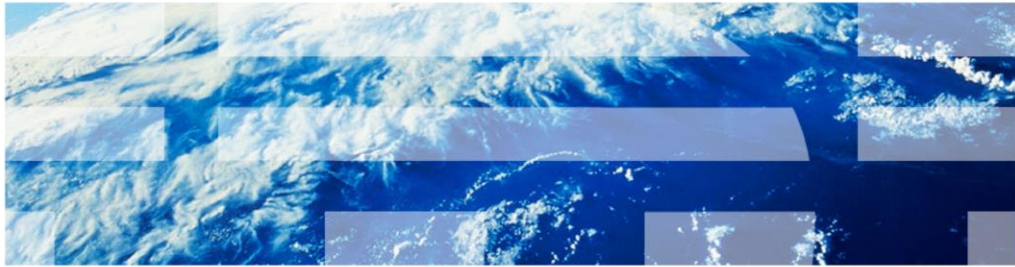


DB2 Text Search

Administration commands



This presentation discusses the administration commands available with DB2® Text Search.

DB2 Text Search administration commands – Overview

- Two ways to run administrative commands
 - Command-line interface
 - Prefixed with 'db2ts'
 - Stored procedure interface
- Database connection capability
 - Explicit using 'CONNECT TO <database-name>'
 - Implicit with 'DB2DBDFT' environmental variable
- Schema for database objects : 'SYSIBMTS'

DB2 Text Search provides both command-line interface and stored procedure interface for running administrative tasks. The administration commands can be ran over the command line from the DB2 Server and the stored procedures can be called from any DB2 client machine with the required privileges.

All DB2 Text Search administration commands must be prefixed with 'db2ts' while executing from a command line. Unlike the DB2 command window, DB2 Text Search uses a