN8910.
- The name of the text search index that is to be dropped. For example, IAA3.

The JCL job DSNTEJ6O performs the following tasks:

- Creates and populates a sample table called DSN8910.TEXT_SEARCH that contains Unicode-encoded text data in fixed-character, varying character, varying graphic, XML, and character LOB (CLOB) format.
- Prepares, binds, and runs sample program DSN8DTS1 to create text search indexes on table DSN8910.TEXT_SEARCH.
- Processes queries on DSN8910.TEXT_SEARCH using the SCORE and CONTAINS functions.
- Prepares, binds, and runs sample program DSN8DTS2 to drop the text search index on DSN8910.TEXT_SEARCH.

The steps of job DSNTEJ6O are listed below. You need to customize job DSNTEJ60 to run on your Db2 system. For instructions, see the job prolog. See the source code for DSN8DTS1 and DSN8DTS1 for more information about those parts.

```
Step     Procstep Description                                    Return code
-------- -------- ---------------------------------------------- ------------
PH06OS01          Drop the sample search table                   0000
PH06OS02          Create and populate sample search table        0000
PH06OS03          Prepare DSN8DTS1
         PC       - precompile                                   0000
         C        - compile                                      0000
         PLKED    - pre-link edit                                0004
         LKED     - link edit                                    0000
PH06OS04          Bind the plan and package for DSN8DTS1         0000 or 0004
PH06OS05          Create a text search index on a VARCHAR col    0000
PH06OS06          Create a text search index on a CHAR col       0000
PH06OS07          Create a text search index on a VARGRAPHIC col 0000
PH06OS08          Create a text search index on a XML col        0000
PH06OS09          Create a text search index on a CLOB col       0000
PH06OS10          Select data using CONTAINS and SCORE functions 0000
PH06OS11          Prepare DSN8DTS2
         PC       - precompile                                   0000
         C        - compile                                      0000
         PLKED    - pre-link edit                                0004
         LKED     - link edit                                    0000
PH06OS12          Bind the plan and package for DSN8DTS2         0000 or 0004
PH06OS13          Drop the text search index on the VARCHAR col  0000
PH06OS14          Drop the text search index on the CHAR col     0000
PH06OS15          Drop the text search index on the VARGRAPHIC cl 0000
PH06OS16          Drop the text search index on the XML col      0000
PH06OS17          Drop the text search index on the CLOB col     0000
```

Validating step PHO60S10

The correct results for validation step PHO60S10 are as follows:

```
SELECT INT1, CONTAINS(VC2, 'text')
     FROM DSN8910.TEXT_SEARCH;
       +---------------------------------+
       |      INT1      |                |
       +---------------------------------+
     1_|              1 |              0 |
     2_|              2 |              1 |
     3_|              3 |              0 |
     4_|              4 |              1 |
     5_|              5 |              0 |
     6_|              6 |              0 |
     7_|              7 |              1 |
     8_|              8 |              1 |
     9_|              9 |              0 |
    10_|             10 |              1 |
       +---------------------------------+
SUCCESSFUL RETRIEVAL OF       10 ROW(S)

SELECT INT1, SCORE(VC2, 'text')
     FROM DSN8910.TEXT_SEARCH
    WHERE CONTAINS(VC2, 'text') = 1;
       +---------------------------------------+
       |      INT1      |                      |
       +---------------------------------------+
     1_|              2 |  4.100000000000000E-02 |
     2_|              4 |  4.100000000000000E-02 |
     3_|              7 |  4.100000000000000E-02 |
     4_|              8 |  4.100000000000000E-02 |
     5_|             10 |  4.100000000000000E-02 |
       +---------------------------------------+
SUCCESSFUL RETRIEVAL OF        5 ROW(S)

   SELECT INT1, CONTAINS(AA3, 'DB2')
     FROM DSN8910.TEXT_SEARCH;
       +---------------------------------+
       |      INT1      |                |
       +---------------------------------+
     1_|              1 |              0 |
     2_|              2 |              0 |
     3_|              3 |              0 |
     4_|              4 |              0 |
```

```
     5_|              5 |              1 |
     6_|              6 |              0 |
     7_|              7 |              0 |
     8_|              8 |              1 |
     9_|              9 |              0 |
    10_|             10 |              0 |
       +--------------------------------+
SUCCESSFUL RETRIEVAL OF       10 ROW(S)

   SELECT INT1, SCORE(AA3, 'DB2')
     FROM DSN8910.TEXT_SEARCH
    WHERE CONTAINS(AA3, 'DB2') = 1;

       +---------------------------------------+
       |      INT1      |                      |
       +---------------------------------------+
     1_|              5 |  6.000000000000000E-02 |
     2_|              8 |  6.000000000000000E-02 |
       +---------------------------------------+
SUCCESSFUL RETRIEVAL OF        2 ROW(S)

   SELECT INT1, VC2, AA3
     FROM DSN8910.TEXT_SEARCH
    WHERE CONTAINS(AA3, 'DB2') = 1
   AND CONTAINS(VC2, 'text') = 1;
       +-------------------------------------------------------------+
       |      INT1      |         VC2              |    AA3          |
       +-------------------------------------------------------------+
     1_|              8 | text search capabilities | in DB2.         |
       +-------------------------------------------------------------+
SUCCESSFUL RETRIEVAL OF        1 ROW(S)
```

Starting text search functions

To start text search functions, the Db2 database administrator runs a program that calls the SYSPROC.SYSTS_START stored procedure.

About this task

Text search support includes SQL statements that use the CONTAINS function, the SCORE function, and the following administration stored procedures:

- SYSPROC.SYSTS_ALTER
- SYSPROC.SYSTS_CREATE
- SYSPROC.SYSTS_DROP
- SYSPROC.SYSTS_RESTORE
- SYSPROC.SYSTS_START
- SYSPROC.SYSTS_STOP
- SYSPROC.SYSTS_TAKEOVER
- SYSPROC.SYSTS_UPDATE

Procedure

Run a program that calls the SYSPROC.SYSTS_START stored procedure.

You can run a stored procedure by issuing a CALL statement using one of the following tools:

- QMF
- The Db2 for z/OS command line processor at the UNIX System Services prompt
- The Db2 client's command line editor on your Linux, UNIX, or Windows workstation

Example

The following example shows a sample program that calls the SYSPROC.SYSTS_START stored procedure:

```
CALL SYSPROC.SYSTS_START();
```

What to do next

Now, you can create a text search index.

Related reference

“SYSPROC.SYSTS_START” on page 55
Invoke the SYSPROC.SYSTS_START stored procedure to start Db2 text search functions.

Creating a text search index

The Db2 database administrator runs a program that calls the SYSPROC.SYSTS_CREATE stored procedure to create a text search index.

Before you begin

The Db2 table must contain a ROWID column. To avoid search performance problems, there must be a Db2 index on the ROWID column. You can create the Db2 index after creating the text search index.

Procedure

Run a program that calls the SYSPROC.SYSTS_CREATE stored procedure.

Example

The following example shows a sample program that creates a text search index by calling the SYSPROC.SYSTS_CREATE stored procedure.

```
SYSPROC.SYSTS_CREATE('IXSCHEMA', 'IXNAME', 'TABSCHEMA.TABNAME(COLNAME)', ' ');
CALL SYSPROC.SYSTS_UPDATE('IXSCHEMA', 'IXNAME', ' ' );
```

The text search index is empty until the first time that you run an update of the text search index.

Related tasks

“Checking the event table after calling SYSPROC.SYSTS_CREATE” on page 66
To get the best performance for the CONTAINS and SCORE functions, the table specified in the SYSTS_CREATE stored procedure must have an index on its ROWID column.

Related reference

“SYSPROC.SYSTS_CREATE” on page 41
Invoke the SYSPROC.SYSTS_CREATE stored procedure to enable a text column for text search indexing, which allows the text search index to be used in SQL queries that contain the CONTAINS function or the SCORE function.

Modifying the options of a text search index

The Db2 database administrator runs a program that calls the SYSPROC.SYSTS_ALTER stored procedure to modify the options of an existing text search index.

Before you begin

Important: IBMTS This information applies to IBM Text Search for Db2 for z/OS. IBMTS

You can alter only text search indexes that were created by using the SYSPROC.SYSTS_CREATE stored procedure.

About this task

You can alter the frequency of updates to a text search index, the minimum number of changes to text documents before a text search index is updated, and some configuration options. You also can alter the schema and name of the user-defined function that accesses text documents.

Procedure

To modify the options of a text search index:

1. Run a program that calls the SYSPROC.SYSTS_ALTER stored procedure.
2. After you modify the options for a text search index, run the SYSPROC.SYSTS_UPDATE stored procedure to update the text search index that is stored on the text search server.

Example

Example 1: The following example shows a sample program that modifies a text search index by calling the SYSPROC.SYSTS_ALTER stored procedure. This example modifies the frequency of updates, the minimum number of changes to text documents before an index is updated, and some configuration options.

```
CALL SYSPROC.SYSTS_ALTER('USRT002', 'INDEX2' ,' UPDATE FREQUENCY NONE
UPDATE MINIMUM 1 INDEX CONFIGURATION (UPDATEAUTOCOMMIT 200 ,
IGNOREEMPTYDOCS 1 , UPDATEWITHBACKUP 1 ,COMMENT NEWINDEXOPTIONS)'  )
```

Example 2: The following example calls the SYSPROC.SYSTS_ALTER stored procedure to modify the name of the user-defined function that accesses text documents. This example also shows calling the SYSPROC.SYSTS_UPDATE stored procedure to update the text search index that is stored on the text search server.

```
CALL SYSPROC.SYSTS_ALTER('USRT002', 'INDEX2' ,‘RENAME FUNCTION
FCT_NEWNAME’ )
CALL SYSPROC.SYSTS_UPDATE('USRT002', 'INDEX2', ' ' );
```

Related tasks

“Scheduling the SYSPROC.SYSTS_UPDATE stored procedure ” on page 63
You can use the Db2 administrative task scheduler to call the SYSPROC.SYSTS_UPDATE stored procedure according to a specific schedule.

Related reference

“SYSPROC.SYSTS_ALTER” on page 35
Invoke the SYSPROC.SYSTS_ALTER stored procedure to modify the options of a text search index that was created by using the SYSPROC.SYSTS_CREATE stored procedure.

“SYSPROC.SYSTS_UPDATE ” on page 59
Invoke the SYSPROC.SYSTS_UPDATE stored procedure to update the text search index to reflect the current contents of the text column that the text search index is associated with.

Updating a text search index

The Db2 database administrator runs a program that calls the SYSPROC.SYSTS_UPDATE stored procedure to update a text search index.

Procedure

Run a program that calls the SYSPROC.SYSTS_UPDATE stored procedure.

Example

The following example shows a sample program that updates a text search index by calling the SYSPROC.SYSTS_UPDATE stored procedure:

```
CALL SYSPROC.SYSTS_UPDATE('IXSCHEMA', 'IXNAME', ' ');
```

What to do next
Now, you can perform search queries on the text search index.
Related reference
“SYSPROC.SYSTS_UPDATE ” on page 59
Invoke the SYSPROC.SYSTS_UPDATE stored procedure to update the text search index to reflect the current contents of the text column that the text search index is associated with.

Searching a text search index

An application programmer can issue an SQL statement with a CONTAINS function or a SCORE function to search a text search index.

Before you begin
The user who is performing the text queries on a Db2 table must have the standard privilege set that is required for any form of query.

Procedure

Issue an SQL statement with a CONTAINS function or a SCORE function.

What to do next
To create, update, and search additional text search indexes, repeat this step for each text search index.
Related reference
“CONTAINS” on page 69
The CONTAINS function searches a text search index using criteria that are specified in a search argument and returns a result about whether or not a match was found.

“SCORE” on page 72
The SCORE function searches a text search index using criteria that are specified in a search argument and returns a relevance score that measures how well a document matches the query.

Adding a text search server to the system

After initially populating the SYSIBMTS.SYSTEXTSERVERS table, the Db2 database administrator can add another text search server by updating the connectivity information in the administration table.

Procedure

To update the text search server connectivity information:

1. Populate the SYSIBMTS.SYSTEXTSERVERS table.
2. Run a program that calls the SYSPROC.SYSTS_START stored procedure.

Example

The following two sample SQL statements show how to add a new text search server to the Db2 subsystem:

```
INSERT INTO SYSIBMTS.SYSTEXTSERVERS(SERVERNAME,SERVERPORT,SERVERAUTHTOKEN,
SERVERMASTERKEY, DB2ENCRYPTEDPW)
VALUES('9.30.176.75',10000,'1aPgk6Q=','1aPgk6TkD/FVt8BuCyO8Bg==',
SYSFUN.SYSTS_ENCRYPT('db2 password value', '1aPgk6TkD/FVt8BuCyO8Bg=='));
```

```
CALL SYSPROC.SYSTS_START();
```

Related tasks
“Populating the text search administration tables for Db2 for z/OS” on page 27

As part of enabling text search support for Db2 for z/OS, you must populate the SYSIBMTS.SYSTEXTSERVERS administration table and the SYSIBMTS.SYSTEXTCONNECTINFO administration table.

"Starting text search functions" on page 30
To start text search functions, the Db2 database administrator runs a program that calls the SYSPROC.SYSTS_START stored procedure.

Chapter 5. Administration stored procedures for text search

A set of administration stored procedures are provided by Db2 for z/OS to start and stop text search functions, and to create, alter, drop, update, take over, and restore text search indexes.

SYSPROC.SYSTS_ALTER

Invoke the SYSPROC.SYSTS_ALTER stored procedure to modify the options of a text search index that was created by using the SYSPROC.SYSTS_CREATE stored procedure.

After you modify the options for a text search index, you must run the SYSPROC.SYSTS_UPDATE stored procedure to update the text search index that is stored on the text search server.

Prerequisites

Before you invoke the SYSPROC.SYSTS_ALTER stored procedure, verify the following prerequisites:

- Db2 text search functionality was started by invoking the SYSPROC.SYSTS_START stored procedure.
- The text search index was created (by invocation of the SYSPROC.SYSTS_CREATE stored procedure).
- The SYSIBMTS.SYSTEXTSERVERS table contains at least one entry, and the DB2ENCRYPTEDPW column in the SYSIBMTS.SYSTEXTSERVERS table has a valid, encrypted password value for the user ID that the text search server uses to connect to Db2 for z/OS.
- The following stored procedures are not running for the index that you want to modify: SYSPROC.SYSTS_CREATE, SYSPROC.SYSTS_DROP, SYSPROC.SYSTS_RESTORE, SYSPROC.SYSTS_TAKEOVER, and SYSPROC.SYSTS_UPDATE.
- At least one text search server is running.
- The distributed data facility (DDF) must be started, even if you call the stored procedure locally on your z/OS system. The DDF is required to allow the text search server to connect to Db2 to store a text search index.

Authorization

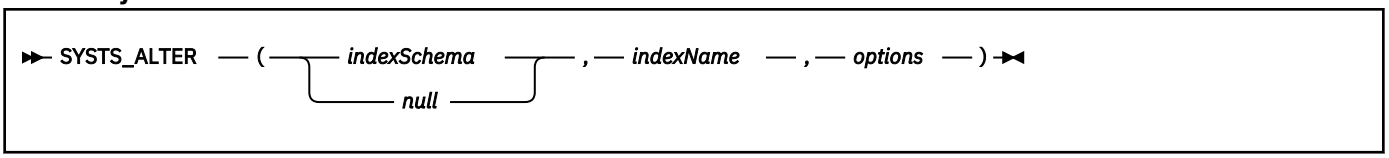
The user ID under which this stored procedure is invoked must have the following privileges:

- EXECUTE on the procedure
- EXECUTE on the packages SYSIBMTS.*
- SELECT and UPDATE privileges on the following administration tables:
  - SYSIBMTS.SYSTEXTCOLUMNS
  - SYSIBMTS.SYSTEXTCONFIGURATION
  - SYSIBMTS.SYSTEXTINDEXES

In addition, the user ID that is listed in the SYSIBMTS.SYSTEXTCONNECTINFO table, or that is obtained from the data source property at the text search server, must have SELECT, INSERT, UPDATE, and DELETE privileges for the text search index table after connecting to Db2 for z/OS on a T4 Java connection and going through the DRDA connect processing. The database name is SYSIBMTS. The index table name is SYSIBMTS.INDEX_[n], where [n] is the index ID according to the INDEXID column of the SYSIBMTS.SYSTEXTINDEXES administration table.

© Copyright IBM Corp. 2007, 2019

Syntax



The schema qualifier is SYSPROC.

Parameters

indexSchema
Identifies the schema of the text search index. If this parameter is null, the value of the CURRENT SCHEMA special register for the invoker is used.

Recommendation: Use a valid SQL name for this parameter.

The data type of this parameter is VARCHAR(128).

indexName
Identifies the name of the text search index. The name of the text search index together with the index schema uniquely identifies the text search index in the Db2 subsystem. You must specify a non-null value for this parameter.

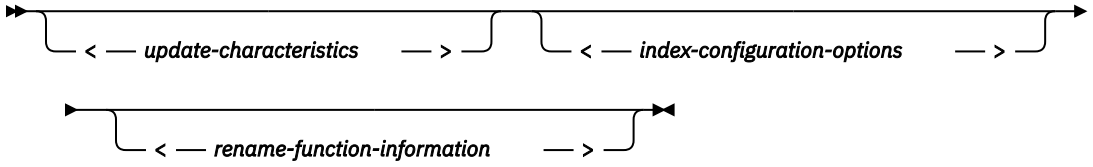
Recommendation: Use a valid SQL name for this parameter.

The data type for this parameter is VARCHAR(128).

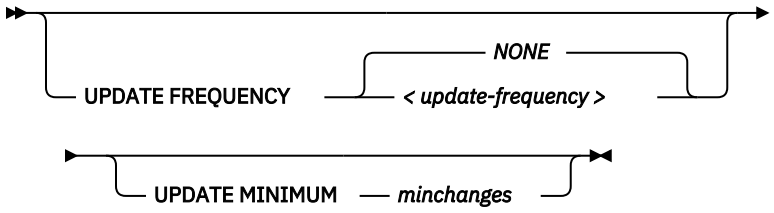
options
A character string that specifies the various options that are available for this stored procedure.

The data type for this parameter is VARCHAR(32000).

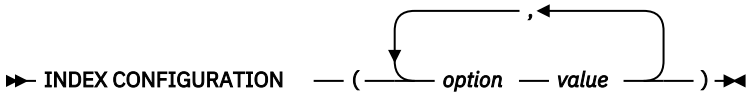
options



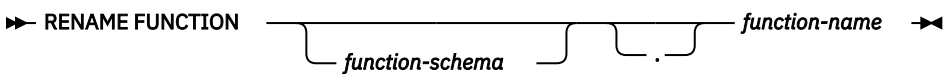
update-characteristics



index-configuration-options



rename-function-information



update-characteristics

Specifies the frequency of updates to the text search index and the minimum number of changes to text documents before the text search index is updated incrementally at the specified time.

UPDATE FREQUENCY ***update-frequency***

Specifies when to make updates to the text search index. The default value is NONE. This option might be useful for a text column in which no further changes are planned.

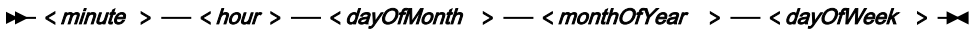
Text search index updates are not performed automatically. The scheduling of update requests is the responsibility of the Db2 for z/OS database administrator.

For each text search index, the UPDATE FREQUENCY value is stored as data type VARCHAR in the UPDATEFREQUENCY column of the SYSIBMTS.SYSTEXTINDEXES administration table. The value is stored in UNIX cron format.

To activate automatic text search index updates, schedule an index update task in Db2 for z/OS for each text search index. This task must call the SYSPROC.SYSTS_UPDATE stored procedure.

To schedule an index update task in Db2 for z/OS, you can use the Db2 administrative task scheduler. Call the SYSPROC.ADMIN_TASK_ADD stored procedure and provide the UPDATEFREQUENCY value from the SYSIBMTS.SYSTEXTINDEXES administration table as the input parameter **point-in-time**.

update-frequency



The format of the *update-frequency* option is a list of the five values separated by a blank space. The five values represent the minutes, the hours, the days of the month, the months of the year, and the days of the week beginning with Sunday.

If you specify an interval of values, or an asterisk (*), you can use a forward slash (/) at the end of the defined interval to specify a step value for this interval.

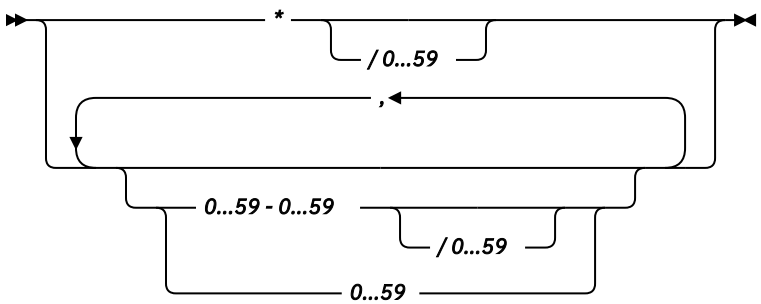
Example: This example specifies that the index update is to run every quarter hour (`0,15,30,45`) on the even hours between 8 a.m. and 6:45 p.m. (`8-18/2` is equivalent to `8,10,12,14,16,18`), from Monday to Friday every month of the year (`* * 1-5`).

```
0,15,30,45 8-18/2 * * 1-5
```

minute

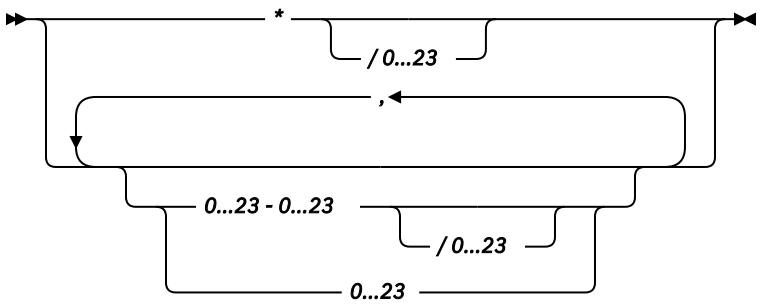
Specifies the minutes of the hour when the text search index is to be updated. You can specify an asterisk (*) for an interval of every five minutes, or you can specify an integer from 0 (zero) through 59. You cannot repeat values. The minimum update frequency is five minutes. A value of 1,4,8 is an invalid interval.

update-frequency (minute)



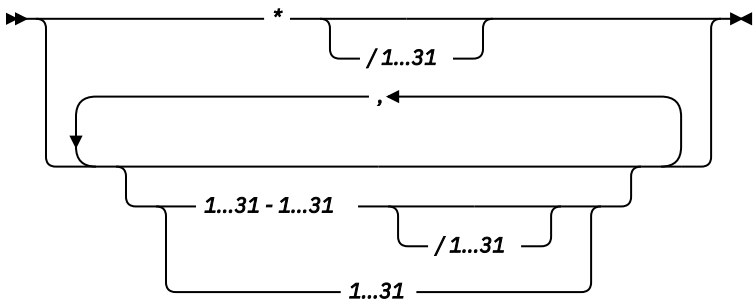
hour
Specifies the hours of the day when the text search index is to be updated. You can specify an asterisk (*) for every hour, or you can specify an integer from 0 (zero) through 23. You cannot repeat values.

update-frequency (hour)



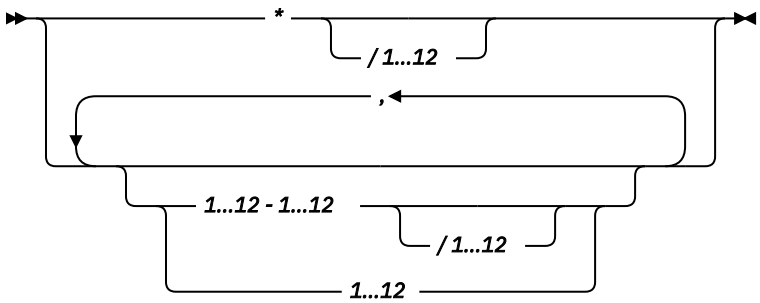
dayOfMonth
Specifies the days of the month when the text search index is to be updated. You can specify an asterisk (*) for every day, or you can specify an integer from 1 through 31. You cannot repeat values.

update-frequency (dayOfMonth)



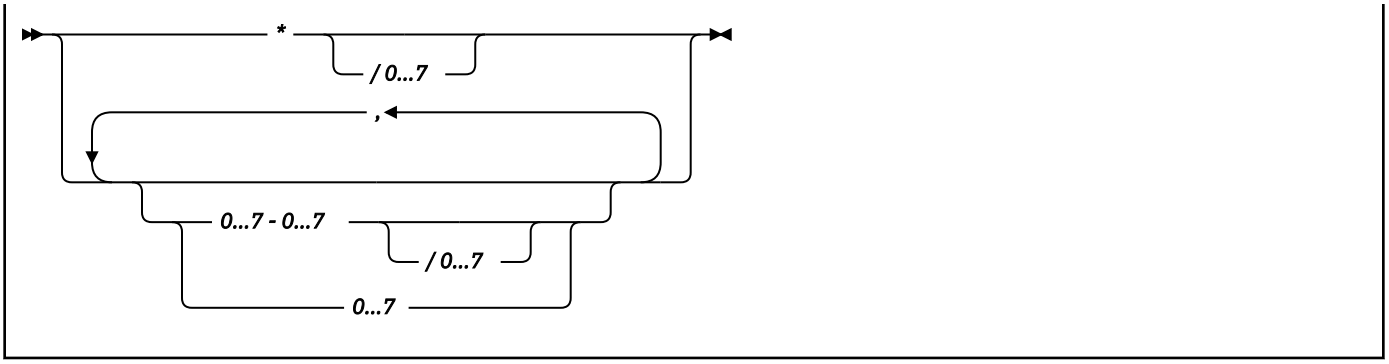
monthOfYear
Specifies the months of the year when the text search index is to be updated. You can specify an asterisk (*) for every month, or you can specify an integer from 1 through 12. You cannot repeat values.

update-frequency (monthOfYear)



dayOfWeek
Specifies the days of the week when the text search index is to be updated. You can specify an asterisk (*) for every day, or you can specify an integer from 0 (zero) through 7. Both 0 and 7 are valid values for Sunday. You cannot repeat values.

update-frequency (dayOfWeek)



UPDATE MINIMUM *minchanges*
Specifies the minimum number of changes that are made to text documents before the text search index is updated incrementally at the time specified in the *update-frequency* option. The value must be an integer between 1 and 2147483647. The default value is taken from the UPDATEMINIMUM column in the SYSIBMTS.SYSTEXTDEFAULTS table.

This option is ignored when you update the text search index, unless you specify the USING UPDATE MINIMUM option in the SYSIBMTS.SYSTS_UPDATE stored procedure.

index-configuration-options
Specifies additional index-specific values as option value pairs. You must enclose string values in single quotation marks. A single quotation mark within a string value must be represented by two consecutive single quotation marks.

COMMENT
Specifies a comment that is stored in the REMARKS column of the SYSIBMTS.SYSTEXTINDEXES administration table and as the description of the text search server collection.

The value for this option is a string value that is less than or equal to 512 bytes.

UPDATEWITHBACKUP
Specifies whether the text search index is backed up.

The supported values for this option are 0 (zero) and 1. The default value is 1.

The value 1 specifies that the text search index is backed up in the index table during the processing of the SYSIBMTS.SYSTS_UPDATE stored procedure.

The value 0 (zero) specifies that the text search index is not backed up in the index table during processing of the SYSIBMTS.SYSTS_UPDATE stored procedure.

If the value is changed from 0 to 1, you must use the SYSIBMTS.SYSTS_UPDATE stored procedure with the ALLROWS option to save the current index in the backup table.

UPDATEAUTOCOMMIT
Specifies how often a commit operation is performed when fetching documents during an index update. A value of 0 (zero) means that a commit operation occurs only at the end of processing. The default value is 100 and COMMITTYPE specified as ROWS.

UPDATEAUTOCOMMIT requires a numeric value. This value represents either a number of rows or a number of hours, depending on the specification for COMMITTYPE. If COMMITTYPE is set to HOURS, the value cannot exceed 24. The text index is committed when the value of UPDATEAUTOCOMMIT is reached.

If update processing takes a very long time and Db2 active logs are small, consider using the default value. Otherwise, the active log entries of customer transactions that run in parallel to the text search index update cannot be cleared and might lead to a full the active log.

COMMITTYPE
Specifies either ROWS or HOURS. If you specify ROWS, the text index is committed after the number of rows that is specified by UPDATEAUTOCOMMIT is reached. If you specify HOURS,

the text index is committed after the period of time that is specified by UPDATEAUTOCOMMIT is reached. The default value is ROWS.

You must use this option in conjunction with UPDATEAUTOCOMMIT or COMMITCYCLES.

COMMITCYCLES
Specifies a numeric value for the number of commit cycles for the processing of an index update. By default, the number of commit cycles is unrestricted. If COMMITCYCLES is 0 (zero), the update process uses as many cycles as needed to finish processing.

You must use this option in conjunction with UPDATEAUTOCOMMIT and COMMITTYPE.

rename-function-information
Specifies the schema and name of the user-defined function that is used to access text documents.

function-schema. function-name
Identifies the schema and the name of a built-in or user-defined function that is to be used by the Db2 for z/OS text search feature to access text documents that are in a column that is not of a supported data type, or that are stored elsewhere. The function has one input parameter for the text column data type (for example, an integer that serves as a foreign key to the document content in another table), and it returns a value of one of the supported data types. The function transforms the text column content to the indexed document content.

Tip: For optimal performance, specify the ALLOW PARALLEL option when the function is created.

Restriction: Cast functions and functions with more than one argument are not allowed.

Examples

Example 1: The following example shows how to use the Db2 CALL statement to invoke the SYSPROC.SYSTS_ALTER stored procedure. This example uses the SYSPROC.SYSTS_ALTER stored procedure to modify the value of the UPDATEWITHBACKUP parameter.

```
CALL SYSPROC.SYSTS_ALTER ('SCHEMA1', 'IVC2',
'INDEX CONFIGURATION (UPDATEWITHBACKUP 0)');
```

This example uses the following parameters:

- `'SCHEMA1'` = *indexSchema*
- `'IVC2'` = *indexName*
- `'INDEX CONFIGURATION (UPDATEWITHBACKUP 0)'` = *options*

Example 2: The following example shows calling the SYSPROC.SYSTS_ALTER stored procedure to modify the value of the UPDATE MINIMUM parameter.

```
CALL SYSPROC.SYSTS_ALTER ('SCHEMA1', 'IVC2', 'UPDATE MINIMUM 10');
```

This example uses the following parameters:

- `'SCHEMA1'` = *indexSchema*
- `'IVC2'` = *indexName*
- `'UPDATE MINIMUM 10'` = *options*

Example 3: The following examples show how you can use the UPDATEAUTOCOMMIT, COMMITTYPE, and COMMITCYCLES options to specify how often a commit operation is performed when documents are fetched during an index update. In the first example, the commit operation is performed after 200 rows with four commit cycles.

```
CALL SYSPROC.SYSTS_ALTER(  'USRT002', 'INDEX2' ,' UPDATE FREQUENCY NONE
UPDATE MINIMUM 1 INDEX CONFIGURATION (   UPDATEAUTOCOMMIT 200, COMMITTYPE ROWS,
COMMITCYCLES 4 )'  )
```

In this next example, the commit operation is performed after four hours with two commit cycles.

```
CALL SYSPROC.SYSTS_ALTER(  'USRT002', 'INDEX2' ,' UPDATE FREQUENCY NONE
UPDATE MINIMUM 1 INDEX CONFIGURATION (   UPDATEAUTOCOMMIT 4, COMMITTYPE HOURS,
COMMITCYCLES 2 )'  )
```

In this example, the commit operation is performed after four hours with no restriction on the number of commit cycles.

```
CALL SYSPROC.SYSTS_ALTER(  'USRT002', 'INDEX2' ,' UPDATE FREQUENCY NONE
UPDATE MINIMUM 1 INDEX CONFIGURATION (   UPDATEAUTOCOMMIT 4, COMMITTYPE HOURS )'  )
```

In this final example, the commit operation is performed after four hours with two commit cycles.

```
CALL SYSPROC.SYSTS_ALTER(  'USRT002', 'INDEX2' ,' UPDATE FREQUENCY NONE
UPDATE MINIMUM 1 INDEX CONFIGURATION (   UPDATEAUTOCOMMIT 4, COMMITTYPE HOURS,
COMMITCYCLES 2 )'  )
```

Related tasks

“Modifying the options of a text search index” on page 31
The Db2 database administrator runs a program that calls the SYSPROC.SYSTS_ALTER stored procedure to modify the options of an existing text search index.

SYSPROC.SYSTS_CREATE

Invoke the SYSPROC.SYSTS_CREATE stored procedure to enable a text column for text search indexing, which allows the text search index to be used in SQL queries that contain the CONTAINS function or the SCORE function.

The physical text search index is created on one of the text search servers that is listed in the SYSIBMTS.SYSTEXTSERVERS table. Because updating a text search index is an extensive operation, the text search index that is maintained on the text search server is not updated synchronously when the Db2 table is updated. Instead, changes to the Db2 table column are captured by triggers to a local log table.

Tip: Consider the following tips when using this stored procedure:

- You can prevent a text search server from being chosen by updating the STATUS column of the text search server entry in the SYSIBMTS.SYSTEXTSERVERS table to a value of 1. Only servers with a STATUS value of 0 (zero) are selected for creating a new text search index. Changing the STATUS value also affects the text search servers that are considered by the SYSPROC.SYSTS_TAKEOVER stored procedure. Other administration stored procedures are not affected by the STATUS value.
- This stored procedure defines only the text search index. The text search index does not contain any data until after the first invocation of the SYSPROC.SYSTS_UPDATE stored procedure for the new text search index. You should create the text search index after the table is initially populated. By creating the text search index after the table is initially populated, you prevent the firing of change triggers before an initial index update.
- To get the best performance for the CONTAINS and SCORE functions, the table that is specified in the SYSPROC.SYSTS_CREATE stored procedure must have an index on its ROWID column.
- If you create the ROWID column in your text table with the option NOT NULL GENERATED BY DEFAULT, Db2 automatically creates an implicit index on the ROWID column. If you create the ROWID column with the option NOT NULL GENERATED ALWAYS, Db2 does not create an implicit index on the ROWID column.

Note: All stored procedures can perform COMMIT statements on the current connection. This capability makes it possible to keep changes on the text search server consistent with your Db2 subsystem. After you invoke a stored procedure, you cannot roll back the changes that are made on your Db2 subsystem or the text search server. Rollback in a stored procedure is done only to a savepoint that is set when you invoke the stored procedure.

Prerequisites

Before you invoke the SYSPROC.SYSTS_CREATE stored procedure, verify the following prerequisites:

- Db2 text search functionality was started by invoking the SYSPROC.SYSTS_START stored procedure.
- The table includes a column that is defined as ROWID column.
- The SYSIBMTS.SYSTEXTSERVERS table contains at least one entry, and the DB2ENCRYPTEDPW column in the SYSIBMTS.SYSTEXTSERVERS table has a valid, encrypted password value for the user ID that the text search server uses to connect to Db2 for z/OS.
- At least one text search server is running.
- The distributed data facility (DDF) must be started, even if you call the stored procedure locally on your z/OS system. The DDF is required to allow the text search server to connect to Db2 to store a text search index.

Authorization

The user ID under which this stored procedure is invoked must have the following privileges:

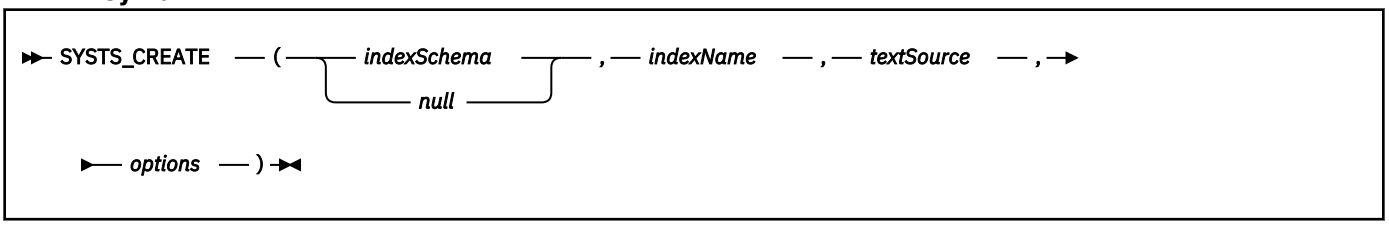
- EXECUTE on the procedure
- EXECUTE on the packages SYSIBMTS.*
- SELECT, INDEX, and TRIGGER privileges on the table that the text search index is created on
- CREATEIN privilege on the schema "*"
- DBADM authority on database SYSIBMTS

Alternatively, for DB2 10 and later, the user ID under which this stored procedure is invoked could have the following privileges instead of the privileges listed previously:

- EXECUTE on the procedure
- EXECUTE on the packages SYSIBMTS.*
- SELECT, INDEX, and TRIGGER privileges on the table that the text search index is created on
- DBADM WITHOUT ACCESSCTRL WITHOUT DATAACCESS

In addition, the user ID that is listed in the SYSIBMTS.SYSTEXTCONNECTINFO table, or that is obtained from the data source property at the text search server, must have SELECT, INSERT, UPDATE, and DELETE privileges for the text search index table after connecting to Db2 for z/OS on a T4 Java connection and going through the DRDA connect processing. The database name is SYSIBMTS. The index table name is SYSIBMTS.INDEX_[n], where [n] is the index ID according to the INDEXID column of the SYSIBMTS.SYSTEXTINDEXES administration table.

Syntax



The schema qualifier is SYSPROC.

Parameters

indexSchema

Identifies the schema of the text search index. If this parameter is null, the value of the CURRENT SCHEMA special register for the invoker is used.

Recommendation: Use a valid SQL name for this parameter.

The data type of this parameter is VARCHAR(128).

indexName
Identifies the name of the text search index. The name of the text search index together with the index schema uniquely identifies the text search index in the Db2 subsystem. You must specify a non-null value for this parameter.

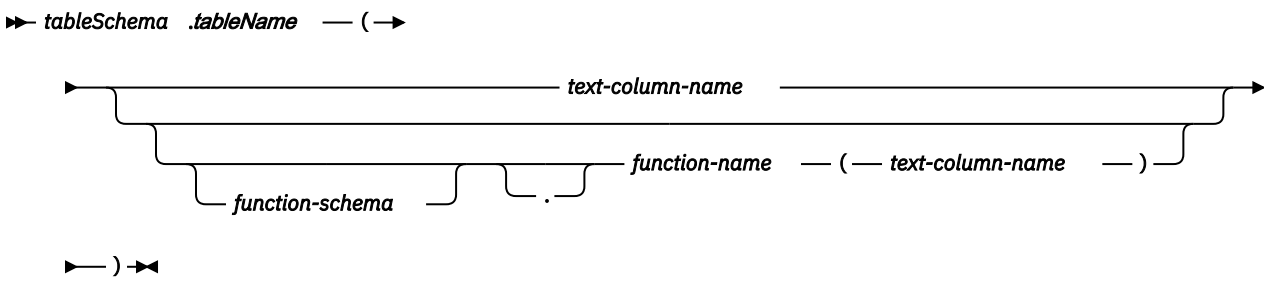
Recommendation: Use a valid SQL name for this parameter.

The data type for this parameter is VARCHAR(128).

textSource
Identifies the table and column specification for the document text source. This parameter can include user-defined functions. You must specify a non-null value for this parameter.

The data type for this parameter is VARCHAR(1024).

textSource



tableSchema
Identifies the schema of the table that the text search index is created on.

tableName
Identifies the name of the text table that contains the column that the external text search index is created on.

Restriction: Views are not supported.

text-column-name
Identifies the name of the column that contains the text that is used for creating the text search index. This column must be a string data type or XML. If the data type of the column is not one of these data types, you can specify an external function that provides access to the text document that you want to index.

Restriction: Only one text search index is allowed for a column. If a text search index already exists for the column, SQLCODE-20427 is returned.

function-schema. function-name
Identifies the schema and the name of a built-in or user-defined function that is to be used by the Db2 for z/OS text search feature to access text documents that are in a column that is not of a supported data type, or that are stored elsewhere. The function has one input parameter for the text column data type (for example, an integer that serves as a foreign key to the document content in another table), and it returns a value of one of the supported data types. The function transforms the text column content to the indexed document content.

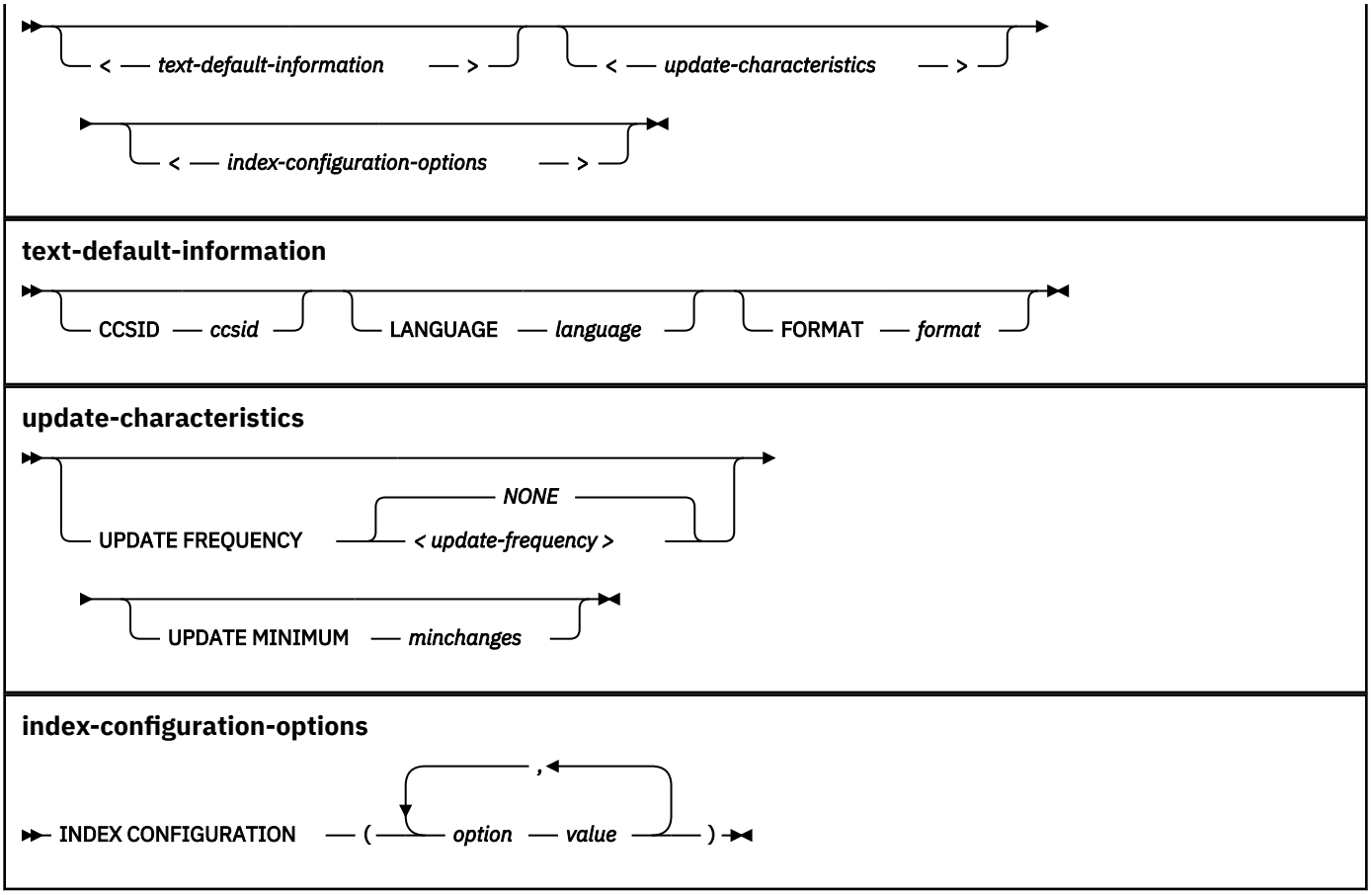
Tip: For optimal performance, specify the ALLOW PARALLEL option when the function is created.

Restriction: Cast functions and functions with more than one argument are not allowed.

options
A character string that specifies the various options that are available for this stored procedure.

The data type for this parameter is VARCHAR(32000).

options



text-default-information

Specifies the coded character set identifier that is used when indexing binary text documents, the language that is used when processing documents, and the format of text documents in the column.

CCSID *ccsid*

Specifies the coded character set identifier that is used for a text search index on a column with a binary data type. The default value is 1208 and is taken from the SYSIBMTS.SYSTEXTDEFAULTS table. All of the CCSIDs that are supported for conversion to UTF-8 by z/OS Unicode Conversion Services are allowed for this parameter.

This parameter is ignored for a text search index on a column with a non-binary data type because text columns inherit the CCSID from the table specification. The *format* value INSO lets the text search server determine the format. In this case, the *ccsid* value is ignored, however *ccsid* must be a valid input if it is specified in SYSTS_CREATE stored procedure. If the text search server cannot determine the document format, then a document error is noted in the events table during processing by the SYSPROC.SYSTS_UPDATE stored procedure.

LANGUAGE *language*

Specifies the language that the text search server uses for the linguistic processing of text documents. The default value is en_US (English). If you specify the value for this parameter as AUTO, the text search server tries to determine the language.

Important: If the language of the documents is not English, do not use the default value of en_US. Change the value to the language of the documents; otherwise, linguistic processing does not work as expected.

FORMAT *format*

Identifies the format of text documents in the column, such as HTML. The text search server needs to know the format, or content type, of the text documents that you intend to index and search. If you do not specify the **FORMAT** parameter, the default value is taken from the

FORMAT column in the SYSIBMTS.SYSTEXTDEFAULTS table. The supported *format* values are TEXT, HTML, XML, and INSO.

An XML column is always interpreted always as format XML, ignoring any value that you specify and ignoring the default value.

The *format* value INSO lets the text search server determine the format. In this case, the *ccsid* value is ignored; however, the ccsid must be a valid input if it is specified in SYSTS_CREATE stored procedure. If the text search server cannot determine the document format, then a document error is noted in the events table during processing by the SYSPROC.SYSTS_UPDATE stored procedure.

update-characteristics

Specifies the frequency of updates to the text search index and the minimum number of changes to text documents before the text search index is updated incrementally at the specified time.

UPDATE FREQUENCY ***update-frequency***

Specifies when to make updates to the text search index. The default value is NONE. This option might be useful for a text column in which no further changes are planned.

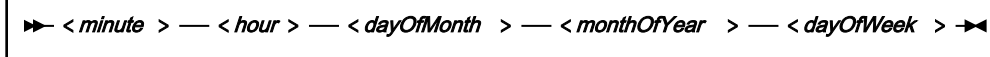
Text search index updates are not performed automatically. The scheduling of update requests is the responsibility of the Db2 for z/OS database administrator.

For each text search index, the UPDATE FREQUENCY value is stored as data type VARCHAR in the UPDATEFREQUENCY column of the SYSIBMTS.SYSTEXTINDEXES administration table. The value is stored in UNIX cron format.

To activate automatic text search index updates, schedule an index update task in Db2 for z/OS for each text search index. This task must call the SYSPROC.SYSTS_UPDATE stored procedure.

To schedule an index update task in Db2 for z/OS, you can use the Db2 administrative task scheduler. Call the SYSPROC.ADMIN_TASK_ADD stored procedure and provide the UPDATEFREQUENCY value from the SYSIBMTS.SYSTEXTINDEXES administration table as the input parameter **point-in-time**.

update-frequency



The format of the *update-frequency* option is a list of the five values separated by a blank space. The five values represent the minutes, the hours, the days of the month, the months of the year, and the days of the week beginning with Sunday.

If you specify an interval of values, or an asterisk (*), you can use a forward slash (/) at the end of the defined interval to specify a step value for this interval.

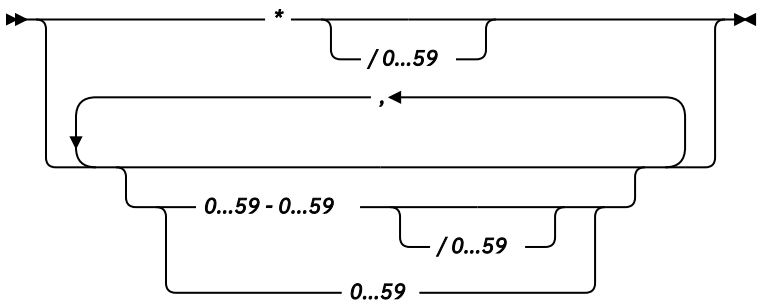
Example: This example specifies that the index update is to run every quarter hour (`0,15,30,45`) on the even hours between 8 a.m. and 6:45 p.m. (`8-18/2` is equivalent to `8,10,12,14,16,18`), from Monday to Friday every month of the year (`* * 1-5`).

```
0,15,30,45 8-18/2 * * 1-5
```

minute

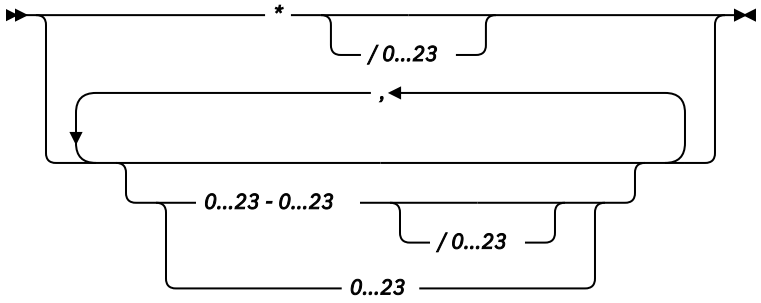
Specifies the minutes of the hour when the text search index is to be updated. You can specify an asterisk (*) for an interval of every five minutes, or you can specify an integer from 0 (zero) through 59. You cannot repeat values. The minimum update frequency is five minutes. A value of 1,4,8 is an invalid interval.

update-frequency (minute)



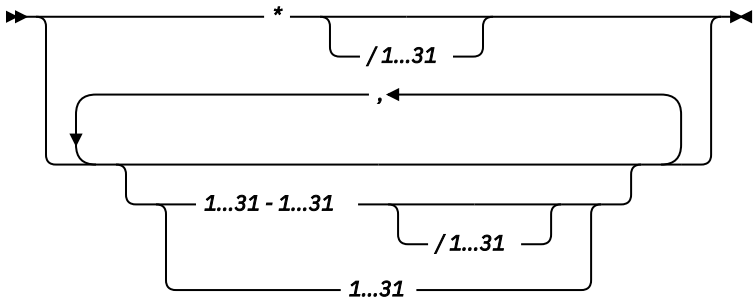
hour
Specifies the hours of the day when the text search index is to be updated. You can specify an asterisk (*) for every hour, or you can specify an integer from 0 (zero) through 23. You cannot repeat values.

update-frequency (hour)



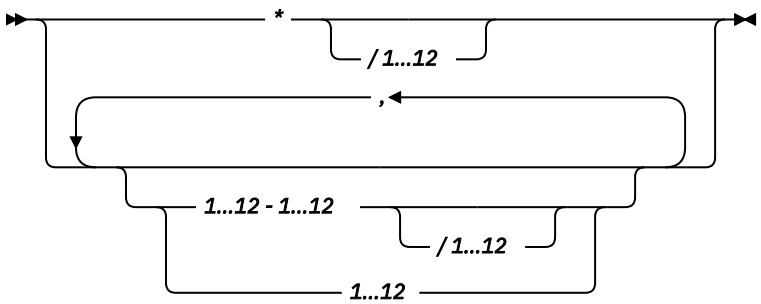
dayOfMonth
Specifies the days of the month when the text search index is to be updated. You can specify an asterisk (*) for every day, or you can specify an integer from 1 through 31. You cannot repeat values.

update-frequency (dayOfMonth)



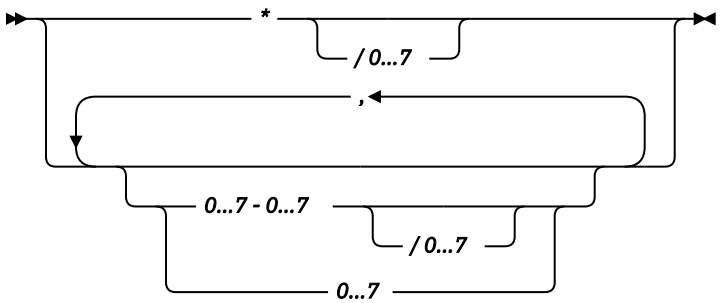
monthOfYear
Specifies the months of the year when the text search index is to be updated. You can specify an asterisk (*) for every month, or you can specify an integer from 1 through 12. You cannot repeat values.

update-frequency (monthOfYear)



dayOfWeek
Specifies the days of the week when the text search index is to be updated. You can specify an asterisk (*) for every day, or you can specify an integer from 0 (zero) through 7. Both 0 and 7 are valid values for Sunday. You cannot repeat values.

update-frequency (dayOfWeek)



UPDATE MINIMUM ***minchanges***
Specifies the minimum number of changes that are made to text documents before the text search index is updated incrementally at the time specified in the *update-frequency* option. The value must be an integer between 1 and 2147483647. The default value is taken from the UPDATEMINIMUM column in the SYSIBMTS.SYSTEXTDEFAULTS table.

This option is ignored when you update the text search index, unless you specify the USING UPDATE MINIMUM option in the SYSIBMTS.SYSTS_UPDATE stored procedure.

index-configuration-options
Specifies additional index-specific values as option value pairs. You must enclose string values in single quotation marks. A single quotation mark within a string value must be represented by two consecutive single quotation marks.

COMMENT
Specifies a comment that is stored in the REMARKS column of the SYSIBMTS.SYSTEXTINDEXES administration table and as the description of the text search server collection.

The value for this option is a string value that is less than or equal to 512 bytes.

IGNOREEMPTYDOCS
Specifies whether to represent empty documents (documents with an empty string or a null value) in the text search index.

The supported values for this option are 0 (zero) and 1. The default value is 1.

If this option is set to 1, empty documents are not represented in the text search index. If you use this option and change the document content to empty, the next incremental update deletes the documents from the text search index.

UPDATEAUTOCOMMIT
Specifies how often a commit operation is performed when fetching documents during an index update. A value of 0 (zero) means that a commit operation occurs only at the end of processing. The default value is 100 and COMMITTYPE specified as ROWS.

UPDATEAUTOCOMMIT requires a numeric value. This value represents either a number of rows or a number of hours, depending on the specification for COMMITTYPE. If COMMITTYPE is set to HOURS, the value cannot exceed 24. The text index is committed when the value of UPDATEAUTOCOMMIT is reached.

If update processing takes a very long time and Db2 active logs are small, consider using the default value. Otherwise, the active log entries of customer transactions that run in parallel to the text search index update cannot be cleared and might lead to a full the active log.

COMMITTYPE
Specifies either ROWS or HOURS. If you specify ROWS, the text index is committed after the number of rows that is specified by UPDATEAUTOCOMMIT is reached. If you specify HOURS, the text index is committed after the period of time that is specified by UPDATEAUTOCOMMIT is reached. The default value is ROWS.

You must use this option in conjunction with UPDATEAUTOCOMMIT or COMMITCYCLES.

COMMITCYCLES
Specifies a numeric value for the number of commit cycles for the processing of an index update. By default, the number of commit cycles is unrestricted. If COMMITCYCLES is 0 (zero), the update process uses as many cycles as needed to finish processing.

You must use this option in conjunction with UPDATEAUTOCOMMIT and COMMITTYPE.

SERVER
Specifies the value for the server on which to create an index.

The value can be an integer specified as the SERVERID in the SYSIBMTS.SYSTEXSTSERVERS table. There is no default value for this option. This option is valid with the SYSTS_CREATE stored procedure only.

UPDATEWITHBACKUP
Specifies whether the text search index is backed up.

The supported values for this option are 0 (zero) and 1. The default value is 1.

The value 1 specifies that the text search index is backed up in the index table during the processing of the SYSIBMTS.SYSTS_UPDATE stored procedure.

The value 0 (zero) specifies that the text search index is not backed up in the index table during processing of the SYSIBMTS.SYSTS_UPDATE stored procedure.

If the value is changed from 0 to 1, you must use the SYSIBMTS.SYSTS_UPDATE stored procedure with the ALLROWS option to save the current index in the backup table.

Default values for the *options* parameter

When you enable text search support for Db2 for z/OS, the SYSIBMTS.SYSTEXTDEFAULTS table is created and populated with default values for the *options* parameter of the SYSPROC.SYSTS_CREATE stored procedure.

The following table lists the options, default values, and descriptions of the options.

Table 6. Default values for the options parameter

Option	Default value	Description
CCSID	1208	Specifies the coded character set identifier that is used when binary text documents are indexed.

Table 6. Default values for the options parameter (continued)

Option	Default value	Description
LANGUAGE	en_US	Specifies the language that the text search server uses to process text documents.
FORMAT	TEXT	Identifies the format of text documents in the column. The default format is plain text, except for XML columns, which are always identified as XML.
UPDATEFREQUENCY	NONE	RESERVED. (Indicates that no updates are scheduled.)
UPDATEMINIMUM	1	If at least one document changed since the last index update, the SYSPROC.SYSTS_UPDATE stored procedure starts processing.
IGNOREEMPTYDOCS	1	Specifies that empty documents (documents with an empty string or a null value) are not represented in the text search index. The metadata fields for these documents are not available for search.
UPDATEAUTOCOMMIT	100 and COMMITTYPE as ROWS	Specifies how often a commit operation is performed when documents are fetched during an index update. If update processing takes a very long time and Db2 active logs are small, consider using the default value. Otherwise, the active log entries of customer transactions that run in parallel to the text search index update cannot be cleared and might lead to a full active log.
COMMITTYPE	ROWS	Specifies whether the value for UPDATEAUTOCOMMIT represents a number of rows or a number of hours.
COMMITCYCLES	0	Specifies the number of commit cycles for the processing of an index update.
MINIMUMUPDATEINTERVAL	5	Specifies the intervals for the UPDATEFREQUENCY option. Intervals cannot be shorter than five minutes.
USEREXITTHREADS	0	Reserved
UPDATEWITHBACKUP	1	Specifies that the Text Search index is backed during the processing of SYSTS_UPDATE.

Examples

Example 1: This example shows how to use the Db2 CALL statement to invoke the SYSPROC.SYSTS_CREATE stored procedure. This example uses the SYSPROC.SYSTS_CREATE stored procedure to create a text column for text search indexing, which allows the text search index to be used in SQL queries that contain the CONTAINS function or the SCORE function.

```
CALL SYSPROC.SYSTS_CREATE('SCHEMA1','IVC2','SYSADM.T1(VC2)','')
```

This example uses the following parameters:

- 'SCHEMA1' = *indexSchema*
- 'IVC2' = *indexName*
- 'SYSADM.T1(VC2)' = *textSource*
- ' ' = *options*

The *options* parameter is null by default. The values in the SYSIBMTS.SYSTEXTDEFAULTS table are used as the default value when the *options* parameter is null. The SYSIBMTS.SYSTEXTDEFAULTS table is created and populated with default values for the *options* parameter of the SYSPROC.SYSTS_CREATE stored procedure.

Example 2: This example shows how to use the Db2 CALL statement to invoke the SYSPROC.SYSTS_CREATE stored procedure with the index configuration SERVER option.

```
call sysproc.systs_create('SCHEMA1', 'IVC3', 'SYSADM.T1(VC3)',
'INDEX CONFIGURATION( SERVER 1 )')
```

This example uses the following parameters:

- 'SCHEMA1' = *indexSchema*
- 'IVC3' = *indexName*
- 'SYSADM.T1(VC3)' = *textSource*
- 'INDEX CONFIGURATION(SERVER 1)' = *options*

Example 3: This example shows how to use the Db2 CALL statement to invoke the SYSPROC.SYSTS_CREATE stored procedure with the index configuration UPDATEWITHBACKUP option.

```
call sysproc.systs_create('SCHEMA1', 'IVC4', 'SYSADM.T1(VC4)',
'INDEX CONFIGURATION( UPDATEWITHBACKUP 0 )')
```

This example uses the following parameters:

- 'SCHEMA1' = *indexSchema*
- 'IVC4' = *indexName*
- 'SYSADM.T1(VC4)' = *textSource*
- 'INDEX CONFIGURATION(UPDATEWITHBACKUP 0)' = *options*

Example 4: The following examples show how you can use the UPDATEAUTOCOMMIT, COMMITTYPE, and COMMITCYCLES options to specify how often a commit operation is performed when documents are fetched during an index update. In the first example, the commit operation is performed after two hours with two commit cycles. In the second example, the commit operation is performed after two hours with no restriction on the number of commit cycles.

```
CALL SYSPROC.SYSTS_CREATE('USRT002', 'ITEST', 'USRT002.TEST(TEXT)',
 'INDEX CONFIGURATION ( UPDATEAUTOCOMMIT 2, COMMITTYPE HOURS ,COMMITCYCLES 2)' )
```

```
CALL SYSPROC.SYSTS_CREATE('USRT002', 'ITEST', 'USRT002.TEST(TEXT)',
 'INDEX CONFIGURATION ( UPDATEAUTOCOMMIT 2, COMMITTYPE HOURS )' )
```

Related concepts

“Supported data types” on page 7
The data in the text columns that you want to index and search can be either binary data or character data.

“Supported document formats” on page 6

The text column data can be plain text, an HTML document, an XML document, or any document that is recognized by the search engine.

Related tasks

"Retrieving messages without truncation" on page 64
You can retrieve long message text from the CONTAINS or SCORE functions and from the text search administration stored procedures.

"Checking the event table after calling SYSPROC.SYSTS_CREATE" on page 66
To get the best performance for the CONTAINS and SCORE functions, the table specified in the SYSTS_CREATE stored procedure must have an index on its ROWID column.

"Scheduling the SYSPROC.SYSTS_UPDATE stored procedure " on page 63
You can use the Db2 administrative task scheduler to call the SYSPROC.SYSTS_UPDATE stored procedure according to a specific schedule.

Related reference

"Supported languages" on page 8
You can specify that text documents be processed by using a specific language. The text search feature supports the linguistic processing of text documents by 26 different language codes.

"SYSPROC.SYSTS_UPDATE " on page 59
Invoke the SYSPROC.SYSTS_UPDATE stored procedure to update the text search index to reflect the current contents of the text column that the text search index is associated with.

SYSPROC.SYSTS_DROP

Invoke the SYSPROC.SYSTS_DROP stored procedure to drop a text search index that was defined by using the SYSPROC.SYSTS_CREATE stored procedure.

Be aware that dropping the table in Db2 does not drop a text search index. You must drop a text search index by using the SYSPROC.SYSTS_DROP stored procedure either before or after dropping the table. The table does not need to exist to invoke this stored procedure; therefore, you can invoke this stored procedure after a table is dropped.

If the text search server is not reachable, the collection on the text search server might become orphaned at the server. If that happens, the collection needs to be deleted manually. This stored procedure writes console message DSN5001I, which contains the name of the collection that needs to be deleted. When the server is available again, use the Administration Tool to delete the collection at the server.

Note: All stored procedures can perform COMMIT statements on the current connection. This capability makes it possible to keep changes on the text search server consistent with your Db2 subsystem. After you invoke a stored procedure, you cannot roll back the changes that are made on your Db2 subsystem or the text search server. Rollback in a stored procedure is done only to a savepoint that is set when you invoke the stored procedure.

Prerequisites

Before you invoke the SYSPROC.SYSTS_DROP stored procedure, verify the following prerequisites:

- Db2 text search functionality was started by invocation of the SYSPROC.SYSTS_START stored procedure.
- The text search index was created (by invocation of the SYSPROC.SYSTS_CREATE stored procedure).
- Ensure that the following stored procedures are not running for the text search index that you want to drop: SYSPROC.SYSTS_CREATE, SYSPROC.SYSTS_UPDATE, and SYSPROC.SYSTS_DROP.
- At least one text search server is running.
- The distributed data facility (DDF) must be started, even if you call the stored procedure locally on your z/OS system. The DDF is required to allow the text search server to connect to Db2 to store a text search index.

Authorization

The user ID under which this stored procedure is invoked must have the following privileges:

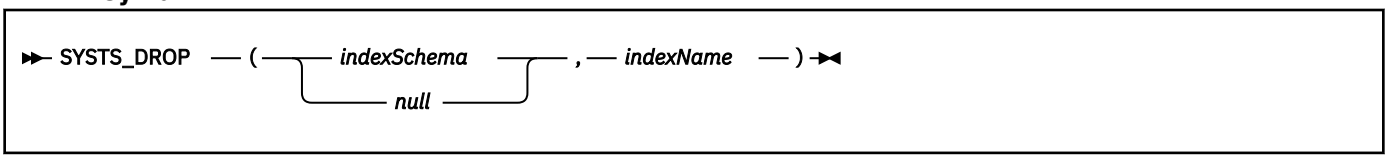
- EXECUTE on the procedure
- EXECUTE on the packages SYSIBMTS.*
- INDEX privilege on the table that the text search index is created on

In addition, if the SQL ID that calls this stored procedure is different from the SQL ID that called the SYSPROC.SYSTS_CREATE stored procedure, the authorizations to drop the following objects are required:

- The table SYSIBMTS.EVENTS_*[n]*
- The table SYSIBMTS.INDEX_*[n]*
- The table SYSIBMTS.STAGING_*[n]*
- The trigger DSNIBMTS.ISTAGING_*[n]*
- The trigger DSNIBMTS.USTAGING_*[n]*
- The trigger DSNIBMTS.DSTAGING_*[n]*

where *[n]* is the index ID according to the INDEXID column of the SYSIBMTS.SYSTEXTINDEXES administration table.

Syntax



The schema qualifier is SYSPROC.

Parameters

indexSchema
Identifies the schema of the text search index. If this parameter is null, the value of the CURRENT SCHEMA special register for the invoker is used.

Recommendation: Use a valid SQL name for this parameter.

The data type of this parameter is VARCHAR(128).

indexName
Identifies the name of the text search index. The name of the text search index together with the index schema uniquely identifies the text search index in the Db2 subsystem. You must specify a non-null value for this parameter.

Recommendation: Use a valid SQL name for this parameter.

The data type for this parameter is VARCHAR(128).

Example

This example shows how to use the Db2 CALL statement to invoke the SYSPROC.SYSTS_DROP stored procedure. This example uses the SYSPROC.SYSTS_DROP stored procedure to drop a text search index that was defined by using the SYSPROC.SYSTS_CREATE stored procedure.

```
CALL SYSPROC.SYSTS_DROP('SCHEMA1','IVC2')
```

Here is a list of the parameters used in this example.

- 'SCHEMA1' = *indexSchema*

- 'IVC2' = *indexName*

Use the SYSPROC.SYSTS_DROP stored procedure either before or after dropping the Db2 table.

Related tasks

"Retrieving messages without truncation" on page 64
You can retrieve long message text from the CONTAINS or SCORE functions and from the text search administration stored procedures.

"Dropping a table without previously calling SYSPROC.SYSTS_DROP" on page 53
When you drop a table without first calling the SYSPROC.SYSTS_DROP stored procedure, the result is orphaned full-text indexes (metadata in Db2 tables and collections on the text search server).

Dropping a table without previously calling SYSPROC.SYSTS_DROP

When you drop a table without first calling the SYSPROC.SYSTS_DROP stored procedure, the result is orphaned full-text indexes (metadata in Db2 tables and collections on the text search server).

About this task

Recreation of the identical table without previously calling the SYSPROC.SYSTS_DROP stored procedure for full-text indexes results in a situation where text search functions always return 0 results, and administration stored procedures might succeed but do not have any effect.

Procedure

To detect and delete the orphaned full-text indexes:

1. Issue the following command:

```
select indexschema,indexname from sysibmts.systextindexes I where
not exists (select 1 from sysibm.systables t where
t.name=i.tablename and t.creator=i.tableschema and
t.createdts<i.createtime);
```

2. Next, drop the orphans by calling the SYSPROC.SYSTS_DROP stored procedure.

   The SYSPROC.SYSTS_DROP stored procedure drops all of the collections on the text search servers that are installed and populated in the SYSIBMTS.SYSTEXTSERVER administration table. The text search servers must be running so that the SYSPROC.SYSTS_DROP stored procedure can drop the collections that are on them.

Related reference

"SYSPROC.SYSTS_DROP" on page 51
Invoke the SYSPROC.SYSTS_DROP stored procedure to drop a text search index that was defined by using the SYSPROC.SYSTS_CREATE stored procedure.

SYSPROC.SYSTS_RESTORE

Invoke the SYSPROC.SYSTS_RESTORE stored procedure to restore a text search index from a Db2 administration table to a text search server. After this stored procedure completes successfully, the new server is used for searching and index maintenance.

Use this stored procedure in the following two scenarios:

- Invoke this stored procedure after a Db2 restore operation that involves the text search administration tables and the tables that have been indexed for full-text. This stored procedure restores the correct version of the text search index on the text search server that is listed in the index entry (row) in the SYSIBMTS.SYSTEXTINDEXES table. In this scenario, the *serverID* parameter of SYSPROC.SYSTS_RESTORE stored procedure must be null.
- When you prepare for a planned outage of a text search server, invoke this stored procedure to copy the specified text search index to a different server. After this stored procedure completes successfully, the new server is used for text searches. In this scenario, you must not create new text search indexes on the server for which the outage is planned. You can prevent the SYSPROC.SYSTS_CREATE stored

procedure from choosing the text search server that is planned for an outage by updating the STATUS column for the server entry in the SYSIBMTS.SYSTEXTSERVERS table to a value other than zero.

The original text search server remains unchanged, and the text search index remains searchable while this stored procedure is running. If the specified text search index already exists on the target server, the text search index is replaced if the version of the collection on the text search server does not match the version in the Db2 index table.

While this stored procedure is running, the SYSPROC.SYSTS_UPDATE stored procedure, the SYSPROC.SYSTS_DROP stored procedure, and the SYSPROC.SYSTS_TAKEOVER stored procedure will fail for the specified text search index.

Note: All stored procedures can perform COMMIT statements on the current connection. This capability makes it possible to keep changes on the text search server consistent with your Db2 subsystem. After you invoke a stored procedure, you cannot roll back the changes that are made on your Db2 subsystem or the text search server. Rollback in a stored procedure is done only to a savepoint that is set when you invoke the stored procedure.

Prerequisites

Before you invoke the SYSPROC.SYSTS_RESTORE stored procedure, verify the following prerequisites:

- Db2 text search functionality was started by invocation of the SYSPROC.SYSTS_START stored procedure.
- The text search index was created (by invocation of the SYSPROC.SYSTS_CREATE stored procedure).
- The DB2ENCRYPTEDPW column in the SYSIBMTS.SYSTEXTSERVERS table has a valid, encrypted password value for the user ID that the text search server uses to connect to Db2 for z/OS.
- The following stored procedures are not running for the text search index that you want to restore: SYSPROC.SYSTS_CREATE, SYSPROC.SYSTS_UPDATE, SYSPROC.SYSTS_DROP, and SYSPROC.SYSTS_TAKEOVER.
- At least one text search server is running.
- The distributed data facility (DDF) must be started, even if you call the stored procedure locally on your z/OS system. The DDF is required to allow the text search server to connect to Db2 to store a text search index.

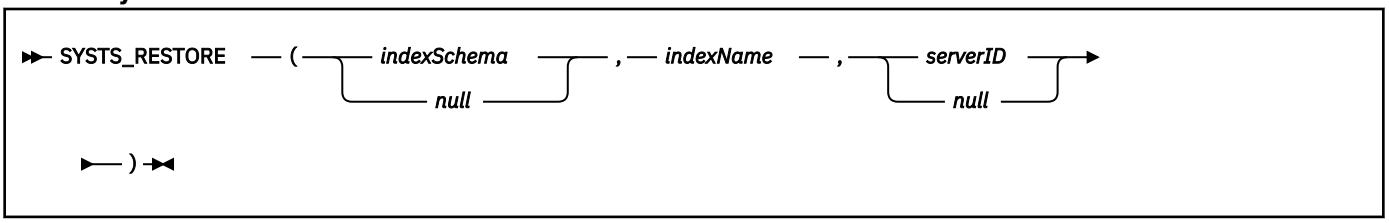
Authorization

The user ID under which this stored procedure is invoked must have the following privileges:

- EXECUTE on the procedure
- EXECUTE on the packages SYSIBMTS.*
- INDEX privilege on the table that the text search index is created on

In addition, the user ID that is listed in the SYSIBMTS.SYSTEXTCONNECTINFO table, or that is obtained from the data source property at the text search server, must have SELECT, INSERT, UPDATE, and DELETE privileges for the text search index table. These privileges must be valid for the user ID after the user ID is used to connect to Db2 for z/OS on a T4 Java connection and going through DRDA connect processing.

Syntax



The schema qualifier is SYSPROC.

Parameters

indexSchema
Identifies the schema of the text search index. If this parameter is null, the value of the CURRENT SCHEMA special register for the invoker is used.

Recommendation: Use a valid SQL name for this parameter.

The data type of this parameter is VARCHAR(128).

indexName
Identifies the name of the text search index. The name of the text search index together with the index schema uniquely identifies the text search index in the Db2 subsystem. You must specify a non-null value for this parameter.

Recommendation: Use a valid SQL name for this parameter.

The data type for this parameter is VARCHAR(128).

serverID
Identifies the server that you want to use as the new server to handle text searching and index maintenance. The server is defined by the SERVERID column in the SYSIBMTS.SYSTEXTSERVERS table.

If this parameter is null, the stored procedure uses the server ID that is listed in the SYSIBMTS.SYSTEXTINDEXES table for this search index.

Example

This example shows how to use the Db2 CALL statement to invoke the SYSPROC.SYSTS_RESTORE stored procedure. This example uses the SYSPROC.SYSTS_RESTORE stored procedure to restore a text search index from a Db2 administration table to a text search server. After this stored procedure completes successfully, the new server is used for searching and index maintenance.

```
CALL SYSPROC.SYSTS_RESTORE ('SCHEMA1','IVC2','2')
```

Here is a list of the parameters used in this example.

- 'SCHEMA1' = *indexSchema*
- 'IVC2' = *indexName*
- '2' = *serverID*

In this example the *serverID* is set to 2 so that the stored procedure restores the index to the text search server that has '2' as the serverID.

Related tasks

“Retrieving messages without truncation” on page 64
You can retrieve long message text from the CONTAINS or SCORE functions and from the text search administration stored procedures.

SYSPROC.SYSTS_START

Invoke the SYSPROC.SYSTS_START stored procedure to start Db2 text search functions.

Text search functions include support for SQL queries that use the CONTAINS function, the SCORE function, and the administration stored procedures that are used to maintain text search indexes. If text search functions are not started, Db2 returns SQLCODE -20424 with reason code 4 for the CONTAINS and SCORE functions.

Run the SYSPROC.SYSTS_START stored procedure, each time that a server is added or changed in SYSIBMTS.SYSTEXTSERVERS table. After this stored procedure runs, the start status is made persistent in the SYSIBMTS.SYSTEXTSTATUS table, even if errors occur.

For the text search servers that are contained in the SYSIBMTS.SYSTEXTSERVERS table, TCP/IP names are resolved. Multiple calls of the SYSPROC.SYSTS_START stored procedure are not considered an error because you might need to refresh the address resolution in the SYSIBMTS.SYSTEXTSERVERS table.

Note: All stored procedures can perform COMMIT statements on the current connection. This capability makes it possible to keep changes on the text search server consistent with your Db2 subsystem. After you invoke a stored procedure, you cannot roll back the changes that are made on your Db2 subsystem or the text search server. Rollback in a stored procedure is done only to a savepoint that is set when you invoke the stored procedure.

Prerequisites

Before you invoke the SYSPROC.SYSTS_START stored procedure, verify the following prerequisites:

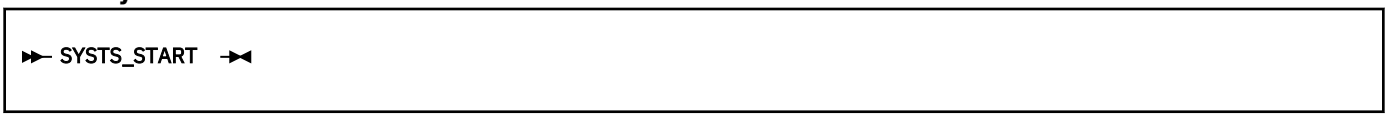
- The SYSIBMTS.SYSTEXTSERVERS table contains at least one entry.
- The SYSIBMTS.SYSTEXTCONNECTINFO table is populated.
- At least one text search server is running.
- The distributed data facility (DDF) must be started, even if you call the stored procedure locally on your z/OS system. The DDF is required to allow the text search server to connect to Db2 to store a text search index.

Authorization

The user ID under which this stored procedure is invoked must have the following privileges:

- EXECUTE on the procedure
- EXECUTE on the packages SYSIBMTS.*

Syntax



The schema qualifier is SYSPROC.

Example

This example shows how to use the Db2 CALL statement to invoke the SYSPROC.SYSTS_START stored procedure. This example uses the SYSPROC.SYSTS_START stored procedure to start Db2 text search functions.

```
CALL SYSPROC.SYSTS_START()
```

Related tasks

“Retrieving messages without truncation” on page 64
You can retrieve long message text from the CONTAINS or SCORE functions and from the text search administration stored procedures.

SYSPROC.SYSTS_STOP

Invoke the SYSPROC.SYSTS_STOP stored procedure to stop Db2 text search functions.

After this stored procedure runs successfully, SQL queries that use the CONTAINS function, the SCORE function, and administration stored procedures that are used for index maintenance will fail without trying to contact a text search server.

The stopped status is made persistent in the SYSIBMTS.SYSTEXTSTATUS table, even if errors occur.

After you invoke the SYSPROC.SYSTS_STOP stored procedure, invocations of the SYSPROC.SYSTS_UPDATE stored procedure fail until you invoke the SYSPROC.SYSTS_START stored procedure again. Updates that are made to columns with a text search index continue to be captured by Db2, even while the SYSPROC.SYSTS_STOP stored procedure is running. SQL queries and administration stored procedures that are currently running are not affected.

Note: All stored procedures can perform COMMIT statements on the current connection. This capability makes it possible to keep changes on the text search server consistent with your Db2 subsystem. After you invoke a stored procedure, you cannot roll back the changes that are made on your Db2 subsystem or the text search server. Rollback in a stored procedure is done only to a savepoint that is set when you invoke the stored procedure.

Prerequisite

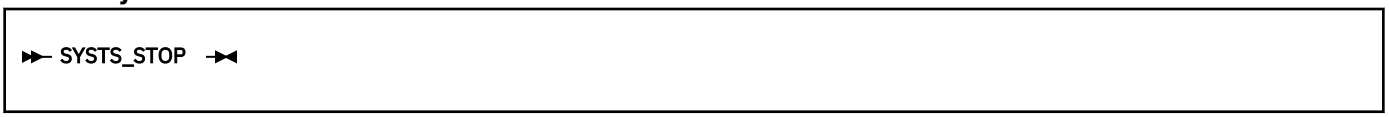
The distributed data facility (DDF) must be started, even if you call the stored procedure locally on your z/OS system. The DDF is required to allow the text search server to connect to Db2 to store a text search index.

Authorization

The user ID under which this stored procedure is invoked must have the following privileges:

- EXECUTE on the procedure
- EXECUTE on the packages SYSIBMTS.*

Syntax



The schema qualifier is SYSPROC.

Example

This example shows how to use the Db2 CALL statement to invoke the SYSPROC.SYSTS_STOP stored procedure. This example uses the SYSPROC.SYSTS_STOP stored procedure to stop Db2 text search functions.

```
CALL SYSPROC.SYSTS_STOP()
```

Related tasks

"Retrieving messages without truncation" on page 64
You can retrieve long message text from the CONTAINS or SCORE functions and from the text search administration stored procedures.

SYSPROC.SYSTS_TAKEOVER

Invoke the SYSPROC.SYSTS_TAKEOVER stored procedure after encountering problems contacting a text search server during search operations.

Usually, such problems are indicated by console message DSN5001I. If the problem persists, this stored procedure chooses a different, available text search server and restores the text search index to that server.

This stored procedure checks whether the text search server can be contacted for the specified text search index by trying several connection attempts. If the text search index on the text search server is not available, this stored procedure chooses any available server from the SYSIBMTS.SYSTEXTSERVERS table and tries to take over the text search index on the new server. If the attempt to take over a text

search index fails, the SYSPROC.SYSTS_TAKEOVER stored procedure selects the next server that is listed in the SYSIBMTS.SYSTEXTSERVERS table and tries again.

After the stored procedure completes successfully, the new server is used for searching and index maintenance. The original text search server remains unchanged. If the stored procedure was unsuccessful on all servers (or if a second server is not listed in the SYSIBMTS.SYSTEXTSERVERS table), this stored procedure reports an error.

The SYSPROC.SYSTS_UPDATE stored procedure, the SYSPROC.SYSTS_DROP stored procedure, and the SYSPROC.SYSTS_RESTORE stored procedure will fail for the specified text search index while this stored procedure is running.

Note: All stored procedures can perform COMMIT statements on the current connection. This capability makes it possible to keep changes on the text search server consistent with your Db2 subsystem. After you invoke a stored procedure, you cannot roll back the changes that are made on your Db2 subsystem or the text search server. Rollback in a stored procedure is done only to a savepoint that is set when you invoke the stored procedure.

Prerequisites

Before you invoke the SYSPROC.SYSTS_TAKEOVER stored procedure, verify the following prerequisites:

- Db2 text search functionality was started by invocation of the SYSPROC.SYSTS_START stored procedure.
- The text search index was created (by invocation of the SYSPROC.SYSTS_CREATE stored procedure).
- The DB2ENCRYPTEDPW column in the SYSIBMTS.SYSTEXTSERVERS table has a valid, encrypted password value for the user ID that the text search server uses to connect to Db2 for z/OS.
- The following stored procedures are not running for the index that you want to take over: SYSPROC.SYSTS_CREATE, SYSPROC.SYSTS_UPDATE, SYSPROC.SYSTS_DROP, and SYSPROC.SYSTS_RESTORE.
- At least one text search server is running.
- The distributed data facility (DDF) must be started, even if you call the stored procedure locally on your z/OS system. The DDF is required to allow the text search server to connect to Db2 to store a text search index.

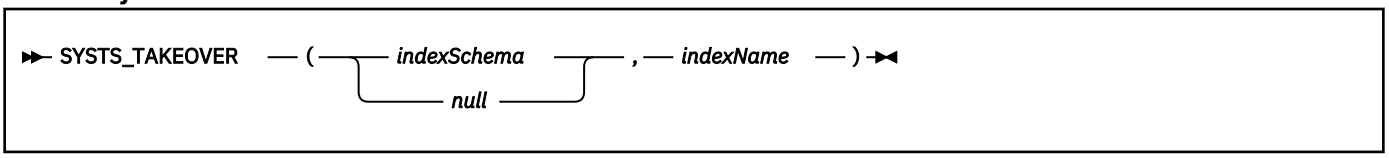
Authorization

The user ID under which this stored procedure is invoked must have the following privileges:

- EXECUTE on the procedure
- EXECUTE on the packages SYSIBMTS.*
- INDEX privilege on the table that the text search index is created on

In addition, the user ID that is listed in the SYSIBMTS.SYSTEXTCONNECTINFO table, or that is obtained from the data source property at the text search server, must have SELECT, INSERT, UPDATE, and DELETE privileges for the text search index table. These privileges must be valid for the user ID after the user ID is used to connect to Db2 for z/OS on a T4 Java connection and going through DRDA connect processing.

Syntax



The schema qualifier is SYSPROC.

Parameters

indexSchema
Identifies the schema of the text search index. If this parameter is null, the value of the CURRENT SCHEMA special register for the invoker is used.

Recommendation: Use a valid SQL name for this parameter.

The data type of this parameter is VARCHAR(128).

indexName
Identifies the name of the text search index. The name of the text search index together with the index schema uniquely identifies the text search index in the Db2 subsystem. You must specify a non-null value for this parameter.

Recommendation: Use a valid SQL name for this parameter.

The data type for this parameter is VARCHAR(128).

Example

This example shows how to use the Db2 CALL statement to invoke the SYSPROC.SYSTS_TAKEOVER stored procedure. This example uses the SYSPROC.SYSTS_TAKEOVER stored procedure to choose a different, available text search server and restore the text search index to the available server.

```
CALL SYSPROC.SYSTS_TAKEOVER('SCHEMA1','IVC2')
```

Here is a list of the parameters used in this example.

- 'SCHEMA1' = *indexSchema*
- 'IVC2' = *indexName*

Related tasks

“Retrieving messages without truncation” on page 64
You can retrieve long message text from the CONTAINS or SCORE functions and from the text search administration stored procedures.

SYSPROC.SYSTS_UPDATE

Invoke the SYSPROC.SYSTS_UPDATE stored procedure to update the text search index to reflect the current contents of the text column that the text search index is associated with.

After the stored procedure completes updating the text search index, the collection data from the text search server is backed up to a Db2 table.

Because updating a text search index is an extensive operation, the text search index that is maintained on the text search server is not updated synchronously when the Db2 table is updated. Instead, changes to the Db2 table column are captured by a trigger to a local log table. The text search index on the text search server is updated the next time that you invoke the SYSPROC.SYSTS_UPDATE stored procedure. Therefore, some search requests might not reflect recent updates to the table.

This stored procedure returns only after all of the update processing for the text search index on the text search server is completed. The duration depends on the number of new documents that are being indexed and the number of documents that are already indexed. During the update process, the text search index remains searchable.

If individual documents contain errors, you must correct the documents, update the table, and update the text search index again. You can look up the ROWIDs of the erroneous documents in the table SYSIBMTS.EVENTS_*[n]*, where *[n]* is the index ID according to the INDEXID column of the SYSIBMTS.SYSTEXTINDEXES administration table. When you change the corresponding rows in the table, the next SYSPROC.SYSTS_UPDATE stored procedure request can process these documents again.

Tip: Use the Db2 administrative task scheduler to call the SYSPROC.SYSTS_UPDATE stored procedure according to a specific schedule by setting the UPDATE FREQUENCY option in the SYSPROC.SYSTS_CREATE stored procedure. For more information about the Db2 administrative task scheduler, see Scheduling administrative tasks.

Important: The text search server might block the index update process if multiple index updates are being performed at the same time. The text search server might also block the index update process if the configuration value for the number of concurrent text search indexing subsystems is too small.

Note: All stored procedures can perform COMMIT statements on the current connection. This capability makes it possible to keep changes on the text search server consistent with your Db2 subsystem. After you invoke a stored procedure, you cannot roll back the changes that are made on your Db2 subsystem or the text search server. Rollback in a stored procedure is done only to a savepoint that is set when you invoke the stored procedure.

Prerequisites

Before you invoke the SYSPROC.SYSTS_UPDATE stored procedure, verify the following prerequisites:

- Db2 text search functionality was started by invocation of the SYSPROC.SYSTS_START stored procedure.
- The text search index was created (by invocation of the SYSPROC.SYSTS_CREATE stored procedure).
- The DB2ENCRYPTEDPW column in the SYSIBMTS.SYSTEXTSERVERS table has a valid, encrypted password value for the user ID that the text search server uses to connect to Db2 for z/OS.
- The following stored procedures are not running for the text search index that you want to update: SYSPROC.SYSTS_CREATE, SYSPROC.SYSTS_DROP, SYSPROC.SYSTS_RESTORE, and SYSPROC.SYSTS_TAKEOVER.
- At least one text search server is running.
- The distributed data facility (DDF) must be started, even if you call the stored procedure locally on your z/OS system. The DDF is required to allow the text search server to connect to Db2 to store a text search index.

Authorization

The user ID under which this stored procedure is invoked must have the following privileges:

- EXECUTE on the procedure
- EXECUTE on the packages SYSIBMTS.*
- SELECT and INDEX privileges on the table that the text search index is created on

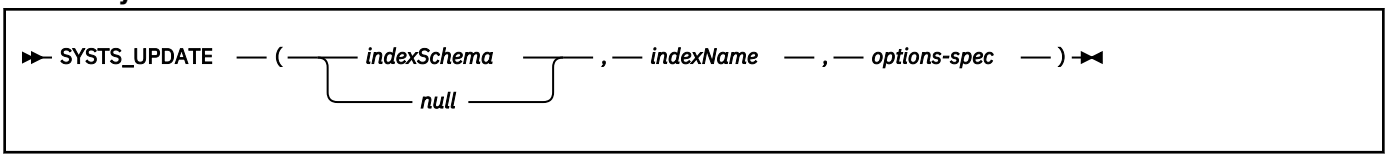
Also, if the SQL ID that calls this stored procedure is different from the SQL ID that called the SYSPROC.SYSTS_CREATE stored procedure, the following index privileges are required:

- SELECT, INSERT, UPDATE (on the column TOBEDELETED), and DELETE privileges on table SYSIBMTS.STAGING_*[n]*
- INSERT privilege on table SYSIBMTS.EVENTS_*[n]*

where *[n]* is the index ID according to the INDEXID column of the SYSIBMTS.SYSTEXTINDEXES administration table.

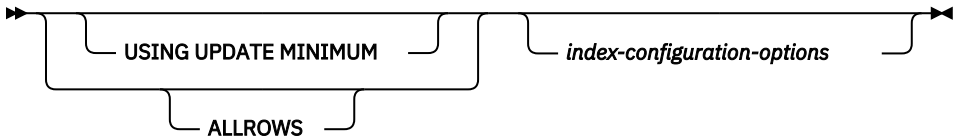
In addition, the user ID that is listed in the SYSIBMTS.SYSTEXTCONNECTINFO table, or that is obtained from the data source property at the text search server, must have SELECT, INSERT, UPDATE, and DELETE privileges for the text search index table. These privileges must be valid for the user ID after the user ID is used to connect to Db2 for z/OS on a T4 Java connection and going through DRDA connect processing. The database name is SYSIBMTS. The index table name is SYSIBMTS.INDEX_*[n]*, where *[n]* is the index ID according to the INDEXID column of the SYSIBMTS.SYSTEXTINDEXES administration table.

Syntax

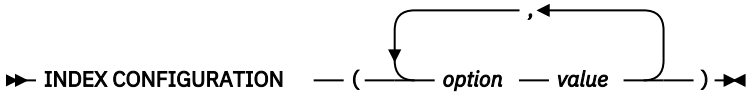


The schema qualifier is SYSPROC.

options-spec



index-configuration-options



Parameters

indexSchema
Identifies the schema of the text search index. If this parameter is null, the value of the CURRENT SCHEMA special register for the invoker is used.

Recommendation: Use a valid SQL name for this parameter.

The data type of this parameter is VARCHAR(128).

indexName
Identifies the name of the text search index. The name of the text search index together with the index schema uniquely identifies the full-text index in the Db2 subsystem. You must specify a non-null value for this parameter.

Recommendation: Use a valid SQL name for this parameter.

The data type for this parameter is VARCHAR(128).

options-spec
A character string that specifies options that are available for this stored procedure. You can specify only one option.

USING UPDATE MINIMUM
This option uses the USING UPDATE MINIMUM settings that you specified for the SYSPROC.SYSTS_CREATE stored procedure and starts an incremental update only if the specified number of changes was reached. The default is to unconditionally start the update process.

ALLROWS
This option uses ALLROWS to start an initial update. Use this option when a load replace to the base table is done, or when you need to restore data from the base table. If a synonym dictionary has been created for the text search index, using the ALLROWS option will remove the dictionary.

`index-configuration-options`
Specifies additional index-specific values as option value pairs. You must enclose string values in single quotation marks. A single quotation mark within a string value must be represented by two consecutive single quotation marks.

UPDATEAUTOCOMMIT
Specifies how often a commit operation is performed when fetching documents during an index update. A value of 0 (zero) means that a commit operation occurs only at the end of processing. The default value is 100 and COMMITTYPE specified as ROWS.

UPDATEAUTOCOMMIT requires a numeric value. This value represents either a number of rows or a number of hours, depending on the specification for COMMITTYPE. If COMMITTYPE is set to HOURS, the value cannot exceed 24. The text index is committed when the value of UPDATEAUTOCOMMIT is reached.

If update processing takes a very long time and Db2 active logs are small, consider using the default value. Otherwise, the active log entries of customer transactions that run in parallel to the text search index update cannot be cleared and might lead to a full the active log.

COMMITTYPE
Specifies either ROWS or HOURS. If you specify ROWS, the text index is committed after the number of rows that is specified by UPDATEAUTOCOMMIT is reached. If you specify HOURS, the text index is committed after the period of time that is specified by UPDATEAUTOCOMMIT is reached. The default value is ROWS.

You must use this option in conjunction with UPDATEAUTOCOMMIT or COMMITCYCLES.

COMMITCYCLES
Specifies a numeric value for the number of commit cycles for the processing of an index update. By default, the number of commit cycles is unrestricted. If COMMITCYCLES is 0 (zero), the update process uses as many cycles as needed to finish processing.

You must use this option in conjunction with UPDATEAUTOCOMMIT and COMMITTYPE.

Examples

Example 1: This example shows how to use the Db2 CALL statement to invoke the SYSPROC.SYSTS_UPDATE stored procedure. This example uses the SYSPROC.SYSTS_UPDATE stored procedure to update the text search index with the current text column that the text search index is associated with.

```
CALL SYSPROC.SYSTS_UPDATE('SCHEMA1','IVC2','')
```

Here is a list of the parameters used in this example.

- 'SCHEMA1' = *indexSchema*
- 'IVC2' = *indexName*
- ' ' = *options*

The *options* parameter is blank by default. The default is to unconditionally start the update process.

Example 2: The following examples show how you can use the UPDATEAUTOCOMMIT, COMMITTYPE, and COMMITCYCLES options to specify how often a commit operation is performed when documents are fetched during an index update. In the first example, the commit operation is performed after two hours with two commit cycles.

```
CALL SYSPROC.SYSTS_UPDATE('USRT002', 'ITEST',
 'INDEX CONFIGURATION ( UPDATEAUTOCOMMIT 2, COMMITTYPE HOURS ,COMMITCYCLES 2)')
```

In the next example, the commit operation is performed after two hours with no restriction on the number of commit cycles.

```
CALL SYSPROC.SYSTS_UPDATE('USRT002', 'ITEST',
 'INDEX CONFIGURATION ( UPDATEAUTOCOMMIT 2, COMMITTYPE HOURS )')
```

In this example, the commit operation is performed after 2000 rows with no restriction on the number of commit cycles.

```
CALL SYSPROC.SYSTS_UPDATE('USRT002', 'ITEST',
 'INDEX CONFIGURATION ( UPDATEAUTOCOMMIT 2000 )')
```

Related concepts

"Document truncation" on page 7
A text search server limits the number of Unicode characters that can be indexed for each text document. Sometimes this character limit results in the truncation of large text documents in the text search index.

Related tasks

"Retrieving messages without truncation" on page 64
You can retrieve long message text from the CONTAINS or SCORE functions and from the text search administration stored procedures.

"Scheduling the SYSPROC.SYSTS_UPDATE stored procedure " on page 63
You can use the Db2 administrative task scheduler to call the SYSPROC.SYSTS_UPDATE stored procedure according to a specific schedule.

Related reference

"SYSPROC.SYSTS_CREATE" on page 41
Invoke the SYSPROC.SYSTS_CREATE stored procedure to enable a text column for text search indexing, which allows the text search index to be used in SQL queries that contain the CONTAINS function or the SCORE function.

Scheduling the SYSPROC.SYSTS_UPDATE stored procedure

You can use the Db2 administrative task scheduler to call the SYSPROC.SYSTS_UPDATE stored procedure according to a specific schedule.

Procedure

To call the SYSPROC.SYSTS_UPDATE stored procedure according to a specific schedule:

1. Set the UPDATE FREQUENCY option in the SYSPROC.SYSTS_CREATE stored procedure.
2. To schedule the automatic index update task, call the SYSPROC.ADMIN_TASK_ADD stored procedure.
3. Provide the UPDATEFREQUENCY value from the SYSIBMTS.SYSTEXTINDEXES table directly as input parameter **`point-in-time`**.
4. Provide the string SYSTS_UPDATE as input parameter **`procedure-name`**.
5. Provide the parameters to the SYSPROC.SYSTS_UPDATE stored procedure as input parameter **`procedure-input`**.

Related reference

"SYSPROC.SYSTS_CREATE" on page 41
Invoke the SYSPROC.SYSTS_CREATE stored procedure to enable a text column for text search indexing, which allows the text search index to be used in SQL queries that contain the CONTAINS function or the SCORE function.

"SYSPROC.SYSTS_UPDATE " on page 59
Invoke the SYSPROC.SYSTS_UPDATE stored procedure to update the text search index to reflect the current contents of the text column that the text search index is associated with.

Archive log consumption

During processing of the SYSPROC.SYSTS_UPDATE stored procedure, write operations on the event and log table for a full-text index consume log space.

Plan for the size of archive logs according to the following space considerations:

- When inserting, updating, and deleting documents from a table with an associated full-text index, a trigger inserts a row to the log table for the full-text index.
- In the SYSPROC.SYSTS_UPDATE stored procedure, there is one SQL insert operation for every document that is not parsed correctly by the text search engine.
- In the SYSPROC.SYSTS_UPDATE stored procedure, there is one delete operation for every row in the log table for the full-text index.

- In the SYSPROC.SYSTS_UPDATE stored procedure, there is an additional update operation for every row in the log table, if the update is an incremental update.

Responding to an automatic takeover

If the SYSPROC.SYSTS_UPDATE stored procedure fails to update the text search index because of a problem with the text search server, a takeover is attempted.

About this task

Usually, such a problem is indicated by message DSN5001I. After attempting a takeover, the SYSPROC.SYSTS_UPDATE stored procedure returns SQLCODE -20427 and error message OF00010I. The message indicates whether a takeover on the text search index was successful.

Return code 0 indicates a successful takeover after an unsuccessful text search index update. The index is now on a new text search server but is not updated yet.

If the return code is not 0, neither the update operation or the takeover was successful. This indicates that every text search server that is listed in the SYSIBMTS.SYSTEXTSERVERS administration table has a problem.

The message text in the following example shows a return code of 0:

```
OF00010I (DSN20427E ERROR OCCURRED DURING TEXT SEARCH ADMINISTRATION STORED
PROCEDURE OF00010I Index takeover attempted during update. Return code 0.
For details see event table 'SYSIBMTS'.'EVENTS_1')
```

Procedure

Depending on the return code that is associated with this message, you must complete one of the following actions.

Option	Description
Return code 0	Call the SYSPROC.SYSTS_UPDATE stored procedure again.
Return code other than 0	a. Re-enable at least one of the text search servers. For example, if all of the servers are down, restart at least one of the servers. b. Call the SYSPROC.SYSTS_UPDATE stored procedure again.

Retrieving messages without truncation

You can retrieve long message text from the CONTAINS or SCORE functions and from the text search administration stored procedures.

About this task

Message text that is longer than 70 bytes is truncated in SQLCA and in the DSNTIAR output.

Procedure

Issue the GET DIAGNOSTICS statement.

Example

```
void sql_longmessage()
{
    EXEC SQL BEGIN DECLARE SECTION;
    char messageId[10] = {0};
    char messageText[2048] = {0};
    EXEC SQL END DECLARE SECTION;
    EXEC SQL GET DIAGNOSTICS CONDITION 1
        :messageId = DB2_MESSAGE_ID,
        :messageText = MESSAGE_TEXT;
```

```
    printf("%s %s\n", messageId, messageText);
}
```

Enabling trace for stored procedures

You can enable tracing for text search administration stored procedures by specifying the appropriate instrumentation facility component identifier (IFCID) in the START TRACE and MODIFY TRACE commands.

Procedure

Specify the appropriate IFCID for the type of trace that you want to enable.

Option	Description
For data trace:	Specify IFCID 345.
For verbose function trace:	Specify IFCID 344.

Example

For example, if you choose generalized trace facility (GTF) as the trace destination and need to trace the IFCID 345 for the SYSPROC.SYSIBMTS_UPDATE stored procedure, you can complete the following steps:

1. Start GTF as described in Recording GTF trace data.
2. Run the following Db2 command:

   ```
   -START TRACE(MON) DEST(GTF) IFCID(345)
   ```

3. Call the SYSPROC.SYSIBMTS_UPDATE stored procedure.
4. Run the following Db2 command:

   ```
   -STOP TRACE(MON) DEST(GTF)
   ```

5. Stop GTF by using the command P GTF.

Unless you specify otherwise, the trace data is written into the dataset SYS1.TRACE. This dataset might be requested by IBM Support for problem determination.

Overview of the event table

Each text search index has an event table associated with it that is created and maintained by the administration stored procedures.

The event table provides information about processing steps in stored procedures and problems that might occur during the indexing of individual documents (text column values) in the SYSPROC.SYSTS_UPDATE stored procedure. If a document cannot be indexed (for example, because the document is ill-formed XML), a row is added to the event table with details about the problem and a reference to the row in the table that the text search index was created on. This information can help to determine the cause of an indexing problem. To allow analysis of event table rows by a program, the rows are further classified by severity.

The schema and name of an event table is SYSIBMTS.EVENTS_[N], where [N] is the index identifier according to the INDEXID column of the SYSIBMTS.SYSTEXTINDEXES administration table. The [N] is unique for each text search index. To retrieve the name of an event table, you can use the following query:

```
SELECT 'SYSIBMTS.' || EVENTTABLENAME FROM SYSIBMTS.SYSTEXTINDEXES
WHERE INDEXSCHEMA = 'myschema' and INDEXNAME = 'myindex';
```

The event table for an index consists of the columns that are specified in the following table.

Table 7. Contents of the event table

Column name	Type	Length	Nulls	Description
RID	CHAR (FOR BIT DATA)	40	Y	Specifies the foreign key to the document in the user table. If null, the event is not related to a document.
OPERATION	CHAR	1	Y	Specifies a document-specific event: ‘I' (insert), ‘U' (update), and ‘D' (delete).
TIME	TIMESTAMP	10	Y	Specifies the time that the event was recorded.
SEVERITY	SMALLINT	2	Y	Specifies the severity of the event: 1 (informational), 4 (warning), 8 (error).
REASON	SMALLINT	2	Y	Specifies the same message number as in the text of the MESSAGE column.
MESSAGE	VARCHAR	1024	Y	Specifies the message text in language according to CURRENT LOCALE LC_CTYPE.

Checking the event table after calling SYSPROC.SYSTS_CREATE

To get the best performance for the CONTAINS and SCORE functions, the table specified in the SYSTS_CREATE stored procedure must have an index on its ROWID column.

About this task

If there is not a ROWID index, the SYSPROC.SYSTS_CREATE stored procedure adds an entry similar to the following in the event table:

```
OF00700E DB2 error: A DB2 performance problem occurred. You must create
an index on the ROWID column. For example, enter the command ''CREATE
UNIQUE INDEX IROWIDCKHTM ON USRT002.CKHTM(RID). File ./dsxdbcat.cpp:165''.
```

If you are not sure if an index exists on the ROWID column, you must check the event table each time after running the SYSTS_CREATE stored procedure.

Procedure

Issue the following SELECT statement:

```
SELECT * FROM SYSIBMTS.EVENTS_[N];
```

where EVENTS_[N] is the name of the event table.

Note: If you create the ROWID column with the option NOT NULL GENERATED BY DEFAULT, Db2 automatically creates an implicit index on the ROWID column. If you create the ROWID column in your text table with the option NOT NULL GENERATED ALWAYS, Db2 does not create an implicit index on the ROWID column.

Related reference

“SYSPROC.SYSTS_CREATE” on page 41

Invoke the SYSPROC.SYSTS_CREATE stored procedure to enable a text column for text search indexing, which allows the text search index to be used in SQL queries that contain the CONTAINS function or the SCORE function.

Handling single document errors

Single document errors occur when one or more text documents in a table cannot be indexed by the text search server.

About this task

In this case, the SYSPROC.SYSTS_UPDATE stored procedure returns SQLCODE +462 (SQLSTATE 01H14) and the message OF00050W:

```
DSN00462W EXTERNAL FUNCTION OR PROCEDURE SYSTS_UPDATE (SPECIFIC NAME SYSTS_UPDATE)
HAS RETURNED A WARNING SQLSTATE, WITH DIAGNOSTIC TEXT OF00050W [n] document(s)
could not be processed correctly. For details see event table
'SYSIBMTS'.'EVENTS_[N]'.
```

In addition, for each document that cannot be processed, the SYSPROC.SYSTS_UPDATE stored procedure adds an entry to the event table. The text search index does not contain these documents. Therefore, the CONTAINS or SCORE functions will not return these documents.

Usually, a single document error occurs if a document contains ill-formed content. Single document errors typically occur for the following reasons:

- The document content is damaged or truncated.
- The document format is different from the format that is specified in the SYSPROC.SYSTS_CREATE stored procedure.
- The document in a binary column is encoded in a CCSID that is different from the CCSID that is specified in the SYSPROC.SYSTS_CREATE stored procedure.
- The document size exceeds the maximum size that is supported by the text search server engine.
- The content is incorrect from the user-defined function, if the user-defined function is specified as a parameter in the SYSPROC.SYSTS_CREATE stored procedure to access the text documents indirectly.

Procedure

To find and correct single document errors and trigger re-indexing:

1. In the OF00050W message, note the name of the event table and the number [n] of documents in error.
2. Find the documents that are in error and their ROWIDs by joining with the RID column of the event table. Use the ROWID function as shown in the following example:

   ```
   SELECT HEX(T.RID), T.TEXT FROM MYTEXT T,SYSIBMTS.EVENTS_[N] E
   WHERE T.RID=ROWID(E.RID);
   ```

   where [N] is the index ID as displayed in the OF00050W message.
3. Correct the documents that are in error in the text table column by using an SQL UPDATE statement. This step is also necessary in cases where a user-defined function returns the document content. In these cases, you must also verify that the user-defined function returns the expected content when applied to the text column. (Make sure to check the CCSID and format.)
4. Call the SYSPROC.SYSTS_UPDATE stored procedure again. If the stored procedure returns successfully without warning, then all updated documents are included in the text search index.

Administering the event table

If a text search index is updated frequently, you can reduce the size of the event table by removing old event entries.

Procedure

Issue the DELETE command for the entries that you want to remove.

Example

For example, you can use the following command to remove events that occurred before a certain date:

```
DELETE FROM SYSIBMTS.EVENTS_[N] E WHERE DATE(E.TIME) < '2007-01-01'
```

where [N] is the index ID.

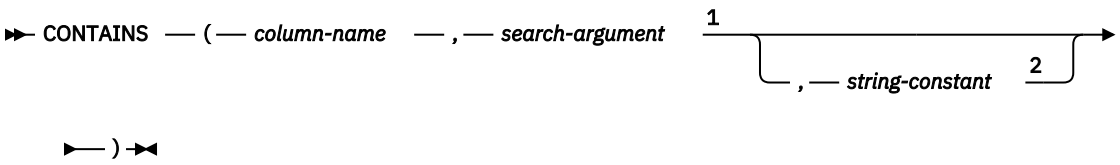
Chapter 6. User-defined functions and search argument syntax

The text search feature for Db2 for z/OS includes support for the CONTAINS function and the SCORE function in an SQL statement to search a text search index using the search argument criteria that you specify.

CONTAINS

The CONTAINS function searches a text search index using criteria that are specified in a search argument and returns a result about whether or not a match was found.

Requirement: To use the CONTAINS function, Text Search for Db2 for z/OS must be installed and configured. See for more information.

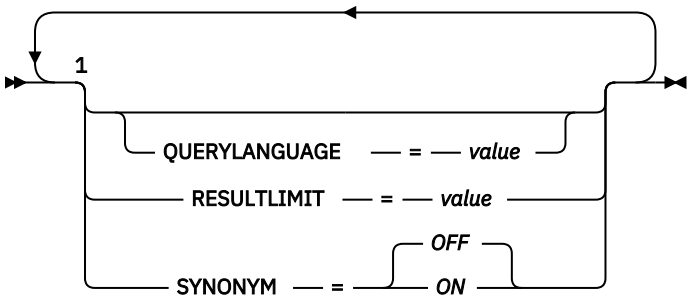


Notes:

[1] The SQL statement that invokes the CONTAINS function can be dynamically prepared by using a typed parameter marker for the *search-argument*, as in the following example: `CONTAINS(C1,CAST(? AS CHAR(10)))`.

[2] *string-constant* must conform to the rules for the *search-argument-options*.

search-argument-options



Notes:

[1] The same clause must not be specified more than once.

The schema is SYSIBM.

column-name

Specifies 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 that is 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. The underlying expression of the column of a view must be a simple column reference to the column of an underlying table, directly or through another nested view.

© Copyright IBM Corp. 2007, 2019

search-argument
Specifies an expression that returns a value that is a string value (except a LOB) that contains the terms to be searched for and must not be all blanks or the empty string. The actual length of the string must not exceed 4096 Unicode characters. 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.

string-constant
Identifies a string constant that specifies the search argument options that are in effect for the function.

The options that can be specified as part of the *search-argument-options* are as follows:

QUERYLANGUAGE = *value*
Specifies the query language. The value can be any of the supported language codes. If the QUERYLANGUAGE option is not specified, the default is the language value of the text search index that is used when this function is invoked. If the language value of the text search index is AUTO, the default value for QUERYLANGUAGE is en_US.

RESULTLIMIT = *value*
Specifies the maximum number of results to be returned from the underlying search engine. The *value* can be an integer value between 1 and 2 147 483 647. If the RESULTLIMIT option is not specified, no result limit is in effect for the query.

This scalar function cannot be called for each row of the result table, depending on the plan that the optimizer chooses. This function can be called once for the query to the underlying search engine, and a result set of all of the primary keys that match are returned from the search engine. This result set is then joined to the table containing the column to identify the result rows. In this case, the RESULTLIMIT value acts like a FETCH FIRST ?? ROWS from the underlying text search engine and can be used as an optimization. If the search engine is called for each row of the result because the optimizer determines that is the best plan, then the RESULTLIMIT option has no effect. Also, the RESULTLIMIT option has no effect when the CONTAINS function is used along with the comparison operators (<, >, <=, and >=) or the equality operator (=) and a value of 0 (zero).

SYNONYM = *OFF* or SYNONYM = *ON*
Specifies whether to use a synonym dictionary that is associated with the text search index. Use the Synonym Tool to add a synonym dictionary to the collection. The default is OFF.

OFF
Do not use a synonym dictionary.

ON
Use the synonym dictionary that is 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 the null value. If the third argument is null, the result is as if the third argument was not specified.

The result is 1 if the document contains a match for the search criteria that are specified in the search argument. Otherwise, the result is 0.

CONTAINS is a non-deterministic function.

Examples

Example 1: Assume that information about employees' skills are stored in a table named EMP_RESUME. The following statement finds all of the employees who have "COBOL" in their resume. The text search argument is not case-sensitive.

```
SELECT EMPNO
    FROM EMP_RESUME
    WHERE RESUME_FORMAT = 'ascii'
        AND CONTAINS(RESUME, 'cobol') = 1
```

Example 2: The search argument does not need to be a string constant. The search argument can be any SQL string expression, including a string contained in a host variable.

The following statement searches for the exact term "ate" in the COMMENT column:

```
char search_arg[100]; /* input host variable */
...
EXEC SQL DECLARE C3 CURSOR FOR
   SELECT CUSTKEY
   FROM K55ADMIN.CUSTOMERS
   WHERE CONTAINS(COMMENT, :search_arg)= 1
   ORDER BY CUSTKEY;
strcpy(search_arg, "ate");
EXEC SQL OPEN C3;
...
```

Example 3: The following statement finds 10 students at random who wrote online essays that contain the phrase "fossil fuel" in Spanish, which is "combustible fósil." These students will be invited for a radio interview.

Use the synonym dictionary that was created for the associated text search index. Because only 10 students are needed, you can optimize the query 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
```

Related concepts

Optimizing usage of the equality operator
Db2 optimizes queries that use the CONTAINS function along with the equality operator (=).

Related tasks

Retrieving messages without truncation
You can retrieve long message text from the CONTAINS or SCORE functions and from the text search administration stored procedures.

Searching a text search index
An application programmer can issue an SQL statement with a CONTAINS function or a SCORE function to search a text search index.

Related reference

SCORE
The SCORE function searches a text search index using criteria that are specified in a search argument and returns a relevance score that measures how well a document matches the query.

Search argument syntax
A search argument is the condition that is specified as part of a search for terms in text documents. It consists of search parameters and one or more search terms. The SQL scalar text search functions that use search arguments are CONTAINS and SCORE.

Optimizing usage of the equality operator

Db2 optimizes queries that use the CONTAINS function along with the equality operator (=).

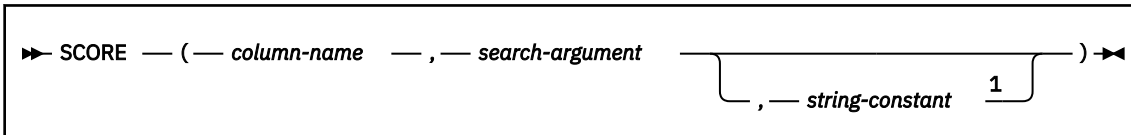
Use the equality operator rather than the comparison operators when you use the CONTAINS function, because Db2 does not optimize queries that use the CONTAINS function along with comparison operators (<, >, <=, and >=). The CONTAINS function can return either 0 or 1.

For example, instead of using the query `CONTAINS()>0`, replace the comparison operator (>) with the equality operator (=). The modified query is `CONTAINS()=1`.

SCORE

The SCORE function searches a text search index using criteria that are specified in a search argument and returns a relevance score that measures how well a document matches the query.

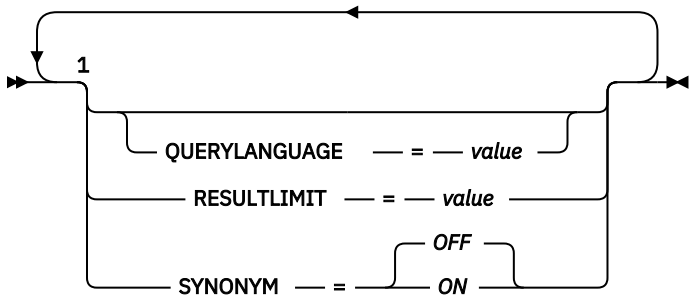
Requirement: To use the SCORE function, Text Search for Db2 for z/OS must be installed and configured. See for more information.



Notes:

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

search-argument-options



Notes:

[1] The same clause must not be specified more than once.

The schema is SYSIBM.

column-name
Specifies 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 that is 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. 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
Specifies an expression that returns a value that is a string value (except a LOB) that contains the terms to be searched for and must not be all blanks or the empty string. The actual length of the string must not exceed 4096 Unicode characters. 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.

string-constant
Identifies a string constant that specifies the search argument options that are in effect for the function.

The options that can be specified as part of the *search-argument-options* are as follows:

QUERYLANGUAGE = *value*
Specifies the query language. The value can be any of the supported language codes. If the QUERYLANGUAGE option is not specified, the default is the language value of the text search index that is used when this function is invoked. If the language value of the text search index is AUTO, the default value for QUERYLANGUAGE is en_US.

RESULTLIMIT = *value*
Specifies the maximum number of results that are to be returned from the underlying search engine. The *value* can be an integer value between 1 and 2 147 483 647. If the RESULTLIMIT option is not specified, no result limit is in effect for the query.

This scalar function cannot be called for each row of the result table, depending on the plan that the optimizer chooses. This function can be called once for the query to the underlying search engine, and a result set of all of the primary keys that match are returned from the search engine. This result set is then joined to the table containing the column to identify the result rows. In this case, the RESULTLIMIT value acts like a FETCH FIRST ?? ROWS from the underlying text search engine and can be used as an optimization. If the search engine is called for each row of the result because the optimizer determines that is the best plan, then the RESULTLIMIT option has no effect.

SYNONYM = *OFF* or SYNONYM = *ON*
Specifies whether to use a synonym dictionary that is associated with the text search index. Use the Synonym Tool to add a synonym dictionary to the collection. The default is OFF.

OFF
Do not use a synonym dictionary.

ON
Use the synonym dictionary that is 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 the null value. If the third argument is null, the result is as if the third argument was not specified.

The result is greater than 0 but less than 1 if the column contains a match for the search criteria that the search argument specifies. The better a document matches the query, the more relevant the score and the larger the result value. If the column does not contain a match, the result is 0.

Notes

Determinism:
SCORE is a non-deterministic function.

Examples

Assume that information about employees' skills are stored in a table named EMP_RESUME. The following statement generates a list of employees in the order of how well their resumes matches the query "programmer AND (java OR cobol)", along with a relevance value that is normalized between 0 (zero) 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
```

Db2 first evaluates the CONTAINS predicate in the WHERE clause, and therefore, does not evaluate the SCORE function in the SELECT list for every row of the table. In this case, the arguments for SCORE and CONTAINS must be identical.

Related concepts

Optimizing usage of comparison operators
Db2 optimizes queries that use the SCORE function along with comparison operators (<, >, <=, and >=).

Related tasks

Retrieving messages without truncation
You can retrieve long message text from the CONTAINS or SCORE functions and from the text search administration stored procedures.

Searching a text search index

An application programmer can issue an SQL statement with a CONTAINS function or a SCORE function to search a text search index.

Related reference

CONTAINS
The CONTAINS function searches a text search index using criteria that are specified in a search argument and returns a result about whether or not a match was found.

Search argument syntax
A search argument is the condition that is specified as part of a search for terms in text documents. It consists of search parameters and one or more search terms. The SQL scalar text search functions that use search arguments are CONTAINS and SCORE.

SYSPROC.SYSTS_CREATE
Invoke the SYSPROC.SYSTS_CREATE stored procedure to enable a text column for text search indexing, which allows the text search index to be used in SQL queries that contain the CONTAINS function or the SCORE function.

Optimizing usage of comparison operators

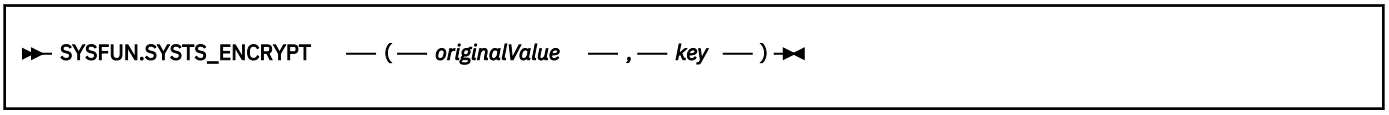
Db2 optimizes queries that use the SCORE function along with comparison operators (<, >, <=, and >=).

Use comparison operators rather than the equality operator (=) when you use the SCORE function, because the SCORE function returns a DOUBLE value between 0 (zero) and 1. This value is not well-suited for the equality operator.

SYSFUN.SYSTS_ENCRYPT

Use the SYSFUN.SYSTS_ENCRYPT user-defined function that is supplied with Db2 for z/OS to encrypt the passed value according to the text search server encryption method by using the input key.

You must use this user-defined function to encrypt a Db2 password that the text search server needs to connect to Db2.



The schema is SYSFUN.

Use this user-defined function to set the Db2 and the text search server passwords in the SYSIBMTS.SYSTEXTSERVERS table. Before invoking this user-defined function, store the key for the text search server encryption in the SERVERMASTERKEY column of the SYSIBMTS.SYSTEXTSERVERS table.

The SYSFUN.SYSTS_ENCRYPT function returns an encrypted version of the *originalValue* parameter value. The encryption is based on the *key* value that is provided as an additional input parameter. This function uses an encryption method that is shared with the text search server. Therefore, you can use this function to encrypt a password value for the DB2ENCRYPTEDPW column of the SYSIBMTS.SYSTEXTSERVERS administration table.

originalValue
Specifies the string that you want to encrypt.

The data type of this parameter is VARCHAR(32).

key
Specifies the key string that is used for encryption. A valid value must be retrieved from the text search server by using the Configuration Tool that is on the server.

The data type of this parameter is VARCHAR(36).

The function returns an encrypted version of the string as a variable-length string with a length attribute of 256.

Example 1

This example inserts a row with an encrypted password to the SYSIBMTS.SYSTEXTSERVERS table after a text search server is installed with the following features:

- Name: 192.30.176.75
- Listening to port 6007
- Requiring the authentication token 'fdRNw+4=' for all service requests
- Requiring the master key '55c1576b4de132dabdb28c10ce04ef36' for encrypting Db2 password information for the Db2 user ID that is in the SYSIBMTS.SYSTEXTCONNECTINFO table

Note: The literals that are used in this example are case-sensitive. Make sure that the environment that is used to run the SQL statements (for example, SPUFI) does not transform them to uppercase.

```
INSERT INTO SYSIBMTS.SYSTEXTSERVERS(SERVERNAME,
                                    SERVERPORT,
                                    SERVERAUTHTOKEN,
                                    SERVERMASTERKEY,
                                    DB2ENCRYPTEDPW)
            VALUES ('192.30.176.75',
                    6007,
                    'fdRNw+4=',
                    '55c1576b4de132dabdb28c10ce04ef36',
                    SYSFUN.SYSTS_ENCRYPT('myDb2Pw',
                                         '55c1576b4de132dabdb28c10ce04ef36'))
```

Example 2

The following example updates the password information in the SYSIBMTS.SYSTEXTSERVERS table after the password for the Db2 user ID in the SYSIBMTS.SYSTEXTCONNECTINFO table has been changed to 'newDB2pw':

```
UPDATE SYSIBMTS.SYSTEXTSERVERS SET
        DB2ENCRYPTEDPW = SYSFUN.SYSTS_ENCRYPT('newDB2pw', SERVERMASTERKEY)
```

Related reference

“SYSPROC.SYSTS_CREATE” on page 41
Invoke the SYSPROC.SYSTS_CREATE stored procedure to enable a text column for text search indexing, which allows the text search index to be used in SQL queries that contain the CONTAINS function or the SCORE function.

Search argument syntax

A search argument is the condition that is specified as part of a search for terms in text documents. It consists of search parameters and one or more search terms. The SQL scalar text search functions that use search arguments are CONTAINS and SCORE.

For any language-specific processing during a search, you can specify a value for the QUERYLANGUAGE parameter as a search argument option. The value can be any of the supported language codes. If the QUERYLANGUAGE parameter is not specified, the default value is the language value of the text search index that is used when this function is invoked. If the language value of the text search index is AUTO, the default value for QUERYLANGUAGE is en_US.

Simple search

To do a simple keyword search, you can enter one or more query terms (keywords). The search engine returns documents that contain all of those keywords, or variations of the keywords.

A keyword search is not case-sensitive. Therefore, a query for the term `king` will return matches for the terms `King` and `Kings`, and vice versa.

For example, if you enter `king`, the search engine returns all documents that contain the term `king`, or certain variations of the term such as `kings`. If you enter the query `king lear`, the search engine returns documents that contain the terms `king` and `lear`.

To see more precise results, use more specific keywords. For example, use `French roast coffee` rather than `coffee`, or use `Kauai hiking tours` rather than `Hawaiian vacations`.

If a simple keyword search returns too many documents that are not what you are looking for, you can use operators to refine your search.

Exclusion of terms in a search

Use the minus sign (-) to exclude terms. For example, if you want to find documents with the term `lear`, and you do not want to see documents with `king`, enter the query `lear -king`.

The minus sign (-) also applies to a term and its variants. For example, the query `-king` will exclude documents that contain the word `kings`.

Phrase search

If you want to ensure that terms appear in the same order as you typed them in and that term variants are not a match, you can use double quotation marks. For example, if you want to find documents with the term `king lear` exactly, and you do not want matches on related phrases such as `king and queen lear`, enter `"king lear"`. A phrase search is not case-sensitive.

Wildcard character in a search

The wildcard character (*) helps you find documents when you do not know the full spelling of a term. By using the wildcard character, you also find variations of the term. For example, the query `czech*` returns documents with the terms `czech`, `czechoslovakia`, `czechoslovakian`, and other possible results.

You can also use the wildcard character in a phrase search. For example, the query `"John * Kennedy"` returns documents with the terms `John Fitzgerald Kennedy` and `John F Kennedy` but not `John Kennedy`. The query `Mi*l Gorbachev` will return `Mikhail Gorbachev`.

You can use a wildcard character at the beginning of a query (for example, `*king`), but doing so might cause the search engine to take longer to return results.

Note: Wildcard card characters are ignored when processing Chinese, Japanese, and Korean documents.

Searches for at least one of the terms

The logical operator OR specifies that at least one of the terms in a query must appear in the returned document. For example, the query `(othello OR otello)` returns documents that contain the term `othello` or `otello`.

You can also use the logical operators AND, OR, and NOT in combinations by using parentheses. For example, the query `cougar OR (jaguar AND NOT car)` returns documents with the term `cougar`, or documents that contain the term `jaguar` but not the term `car`.

You must enter the logical operators AND, OR, and NOT in all uppercase. Use parentheses for grouping.

Fuzzy searches

A fuzzy search query searches for character sequences that are not only the same but similar to the query term. Use the tilde symbol (~) at the end of a term to do a fuzzy search. For example, the following query finds documents that include the terms analytics, analyze, analysis, and so on.

```
analytics~
```

You can add an optional parameter to specify the required similarity. Specify a value greater than 0 and less than 1. The value must be preceded by a 0 and decimal point (for example, 0.8). A value closer to 1 matches terms with a higher similarity. If the parameter is not specified, the default is 0.5.

```
analytics~0.8
```

Restriction: Special characters are not supported in fuzzy search queries. Avoid using double quotation marks (") in fuzzy search queries. If you enclose a search argument in quotation marks, the query is processed like an exact match search. For example, the search argument `CONTAINS (index,'"token"~0.8')` is processed as `CONTAINS (index,'token')`. Also, using the following types of non-alphabetic and nonnumeric characters in a fuzzy search term might cause unexpected results: @ + – & ‖ ! ( ) { } [ ] ^ " ~ * ? : \.

Proximity searches

A proximity search finds documents that contain terms within a specified number of words of each other. Use the tilde symbol (~) to do a proximity search. For example, the following query finds documents that contain "IBM" and "WebSphere®" within seven words of each other.

```
"IBM WebSphere"~7
```

Proximity search is supported for individual terms but not for phrases. Also, a word after a sentence break is not considered adjacent to words in the previous sentence.

Restriction: Special characters are not supported in proximity search queries.

Related concepts

“XML search for IBM Text Search for Db2 for z/OS” on page 85
You can index and search XML documents. The XML search grammar uses a subset of the W3 XPath language with extensions for text search. The extensions support range searches of numeric, date, and datetime values that are associated with an XML attribute or element. Structural elements can be used separately, or combined with free text in queries.

Related reference

“SCORE” on page 72
The SCORE function searches a text search index using criteria that are specified in a search argument and returns a relevance score that measures how well a document matches the query.

“CONTAINS” on page 69
The CONTAINS function searches a text search index using criteria that are specified in a search argument and returns a result about whether or not a match was found.

Special characters in searches

With IBM Text Search for Db2 for z/OS you can index and search special characters. When you include a special character in a search, the special character is handled like any other term in the query.

Important: IBMTS This information applies to IBM Text Search for Db2 for z/OS. IBMTS

When a special character is adjacent to a term in a query, documents that contain the special character and term in the same order are returned. For example, searching for “30$” finds documents that contain “30$”, but does not find documents that contain “$30”. However, searching for “30 $” (with a separating space) finds all documents that contain “30” and “$”, including the documents that contain either “30$” or “$30”.

When a special character separates two terms, the terms are searched for as a sequence. For example, searching for “jack_jones” finds documents that contain “jack_jones” but not documents that contain “jack_and_jones”.

Terms that are adjacent to special characters are lemmatized. For example, searching for “cats&dogs” in English finds documents that contain “cat&dog”. Also, when a term is adjacent to a special character in a query, the term is not removed from the query. For example, searching for “at&t” does not omit the term “at”. However, searching for “at & t” (separated by spaces) omits the term “at”.

You can use special characters in wildcard search expressions. For example, searching for “ja*_” finds documents that contain “jack_jones”. However, you cannot use wildcard characters to find special characters. For example, searching for “ca*s” finds documents that contain “cats”, “categories”, or “cas”, but not documents that contain “ca_s”.

Escaping special characters

Special characters can serve different functions in the query syntax. For example, you can use question marks (?) as wildcard characters. To search for a special character that has a special function in the query syntax, you must escape the special character by adding a backslash before it. For example:

- To search for the string “where?”, escape the question mark as follows: “where\?”
- To search for the string “c:\temp,” escape the colon and backslash as follows: “c\:\\temp”

If you do not escape such special characters, syntax errors can result.

Table 8. Special characters that must be escaped to be searched

Special character	Behavior when not escaped
Ampersand (&)	
Asterisk (*)	Used as a wildcard character.
At sign (@)	A syntax error is generated when an at sign is the first character of a query. In `xmlxp` expressions, the at sign is used to refer to an attribute.
Brackets [ ]	Used in `xmlxp` expressions to search the contents of elements and attributes.
Braces { }	Generates a syntax error.
Backslash (\)	
Caret (^)	Used for weighting (boosting) terms.
Colon (:)	Used to search in the contents of fields.
Equal sign (=)	Generates a syntax error.
Exclamation point (!)	A syntax error is returned when an exclamation point is the first character of a query.
Forward slash (/)	In `xmlxp` expressions, a forward slash is used as an element path separator.
Greater than symbol (>) Less than symbol (<)	Used in `xmlxp` expressions to compare the value of an attribute. Otherwise, these characters generate syntax errors.
Minus sign (-)	When a minus sign is the first character of a term, only documents that do not contain the term are returned.
Parentheses ( )	Used for grouping.
Percent sign (%)	Specifies that a search term is optional.
Plus sign (+)	
Question mark (?)	Handled as a wildcard character.
Semicolon (;)	
Single quotation mark (‘)	Single quotation marks are used to contain `xmlxp` expressions.

Table 8. Special characters that must be escaped to be searched (continued)

Special character	Behavior when not escaped
Tilda (~)	Handled as proximity and fuzzy search operators.
Vertical bar (|)	

Escaping special characters that do not serve a special function in the query syntax is optional. The following table shows some examples of special characters that do not require escaping.

Table 9. Examples of special characters that do not require escaping

Special character	Notes®
Comma (,)	
Dollar sign ($)	
Percentage (%)	
Period (.)	In `xmlxp` expressions, a period is used to search the content of elements.
Pound sign (#)	
Underscore (_)	

Simple query examples

Simple queries for the CONTAINS function and the SCORE function search for a single word or multiple words in a text search index.

The search engine ignores white space between characters. A search argument must not be an empty string or contain only blanks.

The following table shows some examples of simple search queries.

Table 10. Simple query examples

Search word types	Examples	Query results
Single word	`king`	Returns all documents that contain the word `king` or `kings`. This query matches linguistic variations of a word and is not case-sensitive.
Multiple words	`king lear`	Returns all documents that contain `king` and `lear`. The default operator is the logical operator AND.

The AND operator and the + (plus sign) operator are implicit in every query. For example, the query `King Lear` returns the same results as `King AND Lear` or `+King +Lear`.

Advanced search operators

You can use advanced search operators to refine the search results for the CONTAINS function and the SCORE function.

In the following table, the first column describes the operator that you can use in a search query. (You must enter the logical operators NOT, AND, and OR in all uppercase letters.) The second column shows a sample query that you might enter. The third column describes the types of results that you might see from the example query.

Table 11. Advanced search operators and complex query examples

Operators	Examples	Query results
AND	`"King Lear" AND "Othello"` `"King Lear" "Othello"`	Either query returns documents that contain both the phrase `King Lear` and the term `Othello`. The AND operator is the default conjunction operator. If no logical operator is between the two terms, the AND operator is used. In these examples, the first query handles synonyms differently than the second query. If the AND operator is used explicitly, synonyms are ignored. For example, if `Othello` is a synonym of `Richard` and a document contains `King Lear` and `Richard`, the second query will return this document. However, the first query will not return this document.
OR	`"King Lear" OR Queen`	Returns documents that contain either `King Lear` or just `Queen`. The OR operator links the two terms and finds a matching document if either of the terms exist in a document.
NOT	`"Lear"` `NOT "Norman Lear"`	Returns documents that contain `Lear` but not `Norman Lear`. Do not use the NOT operator with only one term. For example, the following search will return no results: `NOT "King Lear"`. A query should contain at least one term that is contained in the document.

" "
(Phrase search)
	First query:
	`"King Lear"`
	Second query:
	`"king"`
		The first query returns the exact phrase `King Lear`.
		The second query returns only the word `king` and no other forms, such as `kings`.

* (Wildcard character)	`mouse*` `mo*se`	Returns documents that can match possible variations of the query. For example, the query `mouse*` returns matches such as `mouse` and `moused`. The query `mo*se` returns matches such as `mouse`, `moose`, and `mongoose`.

Table 11. Advanced search operators and complex query examples (continued)

Operators	Examples	Query results

?
(Wildcard character)
	First query:
	`?Lear King`
	Second query:
	`mo?se`
		The first query returns all documents that contain the word `King` and that might contain the word `Lear`. The term that follows the ? is optional for the query.
		In the second query, the ? is used as a wildcard character for a single character. For example, the query returns all documents that contain the words `moose` and `mouse` but not the words `morose` or `mongoose`.

^ (Score boost factor) *some word or phrase^number*	`"King Lear"^4 "Richard III"`	A *boost factor* influences how documents are ranked in the search results. Documents that match query terms with high boost factors are ranked higher than if the boost factor was not applied. Although a boost factor must be positive, the boost factor can be less than 1. For example, 0.2. The boost factor number has no limit. This query forces a higher weight for documents with the phrase `"King Lear"` than the phrase `"Richard III"` in the list of search results.
+ (Includes)	`+Lear King`	Returns all documents that contain `Lear` and `King`, which is the same as the query `Lear AND King`.
- (Excludes)	`"Lear" -"Lear Jet"`	Returns documents that contain `Lear` but not `Lear Jet`.
( ) (Grouping)	`(King OR Lear) AND plays`	Returns documents that contain `plays` and either `King` or `Lear`. The parentheses ensure that `plays` is found and either term `King` or `Lear` is present.
\ (Escape character)	`http\://www.ibm.com`	Returns documents that contain `http://www.ibm.com`. Use the \ to clear special characters that are part of the query syntax. Special characters are + - & || ! ( ) { } [ ] ^ " ~ * ? : \. If a special character is cleared, the special character is analyzed as part of the query.

Operator precedence
A text search server supports the logical operators NOT, AND, and OR. These operators have normal precedence.

The NOT operator binds stronger than the AND operator, which binds stronger than the OR operator. For example, `a OR NOT b AND c` is equivalent to `(a OR ((NOT b) AND c)`. An operator binds only the terms that immediately precede or follow it.

In the absence of a logical operator, each term is processed as a required term. For example, `a b` is treated like `+a +b`.

All other operators bind stronger than logical operators, but logical operators bind stronger than no operators. For example, the query `a b OR c` is equivalent to `+a +(b OR c)`.

The modifiers +, -, and ? bind stronger than any other operators. The modifier + makes a term required; whereas, the modifier - excludes a term in a search. The modifier ? makes a term optional. Therefore, the query `a -b ?c` returns all documents that contain a, but do not contain b. If a document also contains c, that document will get a higher score, but c is not required.

Tip: Use parentheses to group logical expressions to ensure the precedence that you want. Also, you must enter the logical operators NOT, AND, and OR in all uppercase letters.

Using the OR operator in a WHERE clause
To optimize performance, try to use the OR operator to create queries that combine the second parameters together within a single CONTAINS or SCORE function in a WHERE clause, if the first parameter for the CONTAINS or SCORE functions is the same.

Consider the following query, which has two separate CONTAINS functions:

```
SELECT C2 FROM T1 WHERE CONTAINS(C2, 'fence') = 1 OR CONTAINS(C2, 'horse') = 1;
```

The first parameter in these two CONTAINS functions is the same, and the second parameter is different.

You can rewrite this query to optimize performance by using the OR operator to combine the two CONTAINS functions into one. The updated query is as follows:

```
SELECT C2 FROM T1 WHERE CONTAINS(C2, ' (fence) OR (horse) ') = 1;
```

In some cases the usage of the OR operator in a WHERE clause cannot be avoided. For example, if the query involves the CONTAINS and SCORE functions and several different text search indexes.

Consider that a table is created with two full-text indexes on the columns C2 and C3®, as follows:

```
CREATE TABLE T1(RID ROWID NOT NULL GENERATED BY DEFAULT, C2 VARCHAR(20),
C3 VARCHAR(20));
```

You can use two approaches to improve the performance of the following two queries that use scalar search functions:

```
Query 1:
SELECT C2, C3 FROM T1 WHERE CONTAINS(C2, 'fence') = 1 OR CONTAINS(C3, 'horse') = 1;
```

```
Query 2:
SELECT C2, C3 FROM T1 WHERE CONTAINS(C2, 'fence') = 1 OR C3='horse';
```

One way to improve the performance of these queries is to rewrite the queries as table functions, as follows:

```
Query 1:
WITH CONTAINS1 (RID) AS (SELECT RID FROM T1 WHERE CONTAINS (C2, 'fence') = 1),
     CONTAINS2 (RID) AS (SELECT RID FROM T1 WHERE CONTAINS (C3, 'horse') = 1)
SELECT DISTINCT C2, C3 FROM T1 T, CONTAINS1, CONTAINS2 WHERE
T.RID=CONTAINS1.RID OR T.RID=CONTAINS2.RID;
```

```
Query 2:
WITH CONTAINS1 (RID) AS (SELECT RID FROM T1 WHERE CONTAINS (C2, 'fence') = 1)
```

```
SELECT DISTINCT C2, C3 FROM T1 T, CONTAINS1 WHERE
T.RID=CONTAINS1.RID OR C3='horse';
```

The second approach that you can use to improve performance is to rewrite the queries by using UNION instead of an OR operator. For example:

```
Query 1:
SELECT C2, C3 FROM T1 WHERE CONTAINS(C2, 'fence') = 1
UNION
SELECT C2, C3 FROM T1 WHERE CONTAINS(C3, 'horse') = 1;
```

```
Query 2:
SELECT C2, C3 FROM T1 WHERE CONTAINS(C2, 'fence') = 1
UNION
SELECT C2, C3 FROM T1 WHERE C3='horse';
```

Wildcard character limitations

A search that contains a wildcard character (*) is limited to expanding to a maximum of 1024 matching search terms.

This limitation is due to the way that the text search server processes queries. The internal query processing takes a term such as "`termstart*`" and expands the term similar to looking up a word in a dictionary. For example, matching results are "`termstart1 or termstart2 or . . . termstart1024`" until the limit of 1024 terms is reached.

Due to this limitation, the internal query processing of a wildcard character search might return an error if the search reaches the maximum of 1024 matching search terms.

Restrictions for search argument syntax

Certain restrictions apply to search argument syntax when you are searching for terms in a text search index.

The following restrictions apply:

- If a search string is all uppercase or has irregular casing, the text search server searches for an exact match to the search string. Synonyms are not added to the search, and lemmatization, a process that identifies the root form and different grammatical forms of a word, is not performed.

Example 1: The following CONTAINS statement returns SQL code -171 to indicate that the value of an argument is invalid.

```
SELECT CUSTKEY  FROM ADMIN5.CUSTOMER FROM  WHERE CONTAINS
(COMMENT, '$') = 1;
```

Example 2: In the following CONTAINS statement, the dollar sign ('$') is ignored, resulting in a search for only the term 'total'.

```
SELECT CUSTKEY  FROM ADMIN5.CUSTOMER FROM  WHERE CONTAINS
(COMMENT, 'total$') = 1;
```

Example 3: The following CONTAINS statement searches for the exact term 'HOTEL' in the COMMENT column.

```
SELECT CUSTKEY  FROM ADMIN5.CUSTOMER FROM  WHERE CONTAINS
(COMMENT, 'HOTEL') = 1;
```

Example 4: In the following CONTAINS statement, the search process ignores the SYNONYM option, because the search term contains irregular casing. The search process only searches for the exact term 'hOtel' in the COMMENT column.

```
SELECT CUSTKEY  FROM ADMIN5.CUSTOMER FROM  WHERE CONTAINS
(COMMENT, 'hOtel', ' SYNONYM = ON ') = 1;
```

Example that uses the CONTAINS function and SCORE function

You can use the CONTAINS function and the SCORE function in the same query to search a text search index.

The result of a query that uses the CONTAINS function and the SCORE function returns whether a match was found and a relevance score that measures how well a document matches the query.

The example in the following table uses data from the base table BOOKS with the columns ISBN (VARCHAR(20)), ABSTRACT (VARCHAR(10000)), PRICE (INTEGER), and BOOKRID (ROWID NOT NULL).

Table 12. The base table BOOKS

ISBN	ABSTRACT	PRICE	BOOKRID
i1	"a b c"	7	ROWID NOT NULL
i2®	"a b d"	10	ROWID NOT NULL
i3	"a e a"	8	ROWID NOT NULL

You run the following query:

```
SELECT ISBN, SCORE(ABSTRACT,'"a"')
FROM BOOKS
WHERE CONTAINS (ABSTRACT,'"a"') = 1
```

This query returns the following results:

```
i1,    0.300000000000000E-02
i2,    0.300000000000000E-02
i3,    0.900000000000000E-02
```

The score values might differ depending on the content of the text column.

Using the RESULTLIMIT option in the CONTAINS and SCORE functions

The purpose of using the RESULTLIMIT option in the third parameter of the CONTAINS and SCORE function is to optimize performance.

About this task

Use of this option is effective only if the Db2 optimizer chooses a plan to retrieve a set of results from the text search server before joining those results with the results from other parts of the SQL search condition, which is called table mode for the text search query.

Example

The examples in this section use the RESULTLIMIT option.

Example 1: The following example uses the table mode and returns 5 results at the most:

```
select myColumn from myTable where contains(myColumn, ‘myQuery',
‘RESULTLIMIT=5') =1;
```

Example 2: The following example does not use the table mode. This query asks the text search server for every row in the table, if it matches the query (called "scalar mode" for the text search query). Therefore, the RESULTLIMIT option does not reduce the number of rows that are returned.

```
select contains(myColumn, ‘myQuery', ‘RESULTLIMIT=5') from myTable;
```

Example 3: For more complex queries it might not be obvious if the Db2 optimizer chooses the table or scalar search mode. Therefore, you should know the plan for the SQL statement with text search functions before using the RESULTLIMIT option to avoid unexpected results. You might use Db2 Performance Expert for this task.

Consider the following query:

```
select myColumn from myTable
where score(myColumn, ‘myQuery')>0.4 and
           score(myColumn, ‘myQuery', ‘RESULTLIMIT=1')<0.6 ;
```

If the Db2 optimizer chooses to evaluate the first score predicate in table mode, and the second in scalar mode, the specified RESULTLIMIT value would not be effective. When you exchange the two score predicates, the evaluation order might change, providing you with a different number of results.

Example 4: In the following example, the query returns both the row and score values for the first 10 results that contain the keyword “fence:”

```
SELECT C2, SCORE(C2, 'fence') FROM T1 WHERE CONTAINS(C2, 'fence',
'RESULTLIMIT=10')=1;
```

For this query, the Db2 optimizer creates a table function on `CONTAINS(C2, 'fence', 'RESULTLIMIT=10')` and then performs a scalar search on the corresponding rows again for `SCORE(C2, 'fence')`. Therefore, this query results in multiple calls to the server. In Db2, if the first, second, and third parameters are identical, the query is optimized to perform both the CONTAINS and SCORE functions at the same time. To improve performance, if the first and second parameters are the same for both the CONTAINS and SCORE functions, make sure that the third parameter is identical as well (character for character).

Consider the following query, which has been modified so that the first, second, and third parameter are the same:

```
SELECT C2, SCORE(C2, 'fence', 'RESULTLIMIT=10') FROM T1
WHERE CONTAINS(C2, 'fence', 'RESULTLIMIT=10')=1;
```

In this example, the Db2 optimizer recognizes that the CONTAINS and SCORE functions are the same function and only does a table function on `CONTAINS(C2, 'fence', 'RESULTLIMIT=10')`, while at the same time returning the SCORE results. Even though in this example the 'RESULTLIMIT=10' is not used in the SELECT clause for the SCORE function, including this option helps to improve performance.

XML search for IBM Text Search for Db2 for z/OS

You can index and search XML documents. The XML search grammar uses a subset of the W3 XPath language with extensions for text search. The extensions support range searches of numeric, date, and datetime values that are associated with an XML attribute or element. Structural elements can be used separately, or combined with free text in queries.

Documents must be indexed to include the XML markup before the index can be searched using the XPath query syntax. Document indexing is done by using the “FORMAT XML” option at index creation time.

Indexes created on a previous release can be used to perform searches. However, documents indexed on a previous release do not have the information necessary to use all the XML search capabilities available in a newer release. Documents added or updated in the text search index after the upgrade to the new release include the additional information.

An upgrade might result in documents indexed on the prior release not being included in some search results. You can use the SYSPROC.SYSTS_UPDATE stored procedure with the ALLROWS option to rebuild the index and resolve this problem.

To use the CONTAINS and SCORE built-in functions to search XML data, the query string must start with the `@xmlxp:` query prefix. The prefix is followed by a valid XML Search query expression. The `@xmlxp` 'opaque' term prefix indicates that a search is performed using the query path expression.

For example: `CONTAINS(columnname, '@xmlxp:''query_expression'' ')`.

The single quotes (') surrounding the `query_expression` must be doubled because they are contained within an SQL string, in effect, a string within a string.

The following list highlights the key features of XML search:

XML structural search

By including special opaque XML terms in queries, you can search XML documents for structural elements and text that is scoped by those elements. Structural elements are tag names, attribute names, and attribute values. Element and tag names are case sensitive.

XML query tokenization

Tokenization is the process of parsing input into tokens. Free text in XML query terms is tokenized the same way that text in non-XML query terms is tokenized. An exception is that nested opaque terms are not supported. Free text search is not case sensitive.

XML Schema and DTD

Any XML schema associated with the XML document is not downloaded, and default values are not indexed.

Numeric values

Predicates that compare attribute or element values to numbers are supported.

Element values

Predicates that compare element values to numbers or dates are supported. The element containing the date or number must be an XML element that contains only the number or date. Leading and trailing white space are ignored.

String values

Use of the = operator for a string argument in a predicate requires a complete match of all key words in the string with tokens in the identified text span. The order of the tokens is not significant when matching is performed.

Datetime values

Predicates that compare date or datetime attributes or elements are supported.

Path expressions

Path expressions are only allowed in the forward direction, and only on a single axis.

You should start path expressions with a leading / or //. These leading characters indicate that the initial context of the expression is the root node of the document. When the leading / or // is omitted, the expression is matched at any level. For example, `'Sentences'` is treated as `'//Sentences'`. The behavior is defined this way to be compatible with prior releases, and does not follow the W3 or SQL/XML standard.

The following tables show the supported path expressions and some examples.

Table 13. Path expressions

@xmlxp Expression	**Description**
`TagName`	Selects a tag named `TagName`, and all children of that tag.
`@AttributeName`	Selects an attribute named `@AttributeName`.
/	Selects from root node.
//	Selects matching tags and attributes that are descendants of the current position and match the expression.
.	Self: the current tag or element node.

Table 14. Path expression examples	
@xmlxp Expression	**Result**
`/Document`	Returns all documents with a top-level tag `Document`.
`//Document`	Returns all documents with a tag `Document` at any level.
`/Document/Child1`	Returns all documents with a top-level tag `Document` that has a direct child tag `Child1`.
`/Document//Child1`	Returns all documents with a top-level tag `Document` that has a descendant tag `Child1` at any level.
`/Root/@attr1`	Returns all document with a top-level tag `Root` with an attribute `attr1`.
`/Root//@attr1`	Returns all documents with a top-level tag `Root` with an attribute `attr1` on that root tag or any descendant tag.
`//@attr1`	Returns all documents that have an attribute `@attr1` at any level.

Note: The XML search expression must have an actual tag or attribute name in the relative path expression. The characters `/` and `//` by themselves are not valid search queries.

Wildcard character support

In the path expression, you can use the special wildcard character * to indicate exactly one tag, with any name.

Trailing path expression wildcard characters are ignored.

The following uses of wildcard characters are not supported:

- An expression that references only wildcard characters and no specific elements or attributes.
- A wildcard attribute at any level: `/Tag/@*`.
- A wildcard character that immediately precedes a predicate expression: `/Root/*[//anytag]`.
- A wildcard character that is used in a predicate comparison: `/Root[* > 5]`.
- A wildcard character that is used as an XML namespace prefix: `//*:tagname`.
- A wildcard character that is prefixed with an XML namespace prefix: `//ns:*`.
- A wildcard character that is used as part of a tag name: `/start*`.

The following table shows examples of wildcard characters in path expressions.

Table 15. Wildcard character in path expressions	
@xmlxp Expression	**Result**
`/Root/*/T1`	All documents having a top-level tag `Root` that has a descendant tag `T1` with one intermediate level.
`/Root/*//T1`	All documents having a top-level tag `Root` that has a descendant tag `T1` with one or more intermediate levels.

Predicates

Predicates are used to specify a value or condition that an element or attribute node must satisfy. Predicates are always enclosed in square brackets: `[]`.

Table 16. Predicate examples

@xmlxp Expression	**Result**
`/Book[Sentences]`	Top-level tag is Book and must have a direct child `Sentences`.
`/Book[.//Sentences and .//Author]`	Top-level tag is Book and must have both `Sentences` and `Author` descendants. Each descendant can be at any level below Book.

Because path expressions are always in the forward direction, and limited to a single access, path expressions in predicates must be relative to the current node. `/Book[/Root]` and `/Book[//Root]` are not valid, because in both cases the predicate path expression begins with the top-level tag 'Root' instead of the current node.

Numeric comparisons

IBM Text Search for Db2 for z/OS supports the =, <=, >=, >, <, and `!=` operators for comparisons of elements and attributes to integers and floating point values.

Elements have only their numeric values indexed if they are simple elements. Elements must not contain additional characters (other than white space) and must not have any descendant elements. Complex elements are indexed as text only.

Table 17. Numeric comparison examples

@xmlxp Expression	**Result**
`/Book[@id_num = 12345]`	Top-level tag is Book and must have an attribute `id_num` with a value of `12345`.
`/Book[Cost <= 100.50]`	Top-level tag is Book. Book has a direct child element `Cost` with a numeric value less than or equal to `100.50`.

Date and datetime comparisons

IBM Text Search for Db2 for z/OS supports the following operators for comparisons of elements and attributes to date and datetime values:

= <= >= > < !=

Simple elements have only their datetime values indexed. These elements must not contain additional characters (other than white space) and must not have any descendant elements. Complex elements are indexed as text only.

During indexing, attribute values and text contained within simple XML tags are examined. If the text is determined to match an ISO date or datetime format, it is indexed as a date or datetime that can be searched in a predicate.

During a search, the date or datetime value must be enclosed within an `xs:date()` or `xs:dateTime()` function call in order to be recognized as the correct data type.

An XML datetime data type in an XML document can specify a timezone value. However, when a datetime is indexed, the Text Search server truncates timezone values during indexing. Therefore, time zones are not considered during XML searches that involve date or datetime data types.

In addition, a datetime with an hour of 24 is permitted only if the minutes and seconds are zero. It will be treated as a value between the last instant of that day and the first instant of the next day.

When a value date or datetime is specified in an XML search predicate, a syntax error occurs if a time zone is specified on the value.

The datetime data type supports up to 12 digits of fractional seconds.

Table 18. Date and datetime comparison examples

@xmlxp Expression	**Result**
`/Book[@publishDate > xs:date(“2000-01-01”)]`	Top-level tag is Book. Book has an attribute `publishDate` that is greater than the date of 2000-01-01.
`/Book[purchaseTime > xs:dateTime(“2009-05-20T13:00:00”)]`	Top-level tag is Book. Book has a direct child `purchaseTime` that is a datetime expression greater than 2009-05-20T13:00:00.000000.

Contains and *excludes* in XML markup

The *contains* and *excludes* functions are used to perform full text searches within the XML markup. *Contains* returns true if the query is contained within the target node; *excludes* returns true if the query is NOT contained within the target node.

For example, find all documents with a top-level tag called `email`, and a direct descendant called body that contains variations of the phrase “Department budget”.

```
@xnkxo:''/email[body contains (“department budget”)]''
```

The free text passed to the *contains* or *excludes* function is handled in the same way as any other free text search. The search is not case-sensitive, and linguistic variations are considered. The earlier query matches “departments budgets” and also “budget for the department”.

The search can be restricted to an exact match by using the traditional quotation marks, for example, `@xmlxp:''/email[body contains(“””department budget”””)] ''`. The quotes indicating an exact match must be doubled so that they are not interpreted as the end of the *contains* free text string.

Table 19. Contains and excludes examples

@xmlxp Expression	**Result**
`/Book[abstract contains(“cat AND dog”)]`	Top-level tag Book that has a child tag `abstract` which contains linguistic variations of the terms `cat` and dog.
`/Book[abstract contains(“cat AND dog”)] /Book/@title[. contains(“cat OR dog”)]`	Top-level tag Book has an attribute `title` that contains linguistic variations of either `cat` or dog.
`/Book/Title[. contains(“””All good dogs go to heaven”””)]`	Top-level tag Book with a direct child `Title` that contains `all good dogs go to heaven` in order, and without linguistic variations being considered.
`/Book[abstract excludes(“cat AND dog”)]`	Top-level tag Book that has a child tag `abstract` which does not contain linguistic variations of the terms `cat` and dog.

Complete string match operator

The = operator with a string argument in a predicate calls for a complete match of all tokens in the string with all tokens in the identified text span. Linguistic equivalents are not considered. The order of the

terms searched for is not significant. It is not required that the element or attribute contain only the text that was searched for.

Table 20. Complete string match operator examples	
@xmlxp Expression	**Result**
`/Book[@author = “Nicholas Lawrence”]`	Top-level tag Book that has an attribute `author`. `author` must contain the terms `Nicholas Lawrence`. Linguistic variations on those terms are not considered matches.
`/Book[author = “””Nicholas Lawrence”””]`	Top-level tag Book that has a direct descendant `author`. `author` must contain the terms `Nicholas Lawrence` in order. Linguistic variations on those terms are not considered matches.

Logical operators

The logical operators AND and OR can be used in predicates.

Table 21. Logical operator examples	
@xmlxp Expression	**Result**
`/Book[@author = “””Nicholas Lawrence”””]/Price[. < 1000 and @unit = “dollars”]`	Top-level tag Book that has an attribute `author`. `author` must contain the terms `Nicholas Lawrence` in order. Linguistic variations on those terms are not considered matches. Book must have a direct child `Price` with a value < `1000`. The `Price` node must have an attribute `@unit` that has a value of `dollars`.

Operator precedence

In XML search predicates, containment operators and comparison operators take precedence over logical operators, and all logical operators have the same precedence.

- Containment operators are *contains* and *excludes*.
- Comparison operators are:

  = != < > <= >=
- Logical operators are AND and OR.

You can use parentheses to ensure the precedence that you want.

XML search query grammar

The grammar for XML search is based on a subset of the XPath language, which is defined by Extended Backus-Naur Form (EBNF) grammar. Queries that do not conform to the supported grammar are rejected by the query parser.

The EBNF grammar has been simplified in the following ways by:

- Disallowing absolute path names in predicate expressions
- Recognizing only one axis (tag) and only in the forward direction
- Applying additional semantic restrictions to the use of the wildcard character

- Requiring that the namespace declaration is specified in the search string before any usage, implied or explicit, of the namespace. If the namespace declaration is not included, namespaces are not considered in the search.
- Requiring that relative path expressions have a tag or attribute name included in the expression. The query '/' to select the root node and the query '//' to select all nodes are not valid expressions.

The following table shows the supported grammar in EBNF notation.

Table 22. Supported query grammar in EBNF notation

Symbol	Production
XMLQuery ::=	QueryPrefix NameSpaceDeclaration QueryString | QueryPrefix QueryString
QueryPrefix ::=	@xmlxp:
QueryString ::=	"'" PathExpr "'"
PathExpr ::=	RelativePathExpr | "/" RelativePathExpr? | "//" RelativePathExpr
RelativePathExpr ::=	StepExpr (("/" | "//") StepExpr)*
StepExpr ::=	("." | AbbrevForwardStep) Predicate?
AbbrevForwardStep ::=	"@"? (QName | "*")
Predicate ::=	"[" PredicateExpr "]"
PredicateExpr ::=	Expr | PredicateExpr ("and" | "or") | "(" PredicateExpr ")"
Expr ::=	ComparisonExpr | ContainmentExpr
ComparisonExpr ::=	PathExpr ComparisonOp Literal
ComparisonOp ::=	"=" | "<" | ">" | "!=" | "<=" | ">="
Literal ::=	StringLiteral | NumericLiteral | DateLiteral
ContainmentExpr ::=	PathExpr "contains" "(" StringLiteral ")" | PathExpr "excludes" "(" StringLiteral ")"
StringLiteral ::=	"\"" [^"]* "\"" | "'" [^']* "'"
DateLiteral ::=	"xs:date(\"" xmlDate "\")" | "xs:dateTime(\"" xmlDateTime "\")"
xmlDate ::=	yyyy"-"mm"-"dd
xmlDateTime ::=	yyyy"-"mm"-"dd [T] hh":"mm":"ss"."uuuuuu
NameSpaceDeclaration ::=	defaultNameSpace (NameSpacePrefixDeclaration)*
defaultNameSpace ::=	“declare default element namespace “ StringLiteral “;”

Table 22. Supported query grammar in EBNF notation (continued)	
NameSpacePrefixDeclaration ::=	“declare namespace” NameSpacePrefix “=” StringLiteral “;”
NameSpacePrefix ::=	[^”:]+

Namespaces

You can use a namespace to scope elements and attributes in a document. Namespaces are useful in restricting the query search to the meaningful elements within the document.

In XML, element and attribute names are chosen by the developer. These names can create conflicts when XML documents from different applications are mixed.

Therefore, restricting the query search to the meaningful elements within the document is useful, especially when multiple document types might be indexed. Restricting the search can be accomplished by using namespaces.

Namespaces provide scoping of the elements and attributes of the document to ensure correct interpretation of the values. Namespaces are described with a long name (URI), and optionally, a short name that is called the qualified name (QName).

Consider the following example:

```
<?xml version='1.0'?>
  <doc xmlns:x="http://example.com/ns/abc">
    <x:p/>
  </doc>
```

`http://example.com/ns/abc` is the long name and x is the QName prefix. A QName prefix is useful as a shorthand for the namespace of each element reference. Element p is qualified by namespace `http://example.com/ns/abc`.

Default namespaces

A default namespace can be specified for XML elements. The default namespace applies to the current tag and any descendent tags. Any unqualified tag in the namespace inherits the default namespace.

In the following example, both doc and p elements are in the `http://example.com/ns/abc` namespace.

```
<?xml version='1.0'?>
  <doc xmlns="http://example.com/ns/abc">
    <p/>
  </doc>
```

Attribute namespaces

An attribute might have a different namespace than its associated element.

The following code example shows the qualified element and attribute:

```
<dog xmlns:an="http://example.org/animals" xmlns:sz=”http://example.org/sizes”>
<an:breed sz:size=”Medium”>Mutt</an:breed>
</dog>
```

There is a difference in how elements and attributes inherit a namespace when it is not explicitly specified. Unqualified elements pick up the default namespace of the scope within which they lie. Unqualified attributes do not have any namespace.

This next example shows the element and attribute as non-qualified:

```
<dog xmlns:an="http://example.org/animals">
<breed size=”Medium”>Mutt</an:breed>
</dog>
```

In this example, element `breed` has a namespace of `http://example.org/animals`. However, attribute `size` has no namespace associated with it.

For more information about XML namespaces, see W3C Recommendation for Namespaces in XML.

Reserved QName prefixes

The following QName prefixes are reserved and must not be used to qualify user-defined elements or attributes: **xml**, **xs**, **xsi**, **fn**, **local**.

Namespace declarations in a search

You must define QName prefixes and default element namespaces in the XPath query prolog of the search term.

If a query does not declare any QName prefix or default element namespace, namespaces are not considered in the query. If an element or attribute name exists in any namespace, the element or attribute name is considered a match. If any QName prefix or default namespace is declared, element or attribute names are a match only if they exist in the namespace that is specified. QName prefixes used in the XML search string are not required to match the QName prefix that is used in the XML document. Matches are based solely on the URI.

The following prolog maps namespace *ns1* to long name (URI) `"http://mycompany.com"`:

```
declare namespace ns1 = "http://mycompany.com";
```

In this next example, the prolog specifies that all unqualified elements are qualified by URI `"http://mycompany.com"`:

```
declare default element namespace "http://mycompany.com"
```

The following syntax could be used to indicate that unqualified tags are not in any namespace:

```
declare default element namespace "";
```

Consider these additional examples when defining namespaces.

Example 1: The following example restricts search to attribute `attr` of element `test`, where element `test` is mapped to `namespace "http://posample.org"`, and `attr` is not in any `namespace`. Use default `namespace` to simplify the syntax.

```
CONTAINS(myxmlcol, '@xmlxp:''declare default element namespace
“http://myexample.org”;
 /test[@attr > xs:date(“2005-01-01”)]''')
```

Example 2: This next example restricts search to attribute `attr` of element `test`, where element `test` has a `namespace` of `"http://myexample.org"`. Use explicit `namespace` syntax by using the `QName` prefix `abc`.

```
CONTAINS(myxmlcol, '@xmlxp:''declare namespace abc =  “http://myexample.org”;
 /abc:test[@attr < xs:date(“2009-01-01”)]''')
```

Example 3: This example restricts search to `shipTo` `name` and `billTo` `name` child elements of element `purchaseOrder`, which is explicitly mapped to `namespace "http://myexample.org"` using `QName` prefix `ns1`. A default namespace is also defined (`"http://mastsample.org"`), which applies to `shipTo`, `name`, and `billTo`.

```
CONTAINS(myxmlcol, '@xmlxp:'' declare default namespace "http://mastsample.org";
 declare namespace ns1 = "http://posample.org";
 /ns1:purchaseOrder[shipTo/name = "Jane"  and billTo/name = "Jason"]''')
```

Example 4: This example restricts search to attribute name (explicitly defined in namespace "http://posample.org") of shipTo element (in default namespace "http://mastsample.org"), which is a child of element purchaseOrder (explicitly defined in namespace "http://posample.org"). The default namespace "http://mastsample.org" applies to elements shipTo, billTo and name.

```
CONTAINS(myxmlcol, '@xmlxp:'' declare default namespace "http://mastsample.org";
 declare namespace ns1 = "http://posample.org";
 /ns1:purchaseOrder/shipTo[@ns1:name = "Jane" and billTo/name = "Jason"]''')
```

XML search scenario and query examples

This step-by-step scenario shows you how to create a text search index, insert XML documents into a table, and update a text search index. In addition, this information provides six examples of text search queries that use the CONTAINS function.

To create and update a text search index:

1. Issue the CREATE TABLE statement to create a table XML_DOCUMENTS in schema XMLTEST to store the XML documents:

   ```
   CREATE TABLE XMLTEST.XML_DOCUMENTS (ID INT, XML_DATA XML, PRIMARY KEY (ID));
   ```

2. Run a program that calls the SYSPROC.SYSTS_CREATE stored procedure to create a text search index called XML_INDEX over the XML column:

   ```
   call SYSPROC.SYSTS_CREATE('XMLTEST', 'XML_INDEX',
   'XMLTEST.XML_DOCUMENTS(XML_DATA)', '');
   ```

3. Next, insert some XML documents into the table by issuing the INSERT statement:

   ```
   INSERT INTO XMLTEST.XML_DOCUMENTS (ID, XML_DATA)
   VALUES(1,
         '<BOOK publication_date="2009-01-01">' ||
         ' <TITLE> Text Search Server for DB2 </TITLE>' ||
         ' <ID_NUMBER> 1 ></ID_NUMBER>' ||
         ' <CHAPTER>' ||
         '   <NUMBER> 1 </NUMBER>' ||
         '   <TITLE> Introduction </TITLE>' ||
         '   <ABSTRACT> This chapter will introduce the reader to the capabilities
                        of Text Search for DB2 for z/OS </ABSTRACT>' ||
         ' </CHAPTER>'||
         ' <CHAPTER>' ||
         '   <NUMBER> 2 </NUMBER>' ||
         '   <TITLE> Creating a Text Search Index </TITLE>' ||
         '   <ABSTRACT> This chapter will explain how to create
                        a text search index </ABSTRACT>' ||
         ' </CHAPTER>' ||
         '</BOOK>');
   ```

   ```
   INSERT INTO XMLTEST.XML_DOCUMENTS (ID, XML_DATA)
   VALUES(2,
         '<BOOK publication_date="2010-02-01">' ||
         ' <TITLE>  Using the XML data type for DB2 for z/OS </TITLE>' ||
         ' <ID_NUMBER> 2 ></ID_NUMBER>' ||
         ' <CHAPTER>' ||
         '   <NUMBER> 1 </NUMBER>' ||
         '   <TITLE> Introduction </TITLE>' ||
         '   <ABSTRACT> This chapter will introduce the reader to the
                        DB2 XML data type </ABSTRACT>' ||
         '  </CHAPTER> ' ||
         '  <CHAPTER>' ||
         '    <NUMBER> 2 </NUMBER>' ||
         '    <TITLE>  Inserting XML data into a DB2 table </TITLE>' ||
         '    <ABSTRACT> This chapter will explain how to insert XML
                         data into a DB2 table </ABSTRACT>' ||
         '  </CHAPTER>' ||
         '  <CHAPTER>' ||
   ```

```
      '    <NUMBER> 3 </NUMBER>' ||
      '    <TITLE>  Searching XML data </TITLE>' ||
      '    <ABSTRACT> This chapter will explain how to query data in XML columns
                      using the CONTAINS and SCORE UDFS </ABSTRACT>' ||
      '  </CHAPTER>' ||
      '</BOOK>');
```

```
INSERT INTO XMLTEST.XML_DOCUMENTS (ID, XML_DATA)
VALUES(3,
      '<BOOK xmlns="http://www.ibm.com/digital_media_library"' ||
      '      publication_date="2010-02-01">' ||
      ' <TITLE> Using Namespaces with Text Search Server for
                DB2 for z/OS </TITLE>' ||
      ' <ID_NUMBER> 2 </ID_NUMBER>' ||
      ' <CHAPTER>' ||
      '   <NUMBER> 1 </NUMBER>' ||
      '   <TITLE> Introduction </TITLE>' ||
      '   <ABSTRACT> This chapter will introduce the reader to XML
                     namespaces </ABSTRACT>' ||
      ' </CHAPTER>' ||
      ' <CHAPTER>' ||
      '   <NUMBER> 2 </NUMBER>' ||
      '   <TITLE> Using default namespaces </TITLE>' ||
      '   <ABSTRACT> This chapter will explain how to use a namespace
                     in an XML search </ABSTRACT>' ||
      ' </CHAPTER>' ||
      '</BOOK>');
```

4. Finally, update the copy of the text search index that is stored on the text search server by calling the SYSPROC.SYSTS_UPDATE stored procedure:

```
CALL SYSPROC.SYSTS_UPDATE('XMLTEST', 'XML_INDEX', '');
```

Example queries

Based on the previous scenario, consider the following examples of how to query the XML documents.

Example 1:

This example finds all documents that have a root element BOOK with a direct descendant TITLE that contains DB2.

```
SELECT ID
FROM XMLTEST.XML_DOCUMENTS
WHERE CONTAINS(XML_DATA, '@xmlxp:''/BOOK/TITLE[. contains("DB2")]'' ') = 1;
```

Because a namespace prolog is not specified in the search term, no namespace is considered in the search.

Table 23. Result

ID
1
2
3

Example 2:

This example finds all documents that have a root element BOOK with a direct descendant TITLE that contains DB2. Use a default element namespace to indicate that BOOK and TITLE must be in the "http://www.ibm.com/digital_media_library" namespace.

```
SELECT ID
FROM XMLTEST.XML_DOCUMENTS
WHERE CONTAINS(XML_DATA, '@xmlxp:''declare default element namespace
 "http://www.ibm.com/digital_media_library";
 /BOOK/TITLE[. contains("DB2")]'' ') = 1;
```

Table 24. Result
ID
3

Example 3:

This example finds all documents that have a root element BOOK that has an attribute `publication_date` after `"2010-01-01"` and has a child element `TITLE` that contains DB2. Restrict the search so that tags BOOK and `TITLE` must not exist in any namespace.

```
SELECT ID
FROM XMLTEST.XML_DOCUMENTS
WHERE CONTAINS(XML_DATA, '@xmlxp:''declare default element namespace "";
/BOOK[@publication_date > xs:date("2010-01-01")]/TITLE[. contains("DB2")]'' ') = 1;
```

Table 25. Result
ID
2

Example 4:

This example finds all documents with a root element BOOK (not in any namespace) that have a direct descendant CHAPTER (also not in a namespace) that contains information about inserting data into an XML table.

```
SELECT ID
FROM XMLTEST.XML_DOCUMENTS
WHERE CONTAINS(XML_DATA, '@xmlxp:''declare default element namespace "";
/BOOK/CHAPTER[. contains("inserting XML data into a table")]'' ') = 1;
```

Note: The text contained within CHAPTER includes the text contained within the ABSTRACT and `TITLE` elements that are the descendants of CHAPTER. Also, the search string is not case-sensitive, and linguistic variations of the search words are considered.

Table 26. Result
ID
2

Example 5:

This example finds all documents with a root element BOOK (in namespace “`http://www.ibm.com/digital_media_library`”) that have a direct descendant CHAPTER (also in namespace “`http://www.ibm.com/digital_media_library`”). CHAPTER must have a direct descendant NUMBER (in namespace “`http://www.ibm.com/digital_media_library`") with a value of 1, and also contain text information about searching an XML namespace.

```
SELECT ID
FROM XMLTEST.XML_DOCUMENTS
WHERE CONTAINS(XML_DATA, '@xmlxp:''declare namespace ns1 =
"http://www.ibm.com/digital_media_library";
/ns1:BOOK/ns1:CHAPTER[. contains("search XML using a namespace")
and NUMBER = 1]'' ') = 1;
```

Document 3 is the only document with tags in the correct namespace, but it has key word matches only in a chapter with a number value of 2 (not 1).

No rows are returned.

Table 27. Result
ID

Example 6:

This example finds all documents with a root element BOOK (in namespace "`http://www.ibm.com/digital_media_library`") that have a direct descendant CHAPTER (in namespace "`http://www.ibm.com/digital_media_library`"). CHAPTER must have a direct descendant NUMBER (in namespace "`http://www.ibm.com/digital_media_library`") with a value of 1. BOOK must have a descendant CHAPTER (not necessarily with a NUMBER descendant) that contains text information about searching an XML namespace.

```
SELECT ID
FROM XMLTEST.XML_DOCUMENTS
WHERE CONTAINS(XML_DATA, '@xmlxp:''declare namespace ns1 =
"http://www.ibm.com/digital_media_library";
/ns1:BOOK[ns1:CHAPTER contains("search XML using a namespace")]
/ns1:CHAPTER[ns1:NUMBER = 1]'' ') = 1;
```

Document 3 does have a CHAPTER element that matches the CONTAINS criteria, and also has a CHAPTER element with a descendant NUMBER that has a value of 1. Therefore, document 3 is a match for this query.

Table 28. Result
ID
3

Related reference

CREATE TABLE (DB2 SQL)

INSERT (DB2 SQL)

"SYSPROC.SYSTS_UPDATE " on page 59

Invoke the SYSPROC.SYSTS_UPDATE stored procedure to update the text search index to reflect the current contents of the text column that the text search index is associated with.

Chapter 7. Administering a text search server

You can use the scripts and command-line tools that come with a text search server to help you successfully administer the server and diagnose problems.

Starting a text search server

You can start a text search server manually by using the startup script that is provided.

About this task

If you installed the text search server as a service, the text search server starts automatically each time that the host system starts. However, you can start the text search server manually even if you installed it as a service. You can start only one instance of a text search server for each Linux, zLinux, or Windows server.

Procedure

Run the script that is provided for starting the server.

Option	Description
On a Linux or zLinux server, run the following script:	`<INSTALL_HOME>/bin/startup.sh`
On a Windows server, run the following script:	`<INSTALL_HOME>/bin/startup.bat`

Stopping a text search server

You can stop a text search server manually by using the shutdown script that is provided.

About this task

If you installed the text search server as a service, the text search server stops automatically each time that the host system is shut down. However, you can stop the server manually even if you installed it as a service.

Procedure

Run the script that is provided for stopping the server.

Option	Description
On a Linux or zLinux server, run the following script:	`<INSTALL_HOME>/bin/shutdown.sh`
On a Windows server, run the following script:	`<INSTALL_HOME>/bin/shutdown.bat`

Adding a synonym dictionary to a collection

Use the Synonym Tool to add a synonym dictionary to a specific collection. Specifying synonym groups in a synonym dictionary improves the quality of text search results.

Procedure

To add a synonym dictionary to a collection:

1. Create a synonym XML file by specifying the synonym groups, as shown in the following example:

© Copyright IBM Corp. 2007, 2019

```
<?xml version="1.0" encoding="UTF-8"?>
<synonymgroups version="1.0">
<synonymgroup>
  <synonym>Paixão</synonym>
  <synonym>amor</synonym>
  <synonym>flor</synonym>
  <synonym>linda</synonym>
</synonymgroup>
<synonymgroup>
  <synonym>worldwide patent tracking system</synonym>
  <synonym>wpts</synonym>
</synonymgroup>
</synonymgroups>
```

2. Copy the synonym XML file to any directory on the text search server.
3. Use the Synonym Tool to add the synonym dictionary to a collection.

   You can add a synonym dictionary in append mode or replace mode. If you add a synonym dictionary in append mode, the new synonyms will be added to the existing synonym dictionary. If you add a synonym dictionary in replace mode, the existing synonyms will be replaced by the new synonyms that you defined for the text search index.

Option	Description
On a Linux or zLinux server, enter the following command:	`synonymTool.sh importSynonym -synonymFile <absolute path to synonym XML file> -collectionName <collection name> -replace <[true|false]> -configPath <absolute path to configuration folder>`
On a Windows server, enter the following command:	`synonymTool.bat importSynonym -synonymFile <absolute path to synonym XML file> -collectionName <collection name> -replace <[true|false]> -configPath <absolute path to configuration folder>`

   Note: If the value for *-configPath* contains blanks, you must enclose the value in quotation marks.

   If the format of the XML file is invalid, or the XML file is empty, an error code is returned.

Related reference

“Synonym Tool” on page 105
Use the Synonym Tool to add a synonym dictionary to a collection to improve the quality of search results, or to remove synonyms from a collection.

Removing a synonym dictionary from a collection

Use the script that is provided to remove a synonym dictionary from a collection. The text search server administrator needs to retrieve the name of the collection from which you want the synonym dictionary removed.

Procedure

Run the script to remove the synonym dictionary.

Option	Description
On a Linux or zLinux server, enter the following command:	`synonymTool.sh removeSynonym -collectionName <collection name> -configPath <absolute path to configuration folder>`

Option	Description
On a Windows server, enter the following command:	`synonymTool.bat removeSynonym -collectionName <collection name> -configPath <absolute path to configuration folder>`

Note: If the value for *-configPath* contains blanks, you must enclose the value in quotation marks.

What to do next
If a database has several text search indexes, you must complete this task for each of the corresponding collections.
Related reference
"Synonym Tool" on page 105
Use the Synonym Tool to add a synonym dictionary to a collection to improve the quality of search results, or to remove synonyms from a collection.

Uninstalling a text search server

When you uninstall a text search server, the text search index and the configuration data remain intact.

Before you begin
You must stop a text search server before you can uninstall it.

Procedure

Complete the following steps for the type of server you use.

Option	Description

On a Linux or zLinux server:	a. Enter the following command:
	`cd <INSTALL_HOME>/Uninstall_ECMTextSearch/`
	b. Next, enter the following command:
	`./Uninstall_ECMTextSearch -i silent`

On a Windows server:	a. Open a command prompt window. (Select **Start -> Run**, enter ' cmd ', and then click **OK**.)
	b. Enter the following command:
	`cd <INSTALL_HOME>\Uninstall_ECMTextSearch\`
	c. Next, enter the following command:
	`Uninstall_ECMTextSearch.exe -i silent`

Example

For example, for a Linux server enter the following commands:

```
cd /opt/ibm/ts/Uninstall_ECMTextSearch/
```

```
./Uninstall_ECMTextSearch -i silent
```

Command-line tools

Use the provided command-line tools to complete common tasks, such as configuring and administering the text search server and adding a synonym dictionary to a collection.

These command-line tools do not authenticate user IDs; however, only a user ID with valid access to the text search server can run these command-line tools.

Configuration Tool

Use the Configuration Tool to customize configuration settings after you install a text search server.

To customize most of the configuration settings, you must stop the text search server before running the Configuration Tool. However, when the server is running, you can run the options to display the current authentication token, the server port, and the current properties of the system.

Command

The command that you issue to run the Configuration Tool depends on whether the text search server is installed on a Linux, zLinux, or Windows server.

Table 29. Commands to run the Configuration Tool

On a Linux or zLinux server	On a Windows server
`configTool.sh configuration_command -configPath value [-locale value] -command_specific_arguments`	`configTool.bat configuration_command -configPath value [-locale value] -command_specific_arguments`

Global arguments

-configPath
Specifies the fully qualified path to the configuration directory, such as `/opt/ibm/search/config`. If this value contains blanks, you must enclose the value in quotation marks. This global argument is mandatory.

-locale
Specifies the five-character locale code, such as en_US for English or de_DE for German. The default value is en_US.

Command options

The Configuration Tool supports the following command options.

configureHTTPListener
Initializes the text search server and configures the HTTP port for the text search application, the administration HTTP port, and the host name. After installing a text search server, you need to run this command to initialize the text search server.

You can configure the following parameters:

-adminHTTPPort
Specifies the administration HTTP port number.

-searchHTTPPort
Specifies the HTTP port number for the text search application. This parameter is optional.

Important: You must specify the same single value for both the administration HTTP port number and the HTTP port number for the text search application. Do not modify only one of these values later by using the Configuration Tool options *-adminHTTPPort* or *-searchHTTPPort*.

configureParams
Specifies the system parameters that you can configure. You can configure the following parameters:

-adminHTTPPort
Specifies the administration HTTP port number. This parameter is optional.

-searchHTTPPort
Specifies the HTTP port number for the text search application. This parameter is optional.

-configPath
Specifies the fully qualified path to the configuration directory, such as `/opt/ibm/search/config`. If this value contains blanks, you must enclose the value in quotation marks. This parameter is mandatory.

-logPath
Specifies the fully qualified path to the log directory. This parameter is optional.

-tempDirPath
Specifies the fully qualified path to the temporary directory. This parameter is optional.

-defaultDataPath
Specifies the fully qualified path to the default data directory for the text search index. This parameter is optional.

-installPath
Specifies the fully qualified path to the installation directory for the text search server. This parameter is optional.

-logLevel
Specifies the log level for system messages in the log file. The default level is `INFO` for informational. Additional options are `FINE`, `FINER`, `FINEST`, `OFF`, `WARNING`, and `SEVERE`. This parameter is optional.

printToken
Prints the current authentication token and encryption key.

printAdminHTTPPort
Prints the current value for the administration HTTP port.

printAll
Prints all of the current values for the options that you can configure with this tool.

generateToken
Generates the authentication token.

-seed
Specifies a random string that is used when generating the authentication token.

help
Prints the usage information for this tool.

Examples

Example 1: On a Linux server, use the following command to initialize the text search server and configure the available parameters:

```
configTool.sh configureHTTPListener
-configPath <absolute path to config folder>
[-locale <five-character locale string>]
–adminHTTPPort <admin HTTP port number>
[–searchHTTPPort] <search HTTP port number>
```

Example 2: On a Windows server, use the following command to generate an authentication token:

```
configTool.bat generateToken –configPath
[-locale <5-character locale string>]
–seed <any random string>
```

Example 3: On a Linux server, use the following command to print the current authentication token:

```
configTool.sh printToken -configPath /opt/ibm/search/config
```

Administration Tool

Run the script that starts the Administration Tool to do common administration tasks.

You can use the Administration Tool to do the following tasks:

- Check the status of collections, such as finding out how many documents are present
- Delete orphan collections
- Report the version of the server
- Report all of the collections that reside on the text search server
- Configure trace settings
- Print trace settings

For most tasks, the text search server must be running when you use the Administration Tool. However, when the server is not running, you can still run the commands for reporting the version of the server and printing the help message.

Commands

The command that you issue to run the Administration Tool depends on the task that you want to do and whether the text search server is installed on a Linux, zLinux, or Windows server.

Table 30. Commands to run the Administration Tool

On a Linux or zLinux server	On a Windows server
`adminTool.sh admin_command -configPath [-locale] [-collectionName] [-trace]`	`adminTool.bat admin_command -configPath [-locale] [-collectionName] [-trace]`

admin_command options

delete
Specifies that you want to delete the orphaned collection.

status
Checks the status of all of the collections.

version
Prints the version of the server.

configureTrace
Configures the trace settings for the text search server. You can turn the trace on or off.

printTrace
Prints the trace settings for the text search server.

help
Prints the usage information for this tool.

Parameters

-configPath
Specifies the fully qualified path to the configuration directory, such as `/opt/ibm/search/config`. If this value contains blanks, you must enclose the value in quotation marks. This parameter is mandatory.

-locale
Specifies the five-character locale code, such as en_US for English or de_DE for German. This parameter is optional. If you do not specify a value, the value for the server locale is used.

-collectionName
Specifies the name of the collection that you want to delete or obtain information about.

-trace
Specifies the trace value. Supported values are on or off. The default value is off.

Example

On a Linux server, use the following command to delete the collection that is named "Default":

```
adminTool.sh delete -configPath /opt/ibm/search/config -collectionName Default
```

Synonym Tool

Use the Synonym Tool to add a synonym dictionary to a collection to improve the quality of search results, or to remove synonyms from a collection.

You can use the Synonym Tool whenever the text search server is running. The text search server administrator has the correct authority and privileges to run the Synonym Tool.

A synonym dictionary consists of synonym groups that you define in an XML file. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<synonymgroups version="1.0">
<synonymgroup>
  <synonym>Paixão</synonym>
  <synonym>amor</synonym>
  <synonym>flor</synonym>
  <synonym>linda</synonym>
</synonymgroup>
<synonymgroup>
  <synonym>worldwide patent tracking system</synonym>
  <synonym>wpts</synonym>
</synonymgroup>
</synonymgroups>
```

Commands

The command that you issue to run the Synonym Tool depends on whether the text search server is installed on a Linux, zLinux, or Windows server.

Table 31. Commands to run the Synonym Tool

On a Linux or zLinux server	On a Windows server
`synonymTool.sh synonym_command -configPath value [-locale value] -command_specific_arguments`	`synonymTool.bat synonym_command -configPath value [-locale value] -command_specific_arguments`

Global arguments

-configPath
Specifies the fully qualified path to the configuration directory, such as `/opt/ibm/search/config`. If this value contains blanks, you must enclose the value in quotation marks. This global argument is mandatory.

-locale
Specifies the five-character locale code, such as en_US for English or de_DE for German. The default value is en_US.

Command options

importSynonym
Adds a synonym dictionary to a collection, or updates the synonym dictionary for a collection.

-synonymFile
Specifies the fully qualified path to the synonym file that contains the updated synonyms.

-replace
Specifies whether to replace the existing synonym file with the updated synonyms (true), or append the updated synonyms to the existing synonym file (false).

-collectionName
Specifies the name of the collection for which synonyms are to be updated.

removeSynonym
Deletes all of the synonyms for a collection.

-collectionName
Specifies the name of the collection for which synonyms are to be removed.

help
Prints the usage information for this tool.

Examples

Example 1: On a Linux server, use the following command to append updated synonyms in the Default collection:

```
synonymTool.sh importSynonym
-synonymFile /fullpath/synonym.xml
-collectionName Default
-replace false
-configPath /opt/ibm/search/config
```

Example 2: On a Windows server, use the following command to remove all synonyms from the Default collection:

```
synonymTool.bat removeSynonym
-collectionName Default
-configPath c:\\Program Files\\IBM\\Search\\config
```

Related tasks

"Adding a synonym dictionary to a collection" on page 99
Use the Synonym Tool to add a synonym dictionary to a specific collection. Specifying synonym groups in a synonym dictionary improves the quality of text search results.

"Removing a synonym dictionary from a collection" on page 100
Use the script that is provided to remove a synonym dictionary from a collection. The text search server administrator needs to retrieve the name of the collection from which you want the synonym dictionary removed.

Chapter 8. Text search administration tables

When you enable text search support for Db2 for z/OS, the database SYSIBMTA is created with various database objects in the schema SYSIBMTS, along with specific administration tables that facilitate text search support.

SYSIBMTS.SYSTEXTDEFAULTS administration table

The SYSIBMTS.SYSTEXTDEFAULTS administration table provides the default parameters and values for the database SYSIBMTA, which is created with schema SYSIBMTS when you enable text search support for Db2 for z/OS.

The following table shows the contents of the SYSIBMTS.SYSTEXTDEFAULTS administration table.

Table 32. Contents of the SYSIBMTS.SYSTEXTDEFAULTS administration table

Column name	Data type	Nullable?	Description
NAME	VARCHAR(30)	No	Name of a default parameter for the database for text search.
VALUE	VARCHAR(512)	No	Value for the default parameter for text search.
TYPE	INTEGER	No	Reserved.

SYSIBMTS.SYSTEXTINDEXES administration table

The SYSIBMTS.SYSTEXTINDEXES administration table provides information about each text search index, such as the name and the schema name of the text search index and the name of the associated collection on the text search server.

The following table shows the contents of the SYSIBMTS.SYSTEXTINDEXES administration table. The unique key for this table is the INDEXID column in conjunction with the INDEXSCHEMA column. The primary key is the INDEXID column.

Table 33. Contents of the SYSIBMTS.SYSTEXTINDEXES administration table

Column name	Data type	Nullable?	Description
INDEXID	INTEGER	No	Uniquely generated index ID for the text search index.
INDEXSCHEMA	VARCHAR(128)	No	Schema name for the text search index.
INDEXNAME	VARCHAR(128)	No	Unqualified name of the text search index.
TABLESCHEMA	VARCHAR(128)	No	Schema name of the base table.
TABLENAME	VARCHAR(128)	No	Unqualified name of the base table.
COLLECTIONNAME	VARCHAR(128)	No	Name of the associated collection on the text search server.
SERVERID	INTEGER	No	The server ID for the text search index.

© Copyright IBM Corp. 2007, 2019

Table 33. Contents of the SYSIBMTS.SYSTEXTINDEXES administration table (continued)

Column name	Data type	Nullable?	Description
TAKEOVERSERVERID	INTEGER	Yes	The server ID that a currently running SYSPROC.SYSTS_TAKEOVER or SYSPROC.SYSTS_RESTORE stored procedure is using to take over text search functions for the text search index. When the SYSPROC.SYSTS_TAKEOVER stored procedure is not running, this value is null.
TAKEOVERSERVERPULSE	TIMESTAMP	Yes	The pulse that is used to monitor the progress of a takeover or restore operation.
SEARCHARGS	VARBINARY(1024)	Yes	Reserved for future use.
ALIASSCHEMA	VARCHAR(128)	No	The alias for the schema of the base table that was used in the SYSPROC.SYSTS_CREATE stored procedure. If no alias is used, this value is identical to TABLESCHEMA.
ALIASNAME	VARCHAR(128)	No	The alias for the name of the base table that was used in the SYSPROC.SYSTS_CREATE stored procedure. If no alias is used, this value is identical to TABLENAME.
STAGINGTABLENAME	VARCHAR(128)	Yes	The name of the log table for the text search index.
EVENTTABLENAME	VARCHAR(128)	No	The name of the event table for the text search index.
OFINDEXTABLENAME	VARCHAR(128)	No	The name of the table for the text search index on the text search server.
UPDATEMINIMUM	INTEGER	No	Minimum number of entries in the log table before an incremental update of the text search index is performed.
UPDATEFREQUENCY	VARCHAR(512)	No	The update frequency for the text search index as specified by the SYSPROC.SYSTS_CREATE stored procedure.
UPDATEMODE	INTEGER	No	Indicates the update mode of the text search index. The integer 0 (zero) indicates the initial update of the text search index. A value of 1 indicates subsequent, incremental updates.
CREATETIME	TIMESTAMP	No	The time that the text search index was created.
LASTUPDATETIME	TIMESTAMP	Yes	The time that the text search index was last updated.

Table 33. Contents of the SYSIBMTS.SYSTEXTINDEXES administration table (continued)

Column name	Data type	Nullable?	Description
LASTUPDATESTATUS	CHAR	Yes	Indicates the internal status for optimizing the clean-up process after an initial or incremental update of the text search index. The following values are supported: • C: If an initial update is run again, the collection must be cleared. No clean-up is required for an incremental update. • N: No clean-up is required for the incremental update. • U: Rows in the log table must be unmarked. • D: The specified rows in the log table must be deleted.
SCHEDULERTASKID	INTEGER	Yes	Reserved for future use.
EXPRESSIONLISTS	CLOB(32K)	No	Reserved for future use.
EXPRESSIONNUMBERS	VARBINARY(32)	No	Reserved for future use.
USEREXITFUNCTION	VARCHAR(18)	Yes	Reserved for future use.
REMARKS	VARCHAR(512)	Yes	Remarks made in the **COMMENTS** option of the **index-configuration-options** parameter of the SYSPROC.SYSTS_CREATE stored procedure.

SYSIBMTS.SYSTEXTCOLUMNS administration table

The SYSIBMTS.SYSTEXTCOLUMNS administration table provides information about the text column for a text search index, such as the index ID for the text search index, the name of the text column, and the schema name of the base table.

The following table shows the contents of the SYSIBMTS.SYSTEXTCOLUMNS administration table. The primary key for this table is the INDEXID column in conjunction with the COLUMNNAME column. The foreign key is the INDEXID column.

Table 34. Contents of the SYSIBMTS.SYSTEXTCOLUMNS administration table

Column name	Data type	Nullable?	Description
INDEXID	INTEGER	No	Uniquely generated index ID for the text search index.
COLUMNNAME	VARCHAR(128)	No	Unqualified name of the text column.
TABLESCHEMA	VARCHAR(128)	No	Schema name of the base table.
TABLENAME	VARCHAR(128)	No	Unqualified name of the base table.

Table 34. Contents of the SYSIBMTS.SYSTEXTCOLUMNS administration table (continued)

Column name	Data type	Nullable?	Description
LANGUAGE	VARCHAR(5)	No	The language that the text search server uses for the linguistic processing of text documents. The default value is en_US (English).
FUNCTIONSCHEMA	VARCHAR(128)	Yes	The schema of a user-defined function that is to be used by the text search feature for Db2 for z/OS to access text documents that are in a column that is not of a supported data type, or that are stored elsewhere.
FUNCTIONNAME	VARCHAR(18)	Yes	The name of a user-defined function that is to be used by the text search feature for Db2 for z/OS to access text documents that are in a column that is not of a supported data type, or that are stored elsewhere.
CCSID	INTEGER	No	The coded character set identifier that is used for a text search index on a column with a binary data type.
FORMAT	VARCHAR(30)	No	The format of text documents in the column. The supported format values are TEXT, HTML, XML, and INSO.

SYSIBMTS.SYSTEXTSERVERS administration table

The SYSIBMTS.SYSTEXTSERVERS administration table stores information about where the text search servers are installed.

The following table shows the contents of the SYSIBMTS.SYSTEXTSERVERS administration table. The unique key for this table is the SERVERNAME column in conjunction with the SERVERPORT column. The primary key is the SERVERID column.

Table 35. Contents of the SYSIBMTS.SYSTEXTSERVERS administration table

Column name	Data type	Nullable?	Description
SERVERID	INTEGER	No	Uniquely generated ID for the text search server.
SERVERNAME	VARCHAR(128)	No	The host name or IP address of the text search server.
SERVERADRINFO	VARBINARY(3000)	Yes	The internal representation of the SERVERNAME and SERVERPORT as determined by the SYSPROC.SYSTS_START stored procedure.
SERVERPORT	INTEGER	No	The port number for the text search server.
SERVERRESERVED	VARCHAR(128)	Yes	Reserved for future use.
SERVERAUTHTOKEN	VARCHAR(256)	No	The authentication token for the text search server.

Table 35. Contents of the SYSIBMTS.SYSTEXTSERVERS administration table (continued)

Column name	Data type	Nullable?	Description
DB2ENCRYPTEDPW	VARCHAR(256)	Yes	The encrypted Db2 password that is used by the user ID in the SYSIBMTS.SYSTEXTCONNECTINFO table to connect to the Db2 subsystem from the text search server.
SERVERMASTERKEY	VARCHAR(36)	No	The server key for the text search server.
STATUS	INTEGER	No	Indicates whether the server can be used as a text search server to create new text search indexes. The default value is 0 (zero), which means that the server can be used.

SYSIBMTS.SYSTEXTCONNECTINFO administration table

The SYSIBMTS.SYSTEXTCONNECTINFO administration table stores information about the Db2 for z/OS connection information so that a text search server can connect to the Db2 subsystem.

The following table shows the contents of the SYSIBMTS.SYSTEXTCONNECTINFO administration table.

Table 36. Contents of the SYSIBMTS.SYSTEXTCONNECTINFO administration table

Column name	Data type	Nullable?	Description
DB2HOSTNAME	VARCHAR(128)	No	The host name or IP address of the Db2 host system.
DB2SERVICEPORT	VARCHAR(128)	No	The service name (port number) for the Db2 subsystem.
DB2UID	VARCHAR(128)	No	The user ID that the text search server uses to connect to the Db2 subsystem.

SYSIBMTS.SYSTEXTSTATUS administration table

The SYSIBMTS.SYSTEXTSTATUS administration table contains information that is generated after the SYSPROC.SYSTS_START stored procedure is invoked to start Db2 text search functions. This table also indicates whether text search support is enabled.

The following table shows the contents of the SYSIBMTS.SYSTEXTSTATUS administration table.

Table 37. Contents of the SYSIBMTS.SYSTEXTSTATUS administration table

Column name	Data type	Nullable?	Description
SYSTEMSTATUS	INTEGER	No	Indicates the status of Db2 text search functions. This column supports the following values: • `0 (zero)`: Text search functions are stopped. • `1`: Text search functions are started. • `-1`: Text search functions are not installed.

Table 37. Contents of the SYSIBMTS.SYSTEXTSTATUS administration table (continued)

Column name	Data type	Nullable?	Description
VERSION	INTEGER	No	The internal version identifier for the text search product.

SYSIBMTS.SYSTEXTCONFIGURATION administration table

The SYSIBMTS.SYSTEXTCONFIGURATION administration table contains information about the configuration parameters for the text search index as passed by the SYSPROC.SYSTS_CREATE stored procedure.

The following table shows the contents of the SYSIBMTS.SYSTEXTCONFIGURATION administration table. The primary key is the INDEXID column in conjunction with the PARAMETER column. The foreign key is the INDEXID column.

Table 38. Contents of the SYSIBMTS.SYSTEXTCONFIGURATION administration table

Column name	Data type	Nullable?	Description
INDEXID	INTEGER	No	Uniquely generated index ID for the text search index.
PARAMETER	VARCHAR(30)	No	Parameters that are specified for the text search index in the SYSPROC.SYSTS_CREATE stored procedure.
VALUE	VARCHAR(512)	No	Values for the specified parameters.

SYSIBMTS.SYSTEXTLOCKS administration table

The SYSIBMTS.SYSTEXTLOCKS administration table is an auxiliary table that is used by the administrative stored procedures to implement the exclusion of stored procedures from running on a specific text search index.

The following table shows the contents of the SYSIBMTS.SYSTEXTLOCKS administration table.

Table 39. Contents of the SYSIBMTS.SYSTEXTLOCKS administration table

Column name	Data type	Nullable?	Description
INDEXID	INTEGER	No	Reserved for internal index-level locking.

SYSIBMTS.SYSTEXTSERVERHISTORY administration table

The SYSIBMTS.SYSTEXTSERVERHISTORY administration table is an auxiliary table that records a history of used servers for the SYSPROC.SYSTS_DROP stored procedure.

The following table shows the contents of the SYSIBMTS.SYSTEXTSERVERHISTORY administration table. The foreign key is the INDEXID column.

Table 40. Contents of the SYSIBMTS.SYSTEXTSERVERHISTORY administration table

Column name	Data type	Nullable?	Description
INDEXID	INTEGER	No	The index ID for a created text search index.

Table 40. Contents of the SYSIBMTS.SYSTEXTSERVERHISTORY administration table (continued)

Column name	Data type	Nullable?	Description
SERVERID	INTEGER	No	The server ID where a text search index needs to be dropped on SYSPROC.SYSTS_DROP.

Chapter 9. Troubleshooting

Use this information to help you diagnose, recover from, and resolve problems.

Supporting concurrent index updates and search requests

A text search server consumes system resources, like file descriptors, to handle multiple updates and search requests for text search indexes.

About this task

In a typical system, the number of open file descriptors for each process might be limited to a smaller number, such as 1024. If the text search server runs out of file descriptors, you might observe the following symptoms:

- The update and search requests fail even though the text search server is running.
- The problem might be logged in the server logs. The server logs show an exception with a message string that is similar to `". . . too many open files."`

Procedure

To resolve the issue of running out of file descriptors:

1. Shutdown the text search server.
2. Increase the number of file descriptors that are allowed for each process by following the manual for your operating system.

   This increase in file descriptors must hold across login sessions.
3. Restart the text search server.

Related tasks

“Starting a text search server” on page 99
You can start a text search server manually by using the startup script that is provided.

“Stopping a text search server” on page 99
You can stop a text search server manually by using the shutdown script that is provided.

Preventing a timeout abend

The stored procedures for update, takeover, and restore operations on a text search index might run a long time for tables that have many rows.

About this task

If the stored procedure calls are submitted as a job in the TSO environment, an ABEND CODE=522 can occur. The job stops, because the time limit for the job has been exceeded.

Procedure

In the job, add the parameter TIME=NOLIMIT after the JOB and EXEC statements.

Troubleshooting invalid entries in administration tables

Invalid entries in the SYSIBMTS.SYSTEXTSERVERS and SYSIBMTS.SYSTEXTCONNECTINFO administration tables can result in messages that you can easily troubleshoot.

The following list provides descriptions of the possible invalid entries in the SYSIBMTS.SYSTEXTSERVERS and SYSIBMTS.SYSTEXTCONNECTINFO administration tables, the messages that result, and how you can troubleshoot these issues.

© Copyright IBM Corp. 2007, 2019

- Incorrect values for SERVERAUTHTOKEN, SERVERNAME, or SERVERPORT in the SYSIBMTS.SYSTEXTSERVERS administration table result in the following message from the SYSTS_START stored procedure:

  ```
  OF00317E [n] text search servers are not available. Check the console messages
  for details.
  ```

  This message is accompanied by one or more console messages for the servers that fail:

  ```
  +DSN5001I  ERROR OCCURRED WHILE CONTACTING OMNIFIND SERVER [server name]:
  [port number] FOR SYSPROC.SYSTS_START ON COLLECTION
  ```

  To troubleshoot this issue, compare the values for SERVERAUTHTOKEN, SERVERNAME, and SERVERPORT, and verify that the server name and port number in the console message are correct for the text search server. Also, verify that you inserted the correct, case-sensitive SERVERAUTHTOKEN value from the text search server.
- Incorrect values for SERVERAUTHTOKEN, SERVERNAME, or SERVERPORT in the SYSIBMTS.SYSTEXTSERVERS stored procedure might result in the following messages from the SYSTS_CREATE stored procedure:

  ```
  OF00325E No text search server in table 'SYSIBMTS'.'SYSTEXTSERVERS' was able to
  perform the requested operation. Check the table entries, server, and network
  status. OF00801E A text search engine operation failed. Message code:
  ERR_CONNECT_SERVER. Exception oss::BaseException: "A connection could not be
  established with server [server name] on port [port number]. " was thrown in
  file /u/BCKE/ofbuild.851/wc.31bit/sutterr3c/src/components/comm/src/
  cieWhitneyClient.cpp line 240.
  ```

  To troubleshoot this issue, compare the values for SERVERAUTHTOKEN, SERVERNAME, and SERVERPORT, and verify that the server name and port number in the console message are correct for the text search server. Also, verify that you inserted the correct, case-sensitive SERVERAUTHTOKEN value from the text search server.
- An incorrect value for DB2ENCRYPTEDPW in the SYSIBMTS.SYSTEXTSERVERS table results in the following messages from the SYSTS_CREATE stored procedure:

  ```
  DSN20427E ERROR OCCURRED DURING TEXT SEARCH ADMINISTRATION STORED PROCEDURE
  OF00325E No text search server in table 'SYSIBMTS'.'SYSTEXTSERVERS' was able
  to perform the requested operation. Check the table entries, server, and
  network status. OF00800E A text search server operation failed. Message code:
  IQQG0020E. Exception  : 'IQQG0020E ExtendedException: [...]'  where [...]
  stands for the reason codes from the text search server.
  ```

  Update the DB2ENCRYPTEDPW value by calling the SYSTS_ENCRYPT function with the correct input parameters. These input parameters are the SERVERMASTERKEY from the same row in the SYSIBMTS.SYSTEXTSERVERS table and the password for the user ID that you inserted in DB2UID in the SYSIBMTS.SYSTEXTCONNECTINFO table. Verify that you inserted the correct, case-sensitive SERVERMASTERKEY value from the text search server.

  Incorrect values for DB2HOSTNAME, DB2SERVICEPORT, or DB2UID in the SYSIBMTS.SYSTEXTCONNECTINFO table also result in a DSN20427E message from the SYSTS_CREATE stored procedure. You can troubleshoot this issue in the same way.

Troubleshooting SQL code -430

Use this information if an administration stored procedure, such as SYSPROC.SYSTS_START, returns SQL code -430.

About this task

The message text reads as follows:

```
SQLCODE = -430, ERROR:  PROCEDURE SYSPROC.SYSTS_START
(SPECIFIC NAME SYSPROC.SYSTS_START) HAS ABNORMALLY TERMINATED
```

Procedure

Complete the appropriate actions based on the following information:

- If you received Db2 message DSNX962I, you must install the component IBM International Components for Unicode (ICU) for Db2 for z/OS, which is provided in the Db2 Accessories Suite for z/OS.
- If the ICU component is installed, you can run stored procedures by using the following ICU program objects. These ICU program objects are in the same SDSNLOAD data set that is used in the WLM application environment:
  - DSN5ICDA
  - DSN5ICIN
  - DSN5ICIO
  - DSN5ICUC
  - DSN5IDA3
  - DSN5IIN3
  - DSN5IIO3
  - DSN5IUC3

Related concepts

"System requirements for enabling text search support" on page 2
Before you enable text search support, ensure that your system meets all of the operating system and software requirements.

Related information

-430 (DB2 Codes)

Troubleshooting SQL code -20212

You might receive SQL code -20212 after running the CREATE FUNCTION SYSFUN.SYSTS_ENCRYPT statement.

Procedure

To resolve this problem:

1. Check whether the Java class name `'com.ibm.es.nuvo.util.CryptionUtil.encrypt'` displays in the message in mixed case letters.
2. If this class name is displayed in uppercase letters, run the CREATE FUNCTION SYSFUN.SYSTS_ENCRYPT statement again with the Java class name in mixed case letters.

   You can find this statement in Db2 installation job DSNTIJNX.
3. If the class name displays correctly in the message, verify that you have provided the correct DB2_BASE environment variable in the JAVAENV data set.

   A DB2_BASE statement is required only if the Db2 HFS files are installed in a directory other than `/usr/lpp/db2b10/jdbc`.

Related information

-20212 (DB2 Codes)

Troubleshooting SQL code -20423 with message OF00801E

Even though you can run administration stored procedures, SQL code -20423 with message OF00801E might be returned for only the CONTAINS function and the SCORE function.

About this task

The message text reads as follows:

```
SQLCODE = -20423, ERROR:  ERROR OCCURRED DURING TEXT SEARCH PROCESSING
(server, indexname, OF00801E A text search engine operation failed.
Message code: ERR_CONNECT_SERVER. Exception oss::BaseException:
"A connection could not be established with server <server> on port
<port number>." was thrown [...].)
```

Procedure

To resolve this problem:

1. Assign an OMVS segment to the RACF user ID that is associated with the DSNDBMI address space started task.

   Typically, this user ID is SYSDSP.
2. Restart Db2 for z/OS.

Related concepts

“System requirements for enabling text search support” on page 2
Before you enable text search support, ensure that your system meets all of the operating system and software requirements.

Related information

-20423 (DB2 Codes)

Chapter 10. Messages and codes

You might encounter Db2 messages and codes for text search functions.

Related information

DSN5001I (DB2 Messages)

-20101 (DB2 Codes)

-20423 (DB2 Codes)

-20424 (DB2 Codes)

-20425 (DB2 Codes)

-20426 (DB2 Codes)

-20427 (DB2 Codes)

Text search server messages

This information provides the messages that you might encounter for IBM Text Search for Db2 for z/OS.

IQQA0134E **A problem occurred when communicating to the specified host name: "*host_name*" and port: "*port_number*".**

Explanation:
A problem occurred when communicating to the specified host_name and port_number. The server might be down or an incorrect name was specified.

User response:
Check that the host name and the port are valid and that the specified server is up and listening to the specified port.

IQQD0002E **An error occurred when serializing a message for the collection *collection_ID* for the action*action_name*.**

Explanation:
An error occurred when serializing a message to the client.

User response:
No action is required.

IQQD0005E **The collection *collection_ID* is closed.**

Explanation:
The collection is in a closed state.

User response:
Open the collection before adding documents to it.

IQQD0006E **Collection *collection_ID* does not exist.**

Explanation:
The collection does not exist.

User response:
Create the collection before trying to use it.

IQQD0009E **File *file_path* does not exist.**

Explanation:
The specified file does not exist.

User response:
No action is required.

IQQD0020E **The query length is greater than 4096 characters.**

Explanation:
The server received a query that is greater than 4096 characters.

User response:
Reduce the query to fewer than 4096 characters and run the query again.

IQQD0021E **The query is null.**

Explanation:
The query is null.

User response:
Run the query with valid query terms.

IQQD0022E **The query contains only white space characters.**

Explanation:
The query contains white space characters and no searchable query terms.

User response:
Run the query with valid query terms.

IQQD0023E **An I/O error occurred in the search runtime.**

Explanation:
During query processing, an I/O exception occurred in the search runtime.

User response:
Contact IBM Software Support.

© Copyright IBM Corp. 2007, 2019

IQQD0025E **The directory is already used by the server *server_name*.**

Explanation:
A server is using the specified directory.

User response:
Shut down the server before you run the configuration tool.

IQQD0027E **This action type is not supported.**

Explanation:
This action type is not supported.

User response:
No action is required.

IQQD0028E **A write error occurred when sending data to the client.**

Explanation:
A write error occurred when sending data to the client.

User response:
No action is required.

IQQD0029E ***command_name* is not a valid command.**

Explanation:
A required command option for the configuration tool was omitted.

User response:
Run the tool again and specify the correct command option.

IQQD0030E **A required command argument *argument* is missing.**

Explanation:
A required command argument was omitted.

User response:
Run the tool again and specify the correct command argument.

IQQD0031E **A required argument is missing.**

Explanation:
A required argument was omitted.

User response:
Run the tool again with the required argument.

IQQD0033E **Configuration file *file_name* cannot be renamed. Ensure that the disk or directory is not full.**

Explanation:
The configuration file could not be renamed.

User response:
Ensure that the disk or directory is not full and run the tool again.

IQQD0034E **The port {0} that was specified is already in use.**

Explanation:
The specified port number is not free.

User response:
Run the tool again and specify a port that is not already in use.

IQQD0036E **An instance of the search server is running and needs to be shut down.**

Explanation:
An instance of the search server is running and needs to be shut down.

User response:
Shut down the search server and then run the tool again.

IQQD0037E **Heap size *size* is not a valid JVM parameter. The value must be an integer followed by M or G (M=megabytes, G=gigabytes).**

Explanation:
The specified heap size parameter is not a valid JVM parameter. The value must be an integer followed by M or G (M=megabytes, G=gigabytes).

User response:
Run the tool again with a valid JVM heap size parameter.

IQQD0038E **Installation source *directory_name* is not a directory.**

Explanation:
The specified installation source is not a directory.

User response:
Provide a valid installation source and run the tool again.

IQQD0039W **Port number *port_number* is invalid.**

Explanation:
The specified port number is invalid.

User response:
Provide a valid port number and run the tool again.

IQQD0040E **The client specified the wrong authentication token.**

Explanation:
The client specified the wrong authentication token.

User response:
Specify the correct authentication token and retry the operation.

IQQD0041E **The directory *directory_name* could not be created.**

Explanation:
The directory could not be created.

User response:
Ensure that the disk or directory is not full and that you have the correct permissions. Then run the tool again.

IQQD0041W **The maximum document size parameter does not exist in file *file_name*.**

Explanation:
The maximum document size parameter does not exist.

User response:
Ensure that the maximum document size parameter exists in the file and run the tool again.

IQQD0047E **A runtime exception occurred: *error_code*.**

Explanation:
A runtime exception occurred.

User response:
Try to run the tool again with the correct parameters.

IQQD0048E **An invalid value for an argument was passed: *argument_value*.**

Explanation:
An invalid value for an argument was passed.

User response:
Try to run the tool again with a valid argument.

IQQD0049E **A "not OK" response was received from server *server_name*.**

Explanation:
A "not OK" response was received from the server.

User response:
Restart the search server and try to run the tool again.

IQQD0052W **A previous shutdown request is in progress.**

Explanation:
A shutdown action was previously started and is still in progress.

User response:
No action is required.

IQQD0053E **The shutdown request could not be executed successfully.**

Explanation:
The shutdown request was unable to run.

User response:
Forcefully stop the server JVM.

IQQD0054E **An error occurred when retrieving values from file *file_path*.**

Explanation:
Values could not be retrieved from the file.

User response:
Verify that the specified file exists and that it is not corrupt.

IQQD0055E **The search server is stopped. It must be started for the tool to run.**

Explanation:
The search server is shut down and it must be running before you can use this tool.

User response:
Start the search server and run the tool again.

IQQD0056E **An error occurred when starting the server at port *port_number*.**

Explanation:
The server could not be started at the specified port.

User response:
Ensure that port number is not in use.

IQQD0057E **Log level *level* is not a valid parameter value.**

Explanation:
The specified log level parameter is not a valid parameter.

User response:
Run the configuration tool again with a valid log level parameter.

IQQD0058E **The client failed to connect to the server on port *port_number* at host *host*.**

Explanation:
The server could not be started on the specified port.

User response:
Ensure that the port number is not in use.

IQQD0058I

Explanation:
NA

User response:
NA

IQQD0059I

Explanation:
NA

User response:
NA

IQQD0060W **Concurrent administration actions are not allowed on the same collection.**

Explanation:
A previous administration task is in progress.

User response:
Run the task after the previous task is finished.

IQQD0062E **No token was found.**

Explanation:
No token was found.

User response:
Use the configuration tool to generate the token.

IQQD0063I **The Text Search Server is being started.**

Explanation:
The Text Search Server is being started.

User response:
None

IQQD0064E **An error occurred when connecting to the server *hostname* at port *port_number*.**

Explanation:
The server could not be contacted at the specified port.

User response:
Ensure that server is listening at the specified address.

IQQD0066E **The system encountered a fatal IO error.**

Explanation:
The system encountered a fatal IO error. It could due to disk full or other write error.

User response:
For disk full please shutdown the text search server, free up some space and restart the server. For other write errors please check the file permissions on disk and retry your operation.

IQQD0067I **A request to delete the collection *collection_name* was received.**

Explanation:
A request to delete collection was recieved by the text search server .

User response:
None.

IQQD0068E **The posting list for the payload for the collection *collection_name* may be corrupted.**

Explanation:
The posting list for binary payload for collection *collection_name* is corrupted.

User response:
None.

IQQD0069W **The query *query_String* for collection *collection_name* cannot be processed because it has incorrect syntax.**

Explanation:
The query cannont be processed because it has incorrect syntax.

User response:
None.

IQQD0070E **Unable to obtain collection status for server *hostname* at port *port_number***

Explanation:
Unable to obtain the collection status.

User response:
Please check if the server is in consistent state and responding to requests and then retry the operation.

IQQD0071E **Unable to obtain server version for server *hostname* at port *port_number***

Explanation:
Unable to obtain the server version.

User response:
Please check if the server is in consistent state and responding to requests and then retry the operation.

IQQD0072E **An error occured while communicating with the server *hostname* at port *port_number* . Unable to send the request to the server or the client timed out while receeving a response from the server.**

Explanation:
An error occured while communicating with the server. Unable to send the request to the server or the client timed out while recieving a response from the server.

User response:
Please check if the server is in consistent state and responding to requests and then retry the operation.

IQQD0073E **Import synonyms for the synonym definitions in file *filename* for the collection *collection_name* failed. It could be due to synonym XML definition parsing failed or some other error.**

Explanation:

Import synonyms failed. It could be due to synonym XML definition parsing failed or some other error.

User response:
Please check if the synonym definitions has correct syntax.

IQQD0074E **Remove synonyms for the collection *collection_name* failed.**

Explanation:
Remove synonyms failed.

User response:
Please check if the server is in consistent state.

IQQD0075I **Request *command string* was received for collection *collection id*.**

Explanation:
Commands can be logged for debugging.

User response:
NA

IQQD0076I **The collection *collection_name* is being closed because it has timed out. It has been *nnumber_of_secs* milliseconds since last operation.**

Explanation:
Collection is being closed because it has timed out.

User response:
NA

IQQD0077E **Unable to start the server on default port *port number*. Try changing the port using configuration tool.**

Explanation:
The default port -1 can not be used to start the server.

User response:
Run Configuration tool change the port number and restart the server.

IQQD0081E **The directory *directory_name* is not writeable.**

Explanation:
Write to the specified directory failed

User response:
Check the permission of the directory.

IQQD0083E **The argument *argument* expects a fully qualified path. A relative path value of *path_name* was provided.**

Explanation:
File and directory paths are expected to be fully qualified (absolute) paths.

User response:
Rerun the commands with the correct fully qualified path.

IQQD0084I **The request was successfully executed.**

Explanation:
The last request was successfully executed.

User response:
null

IQQD0085E **The requested operation cannot be done. The server is running in safe mode due to *disk_in_resource_shortage* has an IO error.**

Explanation:
The requested operation can't be performed when system is in safe mode. If system detected IO error which seems be disk full, it turned into safe mode.

User response:
The message indicates "log" or "index" directory is full. Make more disk space available on that directory.

IQQD0087E **The client version is not compatible with the current server version.**

Explanation:
The client version is not compatible with the current server version.

User response:
NA

IQQD0088E ***size* is not a valid memory queue size. The value must be an integer followed by M, K, or B (M=megabytes, K=kilobytes, B=bytes). The default is 50M.**

Explanation:
The specified memory queue size is not valid. The value must be an integer followed by M, K, or B.

User response:
Run the tool again with a valid memory queue size value.

IQQD0089E ***size* is not a valid document batch size. The value must be an integer. The default is 15.**

Explanation:
The specified document batch size is not valid. The value must be an integer.

User response:
Run the tool again with a valid document batch size value.

IQQD0090E ***number* seconds is not a valid document timeout value. The**

value must be an integer. The default is 600.

Explanation:
The specified document timeout value is not valid. The value must be an integer.

User response:
Run the tool again with a valid document timeout value.

IQQD0091E ***number*** **is not a valid document retry number. The value must be an integer. The default is 1.**

Explanation:
The specified document retry value is not valid. The value must be an integer.

User response:
Run the tool again with a valid document retry value.

IQQD0092E **The specified number of preprocessing threads, *number*, is not a valid value. The value must be an integer. The default is 4.**

Explanation:
The specified number of preprocessing threads is not valid. The value must be an integer.

User response:
Run the tool again with a valid number of preprocessing threads.

IQQD0093E **The specified number of indexer threads, *number*, is not a valid value. The value must be an integer. The default is 4.**

Explanation:
The specified number of indexer threads is not valid. The value must be an integer.

User response:
Run the tool again with a valid number of indexer threads.

IQQD0094E **The specified server mode, {0}, is not a valid server mode. The server mode must be either indexing or preprocessing.**

Explanation:
The server mode must be either indexing or preprocessing.

User response:
Run the tool again and specify either indexing or preprocessing for the serverMode parameter.

IQQD0095E **The specified indexing server port value, {0}, is not valid. The port number must be a value between 0 and 65535.**

Explanation:
The indexing server port must be a value between 0 and 65535.

User response:
Run the tool again with a valid port number (between 0 and 65535).

IQQD0097E **The specified preprocessing server batch size, {0} MB, is not valid. The batch size must be a value between 0 and 1000.**

Explanation:
The preprocessing server batch size must be a value between 0 and 1000.

User response:
Run the tool again with a valid batch size (between 0 and 1000).

IQQD0098E **The indexingServerHost, indexingServerPort, indexingServerToken, and preprocessingServerBatchSize parameters are required only if the server mode is preprocessing.**

Explanation:
One or more of the specified parameters are required only if the server mode is preprocessing.

User response:
Run the tool again without the indexingServerHost, indexingServerPort, indexingServerToken, and preprocessingServerBatchSizeparameters.

IQQD0099E **When the server mode is set to preprocessing, you must specify all of the following parameters: indexingServerHost, indexingServerPort, indexingServerToken, and preprocessingServerBatchSize.**

Explanation:
Parameters are missing for the preprocessing server.

User response:
Run the tool again and specify all of the following parameters: indexingServerHost, indexingServerPort, indexingServerToken, and preprocessingServerBatchSize.

IQQD0100E **Another server is currently using the index.**

Explanation:
The index is being used by another server.

User response:
Wait until the other server finishes indexing and then start indexing with the current server.

IQQD0101E **The list of supported client versions cannot be returned for server *hostname* at port *port_number*.**

Explanation:
The list of supported client versions cannot be returned for the server.

User response:
Verify that the server is running and responding to requests. Then retry the operation.

IQQD0103E **Unable to upgrade the instance. The version specified by the installedConfigPath parameter must be newer than the version specified by the configPath parameter.**

Explanation:
The version specified by the installedConfigPath parameter must be newer than the version specified by the configPath parameter.

User response:
Run the tool again with right parameters.

IQQD0104E **The *property_name* property is missing from the *property_file_name* file.**

Explanation:
The *property_name* property must be specified in the *property_file_name* file.

User response:
Add the *property_name* property to the *property_file_name* file.

IQQD0105E **The value of the *property_name* property in the *property_file_name* file is not valid.**

Explanation:
The property value is not valid.

User response:
Change the value of the *property_name* property to a valid value.

IQQD0106E **The system cannot unlock the file *file_name*.**

Explanation:
The system cannot unlock the specified file.

User response:
Try to remove the file manually.

IQQD0107E **The secure and nonsecure ports cannot both be set to 0.**

Explanation:
A valid port number must be specified for the secure port, nonsecure port, or both ports.

User response:
Run the configuration tool and set the adminHTTPPort parameter, securePort parameter, or both parameters to a valid port number (other than 0).

IQQD0108E **The secure and nonsecure ports cannot be set to the same value *port number*.**

Explanation:
The secure and nonsecure ports cannot be set to the same value.

User response:
Run the configuration tool and set the adminHTTPPort or securePort parameter to another port number.

IQQD0109E **The *keyStore path* keystore does not exist.**

Explanation:
The specified keystore does not exist.

User response:
Run the configuration tool and specify the correct path to the keystore.

IQQD0111E **The client received a malformed response from server *server*.**

Explanation:
The client received a malformed response from the server.

User response:
If this error is returned when you try to connect to the server, verify that the host and port connection parameters are correct. If this error is returned after you connect to the server, restart the server.

IQQD0112E **The server lock file (config/config.xml.lock) cannot be opened. Details: *error_code* \nIf the server is running under a different user, shut down the server**

Explanation:
The server lock file cannot be opened. The server might be running under a different user.

User response:
If the server is running under a different user, shut down that instance of the server. Otherwise, manually delete the config/config.xml.lock file.

IQQG0003E **The input and output files must be different files. Correct the appropriate file setting and resubmit the request. Input File: *input_file* Output File: *output_file*.**

Explanation:
The same file was used for both input and output, and the files must be different.

User response:
Resubmit the request with the correct input file and output file.

IQQG0007E **The *argument_value* argument is not valid.**

Explanation:
An invalid argument was used.

User response:
See the other associated messages for more information. Specify a valid argument and try again.

IQQG0008E **The *argument_value* argument is missing.**

Explanation:
The missing argument is required.

User response:
See the other associated messages for more information. Add the missing argument and try again.

IQQG0009E **The argument *argument* has an invalid value of *argument_value*.**

Explanation:
The argument value is invalid for the specified argument.

User response:
See the other associated messages for more information. Correct the invalid argument value and try again.

IQQG0019I ***informational_message***

Explanation:
An external informational message that is not localized (translated) was encountered.

User response:
No action is required.

IQQG0020E ***error_message***

Explanation:
An external error message that is not localized (translated) was encountered.

User response:
Check the specified error message for suggestions on how to solve the problem.

IQQG0026E **The configuration file *file_path* has an error.**

Explanation:
The file has an error.

User response:
If the file was edited, restore it back to the original version.

IQQG0028E **An object with name *cached_object* already exists in cache *cache_name*.**

Explanation:
The named object cannot be added to the cache because an object with that name is already in the cache. The names of cached objects must be unique.

User response:
Contact IBM Software Support.

IQQG0029E **The object cannot be added into cache *cache_name* with an invalid name of *invalid_value*.**

Explanation:
The object cannot be added to the cache because the object has an invalid name.

User response:
Contact IBM Software Support.

IQQG0030E **The cache loader for cache *cache_name* is null. The cached object insertion failed.**

Explanation:
The provided cache loader is null. A non-null value for the cache loader must be provided to the cache.

User response:
Contact IBM Software Support.

IQQG0032E **The following files in collection *collection_name* could not be deleted: \n *file_list*.**

Explanation:
The files were not deleted when the collection was removed.

User response:
Shut down the product and manually delete the files.

IQQG0037W **The collection *collection_name* does not exist.**

Explanation:
The collection name that you entered does not exist.

User response:
Specify the name of an existing collection.

IQQG0038E **The collection cannot be created because another collection has the same name *collection_name*.**

Explanation:
Each collection name must be unique.

User response:
Specify a different collection name.

IQQG0039E **The search engine installation directory *installation_directory* does not exist.**

Explanation:
The search engine installation directory does not exist. The directory is specified by the installPath element in the global configuration file (config.xml) that is created when the product is started.

User response:
Ensure that the installPath element points to the installation directory.

IQQG0040E **The config directory *config_directory* does not exist.**

Explanation:
The directory called config is specified by the configPath element in the global configuration file (config.xml) that is created when the product is started.

User response:
Ensure that the configPath element points to the config directory. The config directory must contain a subdirectory called "collections" and a file called "jetty.xml."

IQQG0041E **The installation directory *installation_directory* is missing the following files: *file_names*.**

Explanation:
The installation directory is missing some files. This directory is specified by the installPath element in the global configuration file (config.xml) that is created when the product is started.

User response:
The configuration might be pointing to the wrong directory. Ensure that the installPath element points to the installation directory.

IQQG0042E **The dictionaries directory *stopword_directory* does not exist.**

Explanation:
The directory called dictionaries for stopword is missing.

User response:
Ensure that the stop words directory is exist and contain dictionaries."

IQQG0045E **The global configuration file *global_configuration_file* does not exist.**

Explanation:
The specified global configuration file does not exist.

User response:
Specify the fully qualified name of the global configuration file. The default file is config.xml in the config subdirectory.

IQQG0046I **The index is currently processing documents in the queue. Current® document count is *queued_documents_number*.**

Explanation:
The system is waiting for the index to process the remaining documents.

User response:
No action is required. The system will shut down after the index finishes processing the queued documents, or if there is no progress.

IQQG0047E **The server cannot start because an invalid server mode [*given_server_mode*] was specified.**

Explanation:
The specified server mode is not valid.

User response:
Specify either 'indexing' or 'preprocessing' for the server mode configuration parameter.

IQQG0048E **The document timeout limit was exceeded for document [*document uri*] of collection [*collection_name*].**

Explanation:
The specified document timeout was exceeded. The document is returned to the indexing server's input queue for preprocessing.

User response:
No action required.

IQQG0049E **Document [*document uri*] of collection [*collection_name*] was removed because the document timeout and document retry values were exceeded.**

Explanation:
The document was removed because the document timeout and document retry values were exceeded.

User response:
No action required.

IQQG0051E **Document [*document uri*] of collection [*collection_name*] was already indexed or removed because the timeout was exceeded.**

Explanation:
The document was already processed, probably by another preprocessing server, and indexed. Therefore this instance of the document is ignored.

User response:
No action required.

IQQG0051I The index processing is stopping.

Explanation:
The index processing is stopping.

User response:
No action is required.

IQQG0052I Index processing is stopped.

Explanation:
Index processing is stopped.

User response:
No action is required.

IQQG0053E The system is missing the following file: *file_path*.

Explanation:
The system is missing a file that was present during installation.

User response:
Re-install the product to restore the file.

IQQG0054W The system cannot create the temporary directory *temporary_directory*. The default directory *directory_name* will be used.

Explanation:
The system cannot create the temporary directory. The system will continue running by using the other specified directory.

User response:
Ensure that the temporary directory exists and has the appropriate write permissions. The temporary directory is specified by either the tempDirectory element in the global configuration file (the default file is config.xml in the config subdirectory), or by the system temporary directory if the configuration file element is not specified.

IQQG0055E The data buffer cache was removed.

Explanation:
A request was made to read the cached data that was already removed.

User response:
Contact IBM Software Support.

IQQG0056E The data buffer cache cannot be read.

Explanation:
A request was made to read the cache data before it is available.

User response:
Contact IBM Software Support.

IQQG0065E The collection *collection_name* cannot be updated because it does not exist.

Explanation:
The collection cannot be updated because the collection was removed or never existed.

User response:
Re-create the collection if needed.

IQQG0067W The system settings cannot be restored from the file *file_path*.

Explanation:
The settings cannot be restored. Therefore, the settings reset to their default values.

User response:
Reset the settings in the administration console.

IQQG0071W The system cannot save information that indicates whether crawlers are active when the system is shut down.

Explanation:
When the system is shut down, information about crawling activity is saved. However, the system cannot save information that indicates whether crawlers for an collection were active when the system was shut down.

User response:
See the other associated messages for more information. When you restart the system, the collections with active crawlers after the system starts might not match the collections that had active crawlers when the system was shut down. After you restart the system, open the administration console and start or stop crawling as needed.

IQQG0073E The component cannot be removed for collection *collection_name*.

Explanation:
A request was made to delete a component of the collection. However, the component could not be removed.

User response:
See the other associated messages for more information. After you resolve the problem, restart the system. After you restart the system, all pending removal requests are processed.

IQQG0074E The system could not delete the following files: *file_paths*.

Explanation:
The system could not remove the files.

User response:
Manually remove the specified files.

IQQG0075E **The collection *collection_name* could not be cleared.**

Explanation:
The system could not clear the collection.

User response:
See the other associated messages for more information. After you resolve the problem, restart the system. During a restart, all pending clear operations are processed.

IQQG0083W **The system could not delete the following files when removing collection *collection_name*: *file_paths***

Explanation:
The collection was removed, but the removal operation could not remove the listed files.

User response:
Manually remove the files that are listed in the message.

IQQG0086E **The system cannot initialize the error event notification service. The system will continue to start, but event notifications in the administration console are unavailable.**

Explanation:
The event manager failed to initialize. The system will still run, but the event notifications in the administration console are unavailable.

User response:
See the other associated messages for more information. After you fix the problem, restart the system.

IQQG0088I **The shutdown request is being sent: *URL*.**

Explanation:
This message provides progress information that shows the status of the shutdown request.

User response:
No action is required.

IQQG0089I **The shutdown request was sent successfully with a response code of *response_code*.**

Explanation:
The shutdown request was sent successfully. A response code of 200 is normal.

User response:
No action is required.

IQQG0090E **The shutdown request failed. The server might not be running at the specified port or is already stopped.**

Explanation:
The shutdown request failed. The server might not be running at the specified port or is already stopped. A connection that is refused means that the server is not running at the specified port.

User response:
See the other associated messages for more information.

IQQG0103E **The directory name *directory_name* contains an invalid character *invalid_character*.**

Explanation:
The file system cannot create a directory that contains the specified character.

User response:
Specify a name that does not contain the invalid character.

IQQG0104E **The directory name *directory_name* is not valid.**

Explanation:
The file system cannot create a directory with the specified name.

User response:
Specify a directory name that is valid for your operating system.

IQQG0105E **The directory *directory_name* cannot be created.**

Explanation:
The file system cannot create a directory with the specified name.

User response:
Verify that the file system has available disk space. You can also try to specify a different name.

IQQG0107E **The encryption key *file_path* could not be loaded.**

Explanation:
The encryption key cannot be loaded from the specified file.

User response:
If the encryption key file was modified, restore it to its original state. A new encryption key can be generated by deleting this file, but any saved passwords cannot be decrypted.

IQQG0113E **The original password did not match the existing password.**

Explanation:
The attempt to change the password failed because the original password given and the existing password do not match. Note that passwords are case sensitive.

User response:
Correct the password and resubmit the request.

IQQG0114E **The operation cannot be completed because the system is shutting down.**

Explanation:
The operation failed because the system is processing a shutdown request.

User response:
Try the action again after the system is restarted.

IQQG0116E **The total number of indexed documents *number_of_documents* exceeds the allowable limit of *number_of_documents_limit* documents.**

Explanation:
The license agreement allows for the product to index only a predefined number of documents.

User response:
Remove some entries from the Web sites or directories that are being crawled.

IQQG0119E **The following files could not be deleted: \n *file_list*.**

Explanation:
The files could not be deleted.

User response:
Shut down the product and manually delete the files.

IQQG0120E **The configuration specified by *configuration_file* is already used by a running server.**

Explanation:
The server could not be started because there is another server that is using the same configuration files. Each instance of the server must have its own set of configuration files.

User response:
Stop the existing running instance before you restart another instance for the same configuration files.

IQQG0122E **The server could not be started with command *command_string*.**

Explanation:
The system failed to start after issuing the command as shown.

User response:
Check the log file startupErrors.html for details about this error. If this file does not exist, try running the specified command from the command line to see any error messages issued by the Java Virtual Machine (JVM).

IQQG0127E ***product_name* is not running.**

Explanation:
An attempt was made to stop *product_name*, but the system was not running.

User response:
No action is required.

IQQG0128E **Environment variable LD_ASSUME_KERNEL is currently set, which can cause problems when running *product_name*. No value should be specified for this variable.**

Explanation:
product_name was started on a Linux system with environment variable LD_ASSUME_KERNEL set, which can cause problems.

User response:
Delete the value that is set for this variable and restart *product_name* to avoid potential problems.

IQQG0139E **A file could not be created in temporary directory *directory*. Ensure that the program can create files in that directory.**

Explanation:
The server could not create a file in the directory that was given as the directory for temporary files.

User response:
Ensure that the server has authority to create files in that directory and that the disk is not full.

IQQG0140E **The system could not load the configuration file *configuration file*.**

Explanation:
The system could not load a required configuration file. The file might not exist or it might be corrupted.

User response:
Check the installation log file for additional messages about this error.

IQQG0141E **The configuration key *configuration_key* for component *component_id* has a non-Boolean value of *value*.**

Explanation:
The configuration key was expected to be a Boolean value, but it has a non-Boolean value.

User response:

Ensure that the configuration key has a valid Boolean value. Valid Boolean values are "true", "yes", "on", "1", "false", "no", "off", "0".

IQQG0142E **The configuration key *configuration_key* for component *component_id* has a non-integer value of *value*.**

Explanation:
The configuration key was expected to be an integer, but it has a non-integer value.

User response:
Ensure that the configuration key has a valid integer value.

IQQG0143E **The Text Extractor cannot start. On the Linux platform, verify that the libstdc++.so.5 library exists in the /usr/lib folder (on 32-bit systems) or in /usr/lib64 (on 64-bit systems). The Text Extractor executable file is named "textExtractor" and is located in the *ECMTS_HOME*/stellent.*platform_name* directory.**

Explanation:
The Text Extractor processes cannot start.

User response:
See the other associated messages for more information. Then, restart the system. On the Linux platform, verify that the libstdc++.so.5 library exists in the /usr/lib folder (on 32-bit systems) or in /usr/lib64 (on 64-bit systems).

IQQG0144E **No shared child processes are available in the system.**

Explanation:
An attempt was made to retrieve a shared child process, but none are available.

User response:
The system might be too busy. Restart the system.

IQQG0145E **An error occurred when a new child process was created.**

Explanation:
An error occurred when a new child process was being created in the system.

User response:
See the other associated messages for more information. Then, restart the system.

IQQG0146E **The child process stopped with the following return code: *process_return_code*.**

Explanation:
An error occurred when a new child process was being created in the system.

User response:
See the other associated messages for more information. Then, restart the system.

IQQG0147E **An error occurred when communicating with a child process.**

Explanation:
An error occurred when communicating with a child process in the system. For example, the system might be too busy.

User response:
See the other associated messages for more information. Then, restart the system.

IQQG0148E **A child process in the system went into an unexpected state: *process_state*.**

Explanation:
A child process in the system went into an unexpected state. For example, the system might be too busy.

User response:
See the other associated messages for more information. Then, restart the system.

IQQG0149E **The system cannot create the directory *directory_name*.**

Explanation:
The system cannot create the specified directory. The directory was specified in a configuration file.

User response:
See the other associated error messages for more information. Ensure that the program has permission to create the directories and that the disk is not full.

IQQG0150E **The system could not determine which release is installed. The missing configuration element is "*configuration_key*".**

Explanation:
The system is unable to resolve the "release" configuration element to determine the release type.

User response:
Ensure that the configuration key has a valid value in the properties file. The default file that the system reads is configPath/releaseinfo/release.properties. Ensure that this file is readable and is not corrupt.

IQQG0151E **The system is configured with an invalid release type: "*release_type*".**

Explanation:

The system is configured with a value for the release element which is not valid.

User response:
Ensure that the release element has a valid value in the properties file. The default file that the system reads is configPath/releaseinfo/release.properties. Ensure that this file is not corrupt.

IQQG0152E **The data file *file_path* does not exist.**

Explanation:
The file that is buffering the data of a document on disk does not exist.

User response:
Check the file name and make sure that there is no problem with writing to the file.

IQQG0153E **The document *document_uri* was ignored.**

Explanation:
The document was ignored when the system was restarted.

User response:
See the other associated error messages for more information. The document needs to be recrawled. Modify the document, and the crawler will include it during the next crawl.

IQQG0154E **The document queue could not be saved to file *file_path*. The following documents need to be recrawled: *document_list***

Explanation:
The document queue could not be saved.

User response:
See the other associated error messages for more information. The documents listed in the message will need to be recrawled. Modify the documents, and the crawler will include them during the next crawl.

IQQG0155E **The document queue could not be loaded from file *file_path*.**

Explanation:
The document queue could not be restored from disk.

User response:
See the other associated error messages for more information. The documents that were in the queue are discarded, and the system starts with an empty queue.

IQQG0156E **Collection *collection_name* is not available because it is in the *state* state.**

Explanation:
The collection cannot be returned because it is in the specified state.

User response:
Wait for the current activity to finish and then try the action again.

IQQG0163W **During the last shutdown, Windows Services terminated the server before shutdown could be completed.**

Explanation:
The server was not able to complete the shutdown in the amount of time allowed by Windows services.

User response:
Increase the amount of time Windows allows for services to shut down. To do this, increase the value stored in the registry key HKEY_LOCAL_MACHINE \SYSTEM\CurrentControlSet\Control \WaitToKillServiceTimeout.

IQQG0164W **Windows Services is set to allow *Windows_Services_shutdown_time* milliseconds to shut down, but the server requires *estimated_shutdown_time* milliseconds.**

Explanation:
The time that Windows Services allows a program to shut down is too low.

User response:
Increase the amount of time Windows allows for services to shut down. To do this, increase the value stored in the registry key HKEY_LOCAL_MACHINE \SYSTEM\CurrentControlSet\Control \WaitToKillServiceTimeout.

IQQG0165E **The rename of collection *collection_name* failed.**

Explanation:
The collection was not able to be renamed due to an error.

User response:
See the other associated messages for more information.

IQQG0166E **Object *existing_name* could not be renamed to *new_name*.**

Explanation:
The object was not able to be renamed.

User response:
Ensure that no program is accessing the specified directory and that the file is not open.

IQQG0167E **The configuration key *configuration_key* for component**

***component_id* has a non-Long value of *value*.**

Explanation:
The configuration key was expected to be a Long, but it has a non-Long value.

User response:
Ensure that the configuration key has a valid Long value.

IQQG0169E **An attempt to update the *file_path* configuration file failed before the changes could be applied.**

Explanation:
To preserve the integrity of configuration files, the system writes updates to a temporary location before putting the changes into effect. A problem occurred before the configuration updates could be applied.

User response:
Ensure that no program is accessing the specified file and that the file is not open.

IQQG0170E **The collection *collection_name* could not be updated because it was updated after this copy of the configuration was retrieved.**

Explanation:
The collection cannot be updated because some other task updated the configuration after this version of the configuration was retrieved.

User response:
Retrieve the configuration and make the changes again.

IQQG0171E **The document pipeline extension class encountered an error when processing collection *collection_name*.**

Explanation:
The document pipeline extension routine encountered an error.

User response:
See the attached stack trace and correct the problem.

IQQG0172E **The system cannot create the file *file_name*.**

Explanation:
The system cannot create the specified file.

User response:
See the other associated error messages for more information. Ensure that the program has permission to create files, and that the disk is not full.

IQQG0177E **The locale string *locale_string* is not valid.**

Explanation:
The locale string is not valie.

User response:
Please provide a valid 5 character locale string.

IQQG0178W **The collection directory *directory_name* was removed because the data in it is incomplete.**

Explanation:
The collection directory was missing some key files, so it could not be used.

User response:
No action is necessary.

IQQG0179E **The constructor initialization process failed. Error details: *error_details*.**

Explanation:
The config/constructors.xml file is not configured correctly.

User response:
Verify that the config/constructors.xml file is configured correctly.

IQQG0180E **The constructor [*constructor_name*] initialization process failed because the constructor class cannot be loaded. Error details: *exception_details*.**

Explanation:
The config/constructors.xml file is not configured correctly.

User response:
Verify that the config/constructors.xml file is configured correctly and contains the correct constructor class name. Also, ensure that the plugins directory contains the correct constructor implementation JAR file.

IQQG0181E **The constructor *constructor_name* initialization process failed. Error details: *error_details*.**

Explanation:
The config/constructors.xml file is not configured correctly.

User response:
Verify that the config/constructors.xml file contains the correct constructor configuration parameters.

IQQG0182E **The class path for constructor *constructor_name* cannot be loaded properly because a required implementation file is missing from the plugins directory.**

Explanation:
A required implementation file is missing from the plugins directory.

User response:
Verify that the plugins directory contains the required implementation file.

IQQG0183E **The constructor configuration file [*constructors.xml file*] cannot be loaded or parsed. Error details: *error_details*.**

Explanation:
The constructors.xml file is missing from the config directory or is not formatted correctly.

User response:
Verify that the constructors.xml file exists in the config directory and is formatted correctly.

IQQG0184E **The runtime configuration cannot be set for the constructor [*constructor name*]. Error details: *error_details*.**

Explanation:
The constructors.xml file is missing from the config directory or is not formatted correctly.

User response:
Verify that the constructors.xml file exists in the config directory and is configured correctly.

IQQG0185E **The runtime configuration for the constructor [{0}] is missing.**

Explanation:
The runtime configuration for the constructor is not formatted correctly.

User response:
Ensure that the client is providing the correct constructor runtime configuration to the server.

IQQG0186E **The constructor cannot be initialized because the constructor name is not specified in the config/constructors.xml file.**

Explanation:
The constructor name is not specified in the config/constructors.xml file.

User response:
Specify the constructor name in the config/constructors.xml file.

IQQG0187E **The constructor cannot be initialized because the constructor class name is not specified in the config/constructors.xml configuration file.**

Explanation:
The constructor class name is not specified in the constructors.xml file.

User response:
Specify the constructor class name in the config/constructors.xml file.

IQQG0188E **The constructor [*constructor name*] is not initialized properly.**

Explanation:
The config/constructors.xml file is not configured correctly.

User response:
Verify that the config/constructors.xml file contains the correct constructor configuration. See the log file in the /log directory for more error details.

IQQG0189E **The constructor [*constructor name*] was not found.**

Explanation:
The constructor is not registered in the system.

User response:
Verify that the constructor name is specified in the config/constructors.xml file.

IQQG0190E **The binary recognizer mechanism cannot be initialized because the configuration file was not found or contains incorrect information.**

Explanation:
The binary recognizer configuration file was not found or contains incorrect information.

User response:
Verify that the binary_recognizer_config.xml file exists in the configdefaults directory and contains the correct information.

IQQG0191E **The binary textual recognizer mechanism cannot be initialized.**

Explanation:
The binary textual recognizer is missing or contains incorrect information.

User response:
Verify that the binary_recognizer_config.xml file exists in the configdefaults directory and contains the correct information.

IQQG0192E **The config/constructors.xml constructor configuration file cannot be validated against the config/constructors.xsd XML schema file. Error details: [*exception details*].**

Explanation:
The constructor configuration file cannot be validated against the XML schema file.

User response:
Validate the constructor configuration file (config/constructors.xml) by using the XML schema file (config/constructors.xsd).

IQQG0193W **Failed reading from the input stream for document ID: *document ID***

Explanation:
Error reading indexable document stream.

User response:
No action is required.

IQQG0194E **Constructor [*constructor name*] cannot populate the list of documents with collection ID [*collection is*]. Error details: *error details***

Explanation:
An error occurred during the constructor population process.

User response:
No action is required.

IQQG0195E **The document text filter service cannot be called in constructor [*constructor name*] for collection ID [*collection id*]. Error details: *error details*.**

Explanation:
An error occurred during the invocation of the document text filter service.

User response:
No action is required.

IQQG0199E **The list of documents with collection ID [*colleation id*] cannot be populated. Error details: *error details*.**

Explanation:
An error occurred during the constructor population process.

User response:
No action is required.

IQQG0200E **The supported client list for the server cannot be obtained. Error details: *message description*.**

Explanation:
The server cannot read its list of supported clients.

User response:
Ensure that the server implementation JAR file contains a list of supported client versions.

IQQG0201E **The client version cannot be determined. Error details: *message description*.**

Explanation:
The client cannot determine its version.

User response:
Ensure that the client implementation JAR file contains the client version.

IQQG0202E **The list of supported server versions cannot be obtained by the client. Error details: *message description*.**

Explanation:
The client cannot determine the list of supported server versions.

User response:
Ensure that the client implementation JAR file contains a list of supported server versions.

IQQG0203E **The document encoding cannot be set because the content type of document {0} is not textual.**

Explanation:
Document encoding can only be set when the content type of a document is textual.

User response:
No action is required.

IQQG0204E **The MIME type of the document {0} cannot be set because its content type is not binary.**

Explanation:
The MIME type can only be set when the content type of a document is binary.

User response:
No action is required.

IQQG0205E **The client [version: *client version*] is not compatible with the server [version: *server version*]. The client supports server versions [*client supported server versions*]. The server supports client versions [*server supported client versions*].**

Explanation:
The client and the server are not compatible.

User response:
Ensure that you use compatible versions of the client and server.

IQQG0206E **The preprocessing server [version: *Preprocessing server version*] and indexing server [version: *Indexing server version*] are not compatible**

because the version numbers are different.

Explanation:
The preprocessing server version and indexing server version must match.

User response:
Ensure that the preprocessing server version and indexing server version are the same.

IQQG0207E **The preprocessing server and indexing server are not compatible because the version numbers are different.**

Explanation:
The preprocessing server version and indexing server version must match.

User response:
Ensure that the preprocessing server version and indexing server version are the same.

IQQG0208E **The search cannot be performed because the result limit is negative.**

Explanation:
The result limit parameter on a query request cannot be negative.

User response:
Specify a positive value for the result limit parameter.

IQQG0209E **The search cannot be performed because the page number should be greated than 0.**

Explanation:
The page number parameter on a query request should be greated than 0.

User response:
Specify a positive value for the page number parameter.

IQQG0210E **The addOrReplaceDocuments or removeDocuments method cannot be called because the subscribeToMessages method was not called as required.**

Explanation:
The subscribeToMessages method must be called before you call the addOrReplaceDocuments or removeDocuments method.

User response:
Call the subscribeToMessages method before you call the addOrReplaceDocuments or removeDocuments method.

IQQG0211E **Method invocation failed for method {0}.**

Explanation:
Method invocation failed for method triggered by the subscribeToMessages method.

User response:
Verify that the method arguments and implementation are correct.

IQQG0212E **The CollectionConfiguration object is null.**

Explanation:
The CollectionConfiguration object cannot be null when calling the deleteCollection or createCollection methods.

User response:
Pass a valid CollectionConfiguration object to the deleteCollection or createCollection method.

IQQG0213W **No data was set to be indexed.**

Explanation:
No data was set to be indexed.

User response:
Set data to be indexed when preparing the IndexableDocument class for the addOrReplaceDocuments method.

IQQG0214E **The requested number of results per page exceeded the maximum number of sorted search results [*maximum number of sorted results*].**

Explanation:
The requested number of results per page exceeded the maximum number of sorted search results [*maximum number of sorted results*].

User response:
Request fewer results per page.

IQQG0215E **The specified synonym XML file was not found in the /config/collections/collection_name/data/synonym directory.**

Explanation:
The specified synonym XML file was not found in the /config/collections/collection_name/data/synonym directory when the updateSynonyms method was called.

User response:
Specify a valid synonym XML file and ensure that the file is located in the /config/collections/collection_name/data/synonym directory.

IQQG0216E **The server object was not initialized before a method call.**

Explanation:

The server object must be initialized before the specified method can be called.

User response:
Initialize the server object before the method is called.

IQQG0217E **The methodHolder or methodName object is null.**

Explanation:
The methodHolder and methodName parameters for the subscribeToMessages method cannot be null.

User response:
Initialize the methodHolder and methodName objects.

IQQG0301E **The server version cannot be obtained. Error details: *message description*.**

Explanation:
The server cannot read its version.

User response:
Ensure that the server implementation JAR file contains an implementation version.

IQQG0305I **A new dictionary file was not generated because the existing dictionary file cannot be backed up.**

Explanation:
The stop words tool will not build a new dictionary because it could not back up the old one.

User response:
No action is required.

IQQG0307W **The *Name of the file that does not exist* file does not exist.**

Explanation:
An XML stop words file does not exist for the current language.

User response:
If you need to define stop words in this language, create a stop words XML file.

IQQG0308W **The file *Name of the file that does not exist* does not exist.**

Explanation:
A stop words dictionary file does not exist for the current language.

User response:
No action is required.

IQQG0309I

Explanation:
No stop words dictionary file was found for the current language. A new dictionary will be compiled based on the words in the existing stop words XML file.

User response:
No action is required.

IQQG0310I **There is no stop words file for this language.**

Explanation:
No stop words XML file was found for a language.

User response:
If you need to define stop words in this language, create a stop words XML file.

IQQG0312I **Unable to compile a new stop words dictionary.**

Explanation:
The system cannot compile a new stop words dictionary.

User response:
No action is required.

IQQG0315I **The *Name of the file that was backed up* file was backed up to *Name of the file that it was backed up to*.**

Explanation:
The existing stop words dictionary file was successfully backed up before the new dictionary was built.

User response:
No action is required.

IQQG0316W **The stop words dictionary file *Name of the file that could not be backed up* cannot be backed up.**

Explanation:
The stop words dictionary file was not successfully backed up.

User response:
No action is required.

IQQG0319E

Explanation:
The specified working directory does not exist or the required XSD file is missing.

User response:
Make sure that the working directory is correct and contains the required XSD file.

IQQG0320E

Explanation:
The DictionaryBuilder failed to create the dictionary file.

User response:
Make sure that the stop word XML file is correctly formatted and that the user who is running the stop

word tool has write permissions on the working directory.

IQQG0322E

Explanation:
The DictionaryBuilder could not create the dictionary file and was unable to restore the backed-up file.

User response:
Rename the backed-up dictionary file by removing the numeric suffix from the file name. For example, if the backed up file name is en-Stw4.dic, rename it to en-Stw.dic.

IQQG0323W Can not find bundle for base name *bundle name* and locale *bundle locale*.

Explanation:
The server cannot load resource bundle accoding to the provided name and locale

User response:
Ensure that the resource bundle with the specified name and locale indeed exists under HOME_DIR \resourcebundles location.

IQQG0324W The input log file *file name* is in a readable format. No formatting is required.

Explanation:
The input log file is in a new XML readable format. No formatting is required.

User response:
No action is required.

IQQG0325E There is no response from transfomation service.

Explanation:
There is no response from transfomation service.

User response:
No action is required.

IQQG0325W Unable to format the *file name* file.

Explanation:
Unable to format the input log file.

User response:
Set a correct format input file.

IQQG0326E This archive file {0} either corrupt or not supported by default system parser.

Explanation:
Unsupported archive format.

User response:
No action is required.

IQQG0327E The document text filter service cannot be called for document ID [*error details*]. Error details: {2}.

Explanation:
An error occurred during the invocation of the document text filter service.

User response:
No action is required.

IQQG0328E The server cannot start because the keyStoreName parameter is missing from the configuration file.

Explanation:
The server is configured to run in secure mode but the keystore name parameter is missing from the configuration file.

User response:
Run the configuration tool and set a value for the keyStoreName parameter. Then restart the server.

IQQG0329E The server cannot start because the keyStorePassword parameter is missing from the configuration file.

Explanation:
The server is configured to run in secure mode but the keystore password parameter is missing from the configuration file.

User response:
Run the configuration tool and set a value for the keyStorePassword parameter. Then restart the server.

IQQG0330E The server cannot start because the secure and nonsecure ports are both set to 0.

Explanation:
A valid port number must be specified for the secure port, nonsecure port, or both ports.

User response:
Run the configuration tool and set the adminHTTPPort parameter, securePort parameter, or both parameters to a valid port number (other than 0). Then restart the server.

IQQG0331E The server cannot start because the secure and nonsecure ports are set to the same value {0}.

Explanation:
The secure and nonsecure ports cannot be set to the same value.

User response:

Run the configuration tool and set the adminHTTPPort or securePort parameter to another port number. Then restart the server.

IQQG0332E **The server cannot start because the keystore password that is stored in the configuration file cannot be decrypted. Details: *the error received during password decryption***

Explanation:
The keystore password that is stored in the configuration file cannot be decrypted.

User response:
Run the configuration tool and set the keystore password. Then restart the server.

IQQG0335E **The server cannot start. See the error log file for more details.**

Explanation:
The server cannot start.

User response:
See the log file for more details. By default, the error log is located in the ECMTS_Home/log directory.

IQQG0336I **The server started successfully. The server is listening on port *Non secure port* and on secure port *Secure port*.**

Explanation:
null

User response:
null

IQQG0337I **The server started successfully. The server is listening on port *port*.**

Explanation:
null

User response:
null

IQQG0338I **The server started successfully. The server is listening on secure port *Secured port*.**

Explanation:
null

User response:
null

IQQG0340E **Directories cannot be created in the *directory_name* directory because access is denied.**

Explanation:
The file system cannot create directories in the specified directory because access is denied.

User response:
Verify that the specified directory has the required access permissions.

IQQG0341E **A secure request was sent through a nonsecure channel.**

Explanation:
The client sent a secure request through a nonsecure channel.

User response:
Verify that the client sends secure requests through a secure channel, and then resend the request to the server.

IQQG0343E **The keystore password cannot be encrypted. Details: *failure explanation***

Explanation:
The keystore password cannot be encrypted.

User response:
Retry to encrypt the password after you fix the problem that is mentioned in the error details.

IQQG0344E **Document *document id* in collection *collection id* could not be added to the output queue.**

Explanation:
Document *document id* in collection *collection id* could not be added to the output queue.

User response:
Resend this document for indexing.

IQQG0345E **Constructor *Constructor name* unable to populate document content with document ID *Document ID* and collection *collection ID* since the document content is not resetable.**

Explanation:
Wraping populated document input stream by markable and resetable fails

User response:
Verify that populated data in the constructor is resetable and markable or can be converted to resetable or markable

IQQG0346E **The specified value of the numberOfTokenizers parameter (*numberOfTokenizers*) is less than the specified value of the numberOfPreprocessingThreads parameter (*numberOfPreprocessingThreads*).**

Explanation:
If there are fewer tokenizers than preprocessing threads, some preprocessing threads need to wait for

a tokenizer to become available. This wait time can increase the total processing time.

User response:
Run the configuration tool and set a higher value for the numberOfTokenizers parameter or set a lower value for the numberOfPreprocessingThreads parameter.

IQQI0008E **The document *document_URL* cannot be inserted.**

Explanation:
The document was waiting to be inserted in the document queue. However, the process was interrupted by a system event.

User response:
Resubmit the document.

IQQI0011E **Documents cannot be processed because the system is shutting down.**

Explanation:
No documents can be processed because the system is shutting down.

User response:
Retry the operation after the system is restarted.

IQQI0013E **The index files for collection *collection_name* cannot be closed.**

Explanation:
The index files cannot be closed.

User response:
See the other associated messages for more information.

IQQI0014E **The documents cannot be processed for collection *collection_name*.**

Explanation:
The documents cannot be processed for the collection.

User response:
See the other associated messages for more information.

IQQI0016W **The default value for *attribute_value* in file *file_name* for collection *collection_name* cannot be used.**

Explanation:
The specified attribute value setting cannot be used.

User response:
Update the setting to a valid value.

IQQI0018I **The system cannot retrieve statistical information for collection *collection_name*.**

Explanation:
The system encountered an error while trying to retrieve statistical information for the collection.

User response:
No action is required because failure to gather these statistics does not affect normal operations.

IQQI0019E **The system encountered a problem when indexing documents.**

Explanation:
The documents cannot be indexed.

User response:
Check the system log for any severe errors.

IQQI0020E **The attempt to add document category *category_name* failed. Reason: *reason*.**

Explanation:
The category could not be added because an I/O error occurred.

User response:
Check to see whether the disk is full and check the system log for any severe errors.

IQQI0021E **The retrieval of multi-faceted category ordinal *ordinal_hash* failed. Reason: *reason*.**

Explanation:
Internal multi-faceted taxonomy information could not be accessed.

User response:
Check the system log for any severe errors.

IQQI0022E **The loading of multi-faceted category information from location *path_to_index* failed. Reason: *reason*.**

Explanation:
Internal multi-faceted taxonomy information could not be loaded.

User response:
Check the system log for any severe errors.

IQQI0024E **The taxonomy index is missing one of its essential files: *path_and_filename*.**

Explanation:
A critical taxonomy file, *path_and_filename*, is missing.

User response:

Verify that the path is correct and that the index has been built. Check the system log for any severe errors.

IQQI0025E **The taxonomy index file *file* metadata is corrupt. (*read_metadata* was found, which should be *valid_metadata*.)**

Explanation:
The index file *file* was not found.

User response:
Verify that the path is correct and that the index has been built. Check the system log for any severe errors.

IQQI0026E **Taxonomy index directory *index_directory* is missing.**

Explanation:
The taxonomy index directory *index_directory* was not found.

User response:
Verify that the path is correct and that the index has been built. Check the system log for any severe errors.

IQQI0027E **An internal taxonomy index error occurred: *error*.**

Explanation:
An unexpected error occurred.

User response:
Contact IBM Software Support.

IQQI0028E **The taxonomy index could not open one of its essential files: *path_and_filename*.**

Explanation:
A critical taxonomy file is missing.

User response:
Verify that the path for this file is correct and that the index has been built at least one time. Check the system log for any severe errors.

IQQI0030E **The taxonomy index failed to copy to the new generation directory *gen_path*.**

Explanation:
The taxonomy index was not copied to the new generation directory.

User response:
Check the system log for any severe errors. On Windows, check that there are no sharing violations such as open files in the new directory.

IQQI0031E **The taxonomy index copy function is missing one of its essential files: *path_and_filename*.**

Explanation:
A critical taxonomy file, *path_and_filename*, was missing during the copy phase to a new generation directory.

User response:
Verify that the path is correct and present on your file system. Check the system log for any severe errors.

IQQI0032E **An I/O error occurred when adding category *category_label* to the taxonomy index. Reason: *I/O Exception string*.**

Explanation:
When adding a new category to the taxonomy index, an I/O error was encountered: *I/O Exception string*.

User response:
Check the system log for any severe errors.

IQQI0033E **An I/O error occurred when closing the taxonomy index. Reason: *I/O Exception string*.**

Explanation:
When closing the taxonomy index, an I/O error occurred. This problem might occur if disk or hardware problems exist, or if the disk has become full.

User response:
Check the file system for disk errors or disk full conditions. Check the system log for any severe errors.

IQQI0049E **Cannot release a lock for collection *Collection name***

Explanation:
This is a pure internal error. If this error occurs it indicated a severe problem with the program logic

User response:
No action is required since the product already failed.

IQQI0050E **Cannot obtain a lock for collection *Collection name***

Explanation:
This is a pure internal error. If this error occurs it indicated a severe problem with the program logic

User response:
No action is required since the product already failed.

IQQI0051I **The optimization command to collection {0} was send to server successfully.**

Explanation:
The optimization of index succeeded.

User response:
null

IQQI0052E **The index [{0}] is busy, currently indexing documents.**

Explanation:

It is not allowed to index or to optimize an index calling the optimize method while indexing is running.

User response:
Call again to optimize after indexing tasks are finished .

IQQI0053I **The optimization to collection {0} terminated successfully.**

Explanation:
The optimization terminated successfuly.

User response:
null

IQQI0054I **The optimization on collection {0} is still processing.**

Explanation:
The optimization is still processing.

User response:
null

IQQI0055E **The optimization on collection {0} terminated with errors.**

Explanation:
The optimization terminated with an error.

User response:
See log file for error explenation.

IQQI0056E **There is an optimization process currently running on collection {0}.**

Explanation:
There is a previous optimization task that did not terminate yet.

User response:
Run the optimazation againg after previous optimazation terminates.

IQQI0057W **No details available about an optimization process on collection {0} (last optimization was done on collection {1}).**

Explanation:
There is no previous optimization details for this collection.

User response:
null

IQQI0058I **Document with id *document id* was indexed successfully**

Explanation:
This message is returned to the client as document indexing process terminator acknowledge

User response:
No action is required

IQQI0059W **No details available about an optimization process on collection {0} .**

Explanation:
There is no previous optimization details for this collection.this is same as I0057W except no details about last optimization

User response:
null

IQQI0061E **The document with ID [*document id*] cannot be indexed due to inadequate disk space or permissions.Verify that there is sufficient disk space and permissions to access the disk. Then run the indexing process again.**

Explanation:
The document cannot be indexed because there is inadequate disk space or permissions.

User response:
Verify that the amount of free disk space is equal to or greater than the total size of the documents to be indexed, and that you have full access privileges to the documents.

IQQI0062E **Occurred error during execution task [*Task id*]. The indexed data might have been lost. Verify that there is sufficient disk space and permissions to access the disk. Then run the indexing process again.**

Explanation:
Occurred task error because there is inadequate disk space or permissions.

User response:
Verify that the amount of free disk space is equal to or greater than the total size of the documents to be indexed, and that you have full access privileges to the documents.

IQQI0063E **Task [*Task id*] did not complete successfully. Collection [{1}] could not be committed. The indexed data might have been lost. Verify that there is sufficient disk space and permissions to access the disk. Then run the indexing process again.**

Explanation:
Unable to commit because there is inadequate disk space or permissions.

User response:

Verify that the amount of free disk space is equal to or greater than the total size of the documents to be indexed, and that you have full access privileges to the documents.

IQQM0010E **The upgrade program was unable to migrate file *file_name*.**

Explanation:
The upgrade process was unable to migrate a file. The upgrade program will continue with the next task.

User response:
See the migration.0.log file in the INSTALL_ROOT/log directory and correct any errors. You can then rerun the upgrade program, which is migrate.bat or migrate.sh in the INSTALL_ROOT/bin directory.

IQQM0016E **The directory *directory_name* could not be created.**

Explanation:
The upgrade program was not able to create the directory.

User response:
Manually create the directory. You can then rerun the upgrade program, which is migrate.bat or migrate.sh in the INSTALL_ROOT/bin directory.

IQQP0001E **The parser component executable file that is specified by *parser_component* cannot be found. The parser component will not be available.**

Explanation:
The system cannot find the parser component.

User response:
Check the path in the error message to verify that the directory with the parser component executable file exists. Check the permissions of the file and directory and ensure that the file is accessible by the ID that runs the program. Check the specification section for the parser component in the global configuration file for a path override.

IQQP0002E **The parser component cannot be started. The parser component will not be available.**

Explanation:
The parser component cannot be started.

User response:
See the system log for the errors about starting the parser component.

IQQP0006E **The parser cannot read the parser configuration file.**

Explanation:
The configuration file for the parser cannot be read.

User response:
Verify that the configuration file for the parser exists. Check the permissions of the file and directory and ensure that the file is accessible. See the other associated messages for more information.

IQQP0008W **Unable to index document *document_ID* because the binary filter was not found or the MIME type [{1}] is unsupported.**

Explanation:
The document is not a supported document type. This document will not be indexed.

User response:
No action is required. See the help for information about supported document types.

IQQP0009W **The parser cannot parse the document *document_ID*. The document will not be indexed.**

Explanation:
The document cannot be parsed because it is corrupted or malformed. The parser tried to correct the problem but failed. This document will not be indexed.

User response:
If you want to index the document, ensure that it is not corrupted or malformed. Then, crawl it again.

IQQP0010W **The parser encountered an incorrect byte sequence during character conversion for the document *document_ID*. The document will not be indexed.**

Explanation:
A character conversion error occurred when parsing the document. The document might contain an incorrect byte sequence with the specified character set. This document will not be indexed.

User response:
Verify that the document does not contain an incorrect byte sequence with the specified character set. For example, if a document is encoded with SJIS, but it declares UTF-8 in the document content, the document will have a byte sequence that cannot be decoded.

IQQP0012W **The document with ID *document_ID* exceeds the specified size limit for text documents. The current size limit is {1} characters. The indexed document will be truncated.**

Explanation:
The length of the extracted characters from the document exceeds the limit of the length of the

document in text format. This document will be indexed, but it will be truncated.

User response:
No action is required.

IQQP0013E **The document {0} cannot be indexed because it is larger than the configured size limit.**

Explanation:
The document is larger than the configured size limit. This document will not be indexed.

User response:
You can increase the size limit by modifying the text size parameters in the configcollections*collection_name*parser_config.xml file.

IQQP0014W

Explanation:
For the specified setting, either no value was set or the value was invalid. The default value will be used for this setting.

User response:
Verify the settings in the parser_config.xml file for the current collection.

IQQP0015W

Explanation:
For the specified setting, the value is set to 0. No documents of this type will be indexed.

User response:
If you want to index documents of this type, update the setting in the parser_config.xml file for the current collection.

IQQP0016W

Explanation:
The value of the specified setting is larger than the value of the max.text.size setting. If the extracted text of a document is larger than the value of the max.text.size setting, the document will be truncated before it is indexed. This may lead to unexpected results when searching.

User response:
To ensure that the entire document is indexed, increase the value of the max.text.size setting or decrease the value of the specified setting that is larger than the value of the max.text.size setting in the parser_config.xml file for the current collection.

IQQP0026E **Unable to parse document *document_ID* because the document encoding could not be recognized either the document contents are too short or contains mixed encodings.**

Explanation:
The document encoding could not be recognized or unsupported.Either the document contents are too short or contains mixedencodings.#

User response:
Validate that the document has size at least 2k or provide document encoding explicitly.

IQQP0027W **The document with id [*document_ID*] does not contain any content that can be indexed.**

Explanation:
The document does not contain content that can be indexed.

User response:
No action is required.

IQQP0028E **The XML document *exception details* cannot be parsed because its format is invalid. Error details: {1}**

Explanation:
The document cannot be parsed because it is corrupted or malformed. The parser was unable to correct the problem. This document will not be indexed.

User response:
If you want to index the document, ensure that it is not corrupted or malformed. Then, index it again.

IQQP0029E **The binary document *exception details* cannot be parsed because its format is invalid. Error details: {1}**

Explanation:
The document cannot be parsed because it is corrupted or malformed. The parser was unable to correct the problem. This document will not be indexed.

User response:
If you want to index the document, ensure that it is not corrupted or malformed. Then, index it again.

IQQP0030E **Part of the query [*field*:*search term*] cannot be processed. Colons (:) must be escaped by using double back slashes.**

Explanation:
The provided xml query contains not escaped colon charcter

User response:
Escape colon character by using double back slashes

IQQP0031E **None of the files in the *file name* archive file can be parsed. The**

archive file might be password protected or corrupted.

Explanation:
None of the files in the archive file cannot be parsed.

User response:
No action is required.

IQQP0032E **The parser [*Parser name*] cannot parse the document with id [*document id*] because the document is corrupted.**

Explanation:
The document cannot be parsed because it is corrupted.

User response:
Provide a version of the document that is not corrupted.

IQQP0033W **The fallback parser [*fallbakc parser name*] was activated. Because the document with id [*document id*] was parsed by the fallback parser, the document might not be indexed properly.**

Explanation:
The fallback parser was activated because the document could not be parsed by the primary parser.

User response:
Verify that the fallback parser processed the document correctly.

IQQP2600E **The parser cannot open a session for the parser component.**

Explanation:
The parser cannot open a session for the parser component because the parser is not enabled for binary documents.

User response:
Enable the parser component for binary documents. To do this, set the StellentEnabled parameter to true in the INSTALL_ROOT/config/releaseinfo/release_XXX.properties file (where XXX represents the edition of OmniFind installed in your system).

IQQP2601E **The parser component cannot be started. The parser component will not be available.**

Explanation:
The parser component cannot be started.

User response:
See the system log for errors about starting the parser component.

IQQP2602E **The parser component cannot be stopped successfully.**

Explanation:
The parser component cannot be stopped successfully.

User response:
See the system log for errors about stopping the parser component.

IQQP5001E **The collection with the name *collection_name* does not exist.**

Explanation:
No configuration for the collection was found. The collection does not exist.

User response:
You can use the administration console to see existing collection names.

IQQP5002E **The system cannot retrieve a tokenizer for collection *collection_name*.**

Explanation:
No free tokenizers are available for the collection, and a timeout error occurred when requesting the tokenizer.

User response:
Change the setting for the timeout value or the number of tokenizers for the collection.

IQQP5003I **The document does not contain stored tokens.**

Explanation:
The specified document does not contain stored tokens. No field map exists to be restored. This is probably because the document is an error document.

User response:
No action is required.

IQQP6000E **The tokenizer cannot be initialized with the UIMA descriptor *UIMA_descriptor* and data directory *UIMA_datapath*.**

Explanation:
The tokenizer cannot initialize the common analysis structure (CAS) tokenizer with the specified UIMA descriptor and data directory.

User response:
Verify that the specified UIMA descriptor file and data directory are correct. Also, ensure that the specified UIMA descriptor is well formed.

IQQP6001E **The tokenizer cannot process the common analysis structure (CAS).**

Explanation:
The document text cannot be tokenized and it cannot be indexed.

User response:

Check the system log for errors about processing the common analysis structure (CAS).

IQQP9000E **The normalizer cannot be initialized with the specified resource *resource_path*.**

Explanation:
The normalizer cannot be initialized. The indexed text will not be normalized.

User response:
Verify that the specified resource for the normalizer is correct. See the other associated messages for more information.

IQQP9001E **The query *query string* cannot be processed because the query contains incorrect syntax. A wildcard character(*) cannot be used in the attribute name.**

Explanation:
XML search query syntax does not support wildcard characters in the name of an element attribute.

User response:
Resubmit the query without wildcard characters.

IQQP9002E **The specified wildcard query cannot be processed because the query exceeded the maximum number of permitted BooleanQuery clauses.**

Explanation:
The specified wildcard query cannot be processed because it contains too many Boolean clauses.

User response:
Provide a more specific query or increase the maximum number of Boolean clauses by setting the queryExpansionLimit parameter in the HOME_DIR/config/config.xml file.

IQQP9003E **The query [*query string*] cannot be parsed. Details: *error description*.**

Explanation:
The specified query cannot be parsed.

User response:
Validate that the syntax of the specified query is correct.

IQQP9004E **The query [*query string*] cannot be parsed because there are too many Boolean clauses.**

Explanation:
The specified query cannot be processed because it contains too many Boolean clauses.

User response:
Provide a more specific query or increase the maximum number of Boolean clauses by setting the queryExpansionLimit parameter in the HOME_DIR/config/config.xml file.

IQQP9005E **Missing return statement in function**

Explanation:
Missing return statement in function

User response:
no action

IQQP9006E **Conversion of XML Search query term [*query string*] failed.**

Explanation:
Provided query cannot be executed becuase it contains tot many boolean clauses

User response:
Validate provided XML query has correct syntax.

IQQP9008E **A wildcard character cannot be specified as the first character in a query.**

Explanation:
A wildcard character cannot be specified as the first character in a query.

User response:
Modify the query so that the first character is not a wildcard character.

IQQP9012E **The term [*term*] cannot end with an escape character.**

Explanation:
A term cannot end with an escape character.

User response:
Modify the term by removing the final escape character or by adding another character at the end of the term.

IQQP9013E **The Unicode escape sequence [*query string*] contains a non-hexadecimal character.**

Explanation:
Unicode escape sequences can contain only hexadecimal characters.

User response:
Remove the non-hexadecimal character from the Unicode escape sequence.

IQQP9014E **The fuzzy argument value [*query string*] is not valid because its data type is not float or double.**

Explanation:
Fuzzy argument values must be either a float or double type value.

User response:
Ensure that the fuzzy argument value is a float or double value.

IQQP9015E **The similarity value for a fuzzy query must be between 0 and 1.**

Explanation:
The similarity value for a fuzzy query must be between 0 and 1. A value closer to 1 matches terms with a higher similarity.

User response:
Specify a similarity value between 0 and 1.

IQQP9016E **The proximity value must be 1 or greater, and cannot exceed the value of the Integer.MAX_VALUE Java variable.**

Explanation:
A proximity value must be 1 or greater, and cannot exceed the value of the Integer.MAX_VALUE Java variable.

User response:
Specify a proximity value from 1 to the value of the Integer.MAX_VALUE Java variable.

IQQP9017E **The query [*query string*] cannot be parsed. Details: *error description*.**

Explanation:
The specified query cannot be parsed.

User response:
Validate that the syntax of the specified query is correct.

IQQP9018E **The query [*query string*] cannot be tokenized.**

Explanation:
The query cannot be tokenized.

User response:
Validate that the syntax of the specified query is correct.

IQQP9019E **The query [*query string*] cannot be processed.**

Explanation:
The query [*query string*] cannot be processed.

User response:
Validate that the syntax of the specified query is correct.

IQQP9020W **The version of the document parser (*actual transformation service version*) is newer than the expected parser version (*specified transformation service version*).**

Explanation:
null

User response:
null

IQQP9021E **The version of the document parser (*actual transformation service version*) is earlier than the expected parser version (*specified transformation service version*).**

Explanation:
The version of the document parser (*actual transformation service version*) is earlier than the expected parser version (*specified transformation service version*).

User response:
Install a supported version of the document parser.

IQQP9023W **The document parser was not loaded because its version cannot be determined.**

Explanation:
The document parser was not loaded because its version cannot be determined.

User response:
Install a supported version of the document parser.

IQQP9024E **Transformation service failed to extract text. Return code: *text extraction return code*, error description: [*text extraction error description*].**

Explanation:
Transformation service encountered an error during text extraction.

User response:
No action is required. The document can not be extracted by the transformation service.

IQQR0003E **The collection service for adding and removing documents and updating indexes cannot be found.**

Explanation:
The collection service for adding and removing documents and updating indexes cannot be found.

User response:
See the other associated messages for more information. If you cannot determine the cause of the problem, contact IBM Software Support.

IQQR0004E **The document with ID *document_ID* cannot be added to the *collection_name* collection.**

Explanation:
The document with ID *document_ID* cannot be added to the collection.

User response:
Verify that the document content is valid. If the document is XML, ensure that the XML is well formed. After you correct the problem, resubmit the document by using the API that was used before. If the problem reoccurs, check the system log files through the administration console for more information about errors that are related to adding documents.

IQQR0007E **The *collection_name* collection cannot be created.**

Explanation:
The *collection_name* collection cannot be created.

User response:
Ensure that the index parameter value that is specified in the create index request is valid. The value cannot contain any of the following characters: " * : ? < > |. If the value does contain any of these characters, modify the value so that is does not contain any of the characters and then resubmit the request. If the value is valid, check the system log files through the administration console for errors about creating the collection.

IQQR0008E **The *collection_name* collection cannot be deleted.**

Explanation:
The collection cannot be deleted.

User response
See the other associated messages for more information. Also, check the system log files through the administration console. If you cannot determine the cause of the problem, follow these steps:

1. Stop the search system.
2. Manually delete the index directory in the INSTALL_ROOT/package/config/collections directory.
3. Restart the search system.

IQQR0012E **The index parameter value *collection_name* in the request does not exist in the system.**

Explanation:
The request specified a collection name that does not exist in the search system.

User response:
Ensure that the collection name in the request is spelled correctly.

IQQR0023E **The search response for the query "*query*" on collection *collection_name* could not be created.**

Explanation:
The query was processed, but the server could not return the results.

User response:
Try sending the request again.

IQQR0036E **The *collection_name* collection successfully deleted.**

Explanation:
The collection successfully deleted.

User response
See the other associated messages for more information. Also, check the system log files through the administration console. If you cannot determine the cause of the problem, follow these steps:

1. Stop the search system.
2. Manually delete the index directory in the INSTALL_ROOT/package/config/collections directory.
3. Restart the search system.

IQQR0040E **Unable to obtain system trace information.**

Explanation:
The system trace information cannot be obtained.

User response:
Verify that the server is in consistent state and retry the operation.

IQQR0041E **Unable to configure system trace.**

Explanation:
Unable to configure the system trace.

User response:
Please check if the server is in consistent state and retry the operation.

IQQR0052E **Documents from the indexing server cannot be accessed.**

Explanation:
Documents cannot be accessed from the indexing server if the server is not in a consistent state.

User response:
Verify that the server is in a consistent state and retry the operation.

IQQR0053E **Preprocessed documents from collection {0} cannot be written to the indexing server.**

Explanation:
Preprocessed documents cannot be written to the indexing server if the server is not in a consistent state.

User response:
Verify that the server is in a consistent state and retry the operation.

IQQR0054E The indexing server is not running.

Explanation:
The indexing server is not running

User response:
Start the indexing server.

IQQR0055E The indexing server on host {0} port {1} cannot be accessed.

Explanation:
The indexing server cannot be accessed.

User response:
Verify that the specified host, port number, and authentication token are correct.

IQQR0056E The configuration of collection {0} cannot be retrieved.

Explanation:
The collection configuration cannot be retrieved.

User response:
Verify that the specified host, port number, and authentication token are correct.

IQQS0004E The reader for index *index_path* cannot be created.

Explanation:
The reader for the index cannot be created. Either the index is not accessible, or there is an error in the index configuration.

User response:
Ensure that the index is readable and that it exists in the specified directory.

IQQS0013E The spelling checker named "*name*" cannot be created.

Explanation:
The spelling checker cannot be created.

User response:
Ensure that the dictionary with the specified name exists. If it does not exist, you will not have spelling suggestions for that name.

IQQS0074E The *collection_name* collection is not enabled for search.

Explanation:
An attempt was made to obtain a searchable object for the collection, but the collection is not enabled for search.

User response:
No action is required. You can try to search again after the search administrator enables the collection for search.

IQQS0077E The query cannot be processed because it contains no valid search terms.

Explanation:
The query cannot be processed because it contained no valid search terms. This might occur if the query contained only stop words or punctuation that was ignored by the search engine. (A stop word is a word that is commonly used, such as the, an, or and.)

User response:
Rewrite the query so that it contains at least one valid search term.

IQQW0001I Service *service_name* was installed successfully.

Explanation:
The application was successfully installed as a Windows service.

User response:
No action is required.

IQQW0002E The installation of service *service_name* failed. Verify that you have authority to create Windows services and that this application is not already installed as a service.

Explanation:
The application was unable to install a Windows service.

User response:
Ensure that you have administrator authority and are able to create services. Also verify that the executable file is not already installed as a service.

IQQW0009E The service could not be removed. Verify that the service is installed and ensure that you have authority to delete services.

Explanation:
A request to remove the Windows service failed.

User response:
Check that the service is installed and verify that the user account has administrator privileges.

IQQW0011E The service could not be installed. A service with name {0} already exists.

Explanation:
A request to install the Windows service failed since a service with the given name already exists.

User response:

Check that the service is not already installed - If it is installed you should first uninstall it. If it is not already installed specify another name.

IQQW0012E **The service could not be installed. A service running this process already exists.**

Explanation:
A request to install the Windows service failed since a service That run this process already exists.

User response:
Check that the service is not already installed - If it is installed (even with another name) you should first uninstall it.

IQQW0013E **The service could not be installed. A service already exists in the configuration file.**

Explanation:
A request to install the Windows service failed since a service already exists in the configuration file.

User response:
Check that the service is not already installed - If it is installed you should first uninstall it.

IQQW0015E **The request failed since the configuration path {0} does not exists or not sufficient permissions to access it.**

Explanation:
The request failed since the given configuration path is incorrect or not sufficient permissions to access it.

User response:
Check that the given configuration path is correct and that there are sufficient permissions to access it.

IQQW0016E **The request failed since the service name does not exists in the configuration file.**

Explanation:
The request failed since the service name does not exists in the configuration file.

User response:
Check that the service is installed.

IQQX0033E **Comparisons can be applied only to attributes, not to tags.**

Explanation:
An XML search query term compared a tag element to a value. Only attributes can be compared to values.

User response:
Use the contains or excludes operator instead.

IQQX0034E **There is an error in the query string at position *error_position*.**

Explanation:
An XML search query term does not conform to the supported grammar. The error is at the indicated character.

User response:
Rewrite the query.

IQQX0035E **There is an error in the query string at position *error_position*. Token type *expected_token_type* was expected; type *observed_token_type* was found.**

Explanation:
An XML search query term does not conform to the supported grammar. The error is at the indicated character.

User response:
Rewrite the query.

IQQX0036E **The query contains a number written in an unsupported format.**

Explanation:
An XML search query term contains a number whose format is not recognized by the parser.

User response:
Rewrite the query.

IQQX0037E **Free text in the XML search query term could not be parsed.**

Explanation:
An XML search query term contains free text that could not be parsed.

User response:
Rewrite the query.

IQQX0038E **An XML search query term could not be parsed because of an unexpected *unsupported_element*.**

Explanation:
An XML search query term could not be parsed because of an unsupported term or an exception.

User response:
Rewrite the query.

IQQX0039W **Collection *collection_name* cannot be backed up or restored because backup and restore are disabled.**

Explanation:
Backup and restore have been disabled, possibly because the system is shutting down.

User response:
No action is required.

IQQX0040W **Collection *collection_name* cannot be backed up or restored because**

there is already a backup or restore in progress.

Explanation:
A backup or restore was requested for the specified collection, but there is already a backup or restore in progress, and only one such operation at a time is permitted.

User response:
Wait until the backup or restore in progress has finished, then try again.

IQQX0041W **An error occurred when shutting down the backup or restore that was in progress on collection *collection_name*.**

Explanation:
An exception occurred when stopping the backup or restore on the indicated collection.

User response:
See the other associated messages for more information.

IQQX0042W **The background task cannot begin *new_state* because it is currently *current_state*.**

Explanation:
An operation (for example: configure, start, stop) was attempted on a background task, but the task was not in a proper state for that operation, due to earlier errors.

User response:
See the other associated messages for more information.

IQQX0043W **The required property *property_name* is not defined in the configuration.**

Explanation:
The named property is required for backup or restore, but it is not defined in the specified configuration.

User response:
Find the correct value for the property, and add it to the configuration.

IQQX0044W **An error occurred during backup or restore of collection *collection_name*.**

Explanation:
An exception occurred, which caused the backup or restore of the indicated collection to be stopped before completion.

User response:
See the other associated messages for more information, and retry the backup or restore.

IQQX0045W **Collection *collection_name* cannot be restored because the backup volume is missing or empty.**

Explanation:
A collection cannot be restored unless a nonempty backup volume exists.

User response:
Verify that the collection name is correct and that a backup volume exists, and retry the restore.

IQQX0046E **Collection *collection_name* file *file_name* segment *segment_number* could not be backed up.**

Explanation:
One of the segments of the named file was not written to the backup volume. The backup is invalid.

User response:
Verify that the persistent store has disk space, and is reachable, and retry the backup.

IQQX0047E **Collection *collection_name* file *file_name* checksum could not be recorded.**

Explanation:
The checksum of the named file was not written to the backup volume. The backup is invalid.

User response:
Verify that the persistent store has disk space, and is reachable, and retry the backup.

IQQX0048E **Collection *collection_name* file *file_name* could not be deleted from the backup volume.**

Explanation:
The named file was not deleted from the backup volume. The backup is invalid.

User response:
Verify that the persistent store has disk space, and is reachable, and retry the backup.

IQQX0049E **An error occurred when decrypting the backup and restore password.**

Explanation:
The password supplied for backup or restore could not be decrypted.

User response:
See the other associated messages for more information. Verify that the password is correct.

IQQX0050E **An error occurred while starting a backup or restore of collection *collection_name*.**

Explanation:

A backup or restore operation could not be started because of an error in the configuration.

User response:
See the other associated messages for more information. Verify that all configuration values are correct.

IQQX0051E **The XML search query included a path containing only wildcard elements.**

Explanation:
An XML search query included a path containing only wildcard (*) tag or attribute names; such a path is not supported.

User response:
Rewrite the query.

IQQX0052E **The XML search query applied a predicate to a wildcard tag or attribute name.**

Explanation:
An XML search query included a comparison or text search on a wildcard tag or attribute.

User response:
Rewrite the query to specify a tag or attribute name for every predicate.

IQQX0053E **The XML search query specified an element as the child of an attribute.**

Explanation:
XML attributes cannot have descendants.

User response:
Rewrite the query to avoid specifying descendants of attributes.

IQQX0054E **The collection *collection_name* cannot be made quiescent.**

Explanation:
Before you restore a collection, you must make that collection inactive. If this is not possible, due to the system's or collection's current state, the restore cannot proceed, but you can try to restore the collection later.

User response:
Retry the restore later.

IQQX0065E **The system build timestamp is not available for backup version encoding or checking.**

Explanation:
A backup or restore was attempted, but the system build timestamp is null or empty. A build timestamp is needed for backup-volume labeling and version checking during restore.

User response:
Determine the correct timestamp value for the system and record it in the build# properties file as build.timestamp.

IQQX0066E **The backup volume contains an invalid version number *number*.**

Explanation:
The version number found in the backup volume is invalid, possibly indicating data corruption. The system is unable to determine if it is safe to restore from the volume.

User response:
Make sure the backup volume contains the correct kind of data (a successful backup)# and that access to it is error-free.

IQQX0067E **The backup volume version *volumenumber* is not compatible with the server version *servernumber*.**

Explanation:
The version number found in the backup volume is incompatible with the version of the server, because reverse migration is not supported.

User response:
Upgrade the server to the required version.

IQQX0077W **The collection *collection_ID* cannot be backed up because it does not exist.**

Explanation:
The collection specified does not exist.

User response:
Verify that the correct collection name was specified.

IQQX0078E **The collection *collection_ID* cannot be backed up because it is currently being used by another process.**

Explanation:
The specified collection cannot be backed up because it is currently being updated or used by another administrative process.

User response:
Wait until the current administrative process completes and then run back up again.

Information resources for Db2 11 for z/OS and related products

Information about Db2 11 for z/OS and products that you might use in conjunction with Db2 11 is available online in .

You can find the complete set of product documentation for Db2 11 for z/OS in IBM Knowledge Center.

You can also download other PDF format manuals for Db2 11 for z/OS from in .

© Copyright IBM Corp. 2007, 2019

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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, MD-NC119
Armonk, NY 10504-1785 US

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing Legal and Intellectual Property Law IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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 jurisdictions 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.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

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

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 which has been exchanged, should contact:

IBM Director of Licensing IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785 US

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.

© Copyright IBM Corp. 2007, 2019

This information contains 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 actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as shown below:

© (*your company name*) (*year*).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. (*enter the year or years*).

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered marks 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: http://www.ibm.com/legal/copytrade.shtml.

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.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled “Cookies, Web Beacons and Other Technologies” and

the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

Glossary

The glossary is available in IBM Knowledge Center.

See the Glossary topic for definitions of Db2 for z/OS terms.

© Copyright IBM Corp. 2007, 2019

Index

A

accessibility
  keyboard viii
  shortcut keys viii
administration tables
  entries
    troubleshooting 115
  SYSIBMTS.SYSTEXTCOLUMNS 109
  SYSIBMTS.SYSTEXTCONFIGURATION 112
  SYSIBMTS.SYSTEXTCONNECTINFO 111
  SYSIBMTS.SYSTEXTDEFAULTS 107
  SYSIBMTS.SYSTEXTINDEXES 107
  SYSIBMTS.SYSTEXTLOCKS 112
  SYSIBMTS.SYSTEXTSERVERHISTORY 112
  SYSIBMTS.SYSTEXTSERVERS 110
  SYSIBMTS.SYSTEXTSTATUS 111
Administration Tool 104
advanced search operators
  CONTAINS function 79
  SCORE function 79
asynchronous indexing 5
attributes 92
automatic takeover
  responding 64

B

bi-gram segmentation 9

C

client and server communication 1
command-line tools
  Administration Tool 104
  Configuration Tool 102
  Synonym Tool 105
communication
  client and server 1
comparison operators
  optimizing 74
concurrent index updates
  supporting 115
  troubleshooting 115
Configuration Tool 19, 102
configuring
  text search functions 13
  text search servers 13
CONTAINS function
  equality operator 71
  examples 84
  RESULTLIMIT option 84

D

data sharing 5
data types
  supported 7
database administrators 11
Db2 subsystem installers 11
dictionary packs 8
dictionary-based word segmentation 8
disability viii
document formats
  supported 6
document size 7
document truncation 5, 7
document types 5
dropping
  tables 53

E

EBNF grammar 90
elements 92
enabling
  text search support
    Db2 for z/OS 23
equality operator
  optimizing 71
event table
  administering 68
  checking 66
  columns 65
  entries
    removing 68
  overview 65
  single document errors 67

F

fixes
  installing 17
functions
  CONTAINS 69
  SCORE 72
  troubleshooting 115

G

GET DIAGNOSTICS
  messages
    retrieving 64

H

hardware requirements
  IBM Text Search for Db2 for z/OS 3

I

IBM Text Search for Db2 for z/OS

IBM Text Search for Db2 for z/OS *(continued)*
fixes
installing 17
migrating 20, 21
migrating to 20
overview 2
XML search
features 85
IIBM Text Search for Db2 for z/OS
system requirements 3
installation
scenario 13
troubleshooting 18
verifying 28
installing
IBM Text Search for Db2 for z/OS 14
text search functions 13
text search servers
troubleshooting 18
installing Docker images
IBM Text Search for Db2 for z/OS 13

J

Java routines
runtime environment
setting up 26

L

language codes 8
languages
supported 8
lemmatization 8
linguistic processing
Chinese 9
Japanese 9
Korean 9
log consumption
SYSPROC.SYSTS_UPDATE stored procedure 63

M

message OF00010I
responding 64
message OF00801E
troubleshooting 118
messages
retrieving 64
truncated 64
migration
considerations 3
IBM Text Search for Db2 for z/OS 20, 21
incompatibilities 3
multiple search requests
supporting 115
troubleshooting 115

N

n-gram segmentation 8, 9
namespaces
defining 93
namespaces *(continued)*
long names (URI) 92, 93
qualified names (QName) 92, 93

O

operating system requirements
IBM Text Search for Db2 for z/OS 3
text search feature 2
operator precedence 82
OR operator 82

Q

query examples
CONTAINS function 79
SCORE function 79

R

restrictions
search argument syntax 83
RESULTLIMIT option
examples 84
retrieving
messages 64
runtime environment
setting up 26

S

scenario
configuration 13
installation 13
scheduling
SYSPROC.SYSTS_UPDATE stored procedure 63
text search index updates 63
SCORE function
comparison operators 74
examples 84
RESULTLIMIT option 84
search argument syntax
excluding terms 75
fuzzy searches 75
including terms 75
operator precedence 82
OR operator 82
phrase search 75
proximity searches 75
restrictions 83
simple search 75
special characters 77
wildcard character
limitations 83
Security-Enhanced Linux (SELinux)
troubleshooting 18
setting
UPDATE FREQUENCY option 63
setting up
runtime environment 26
WLM environment 25
shortcut keys
keyboard viii

single document errors
event table 67
software requirements
text search feature 2
special characters
escaping 77
SQL code -20212
troubleshooting 117
SQL code -20423
troubleshooting 118
SQL code -430
troubleshooting 116
SQL return codes 119
stored procedures
SYSPROC.SYSTS_ALTER 35
SYSPROC.SYSTS_CREATE 41
SYSPROC.SYSTS_DROP 51
SYSPROC.SYSTS_RESTORE 53
SYSPROC.SYSTS_START 55
SYSPROC.SYSTS_STOP 56
SYSPROC.SYSTS_TAKEOVER 57
SYSPROC.SYSTS_UPDATE 59
tracing 65
troubleshooting 115, 116
WLM environment
setting up 25
synonym dictionaries
adding 99
removing 100
synonym support 5, 7
Synonym Tool 105
syntax diagram
how to read viii
SYSFUN.SYSTS_ENCRYPT user-defined function
running 26
runtime environment
setting up 26
SYSIBMTS.SYSTEXTCOLUMNS table 109
SYSIBMTS.SYSTEXTCONFIGURATION table 112
SYSIBMTS.SYSTEXTCONNECTINFO table
populating 27
SYSIBMTS.SYSTEXTDEFAULTS table 107
SYSIBMTS.SYSTEXTINDEXES table 107
SYSIBMTS.SYSTEXTLOCKS table 112
SYSIBMTS.SYSTEXTSERVERHISTORY table 112
SYSIBMTS.SYSTEXTSERVERS table
populating 27
SYSIBMTS.SYSTEXTSTATUS table 111
SYSPROC.SYSTS_ALTER stored procedure 35
SYSPROC.SYSTS_CREATE stored procedure
UPDATE FREQUENCY option
setting 63
SYSPROC.SYSTS_DROP stored procedure 51
SYSPROC.SYSTS_RESTORE stored procedure 53
SYSPROC.SYSTS_START stored procedure 55
SYSPROC.SYSTS_STOP stored procedure 56
SYSPROC.SYSTS_TAKEOVER stored procedure 57
SYSPROC.SYSTS_UPDATE stored procedure
log consumption 63
message OF00010I 64
scheduling 63
system requirements
IBM Text Search for Db2 for z/OS 3
text search feature 2

T

tables
dropping 53
text scores 5, 7
text search feature
enabling
Db2 for z/OS 23
installation
verifying 28
overview 1
system requirements 2
text search functions
key concepts 5
starting 30
text search index creation 5
text search indexes
altering 31, 35
concurrent updates 115
creating 5, 31
multiple search requests 115
searching 33
updating 5, 32
text search server
messages 119
text search server administrators 11
text search servers
adding 33
configuring 19
data sharing 5
fixes
installing 17
installing 14
installing Docker images 13
migrating
when running 21
when stopped 20
overview 2
starting 99
stopping 99
system requirements 3
uninstalling 101
text search support
enabling 23
timeout abend
preventing 115
trace
enabling 65
tracing
stored procedures 65
triggers 5
troubleshooting
administration tables
entries 115
concurrent index updates 115
functions 115
message OF00801E 118
multiple search requests 115
SQL code -20212 117
SQL code -20423 118
SQL code -430 116
stored procedures 115, 116
timeout abend 115

U

UPDATE FREQUENCY option
    setting 63
user performing search queries 12
user roles
    database administrators 11
    Db2 subsystem installers 11
    text search server administrators 11
    user performing search queries 12
user-defined functions
    SYSFUN.SYSTS_ENCRYPT 74

W

wildcard character
    limitations 83
WLM environment
    setting up 25

X

XML data
    indexing 6
XML search
    attributes 92
    elements 92
    examples 94
    features
        IBM Text Search for Db2 for z/OS 85
    namespaces
        defining 93
    query grammar 90
    scenario 94
XPath language 90

Z

z/OS Unicode Conversion Services 7



Product Number: 5697-Q04

GC19-4070

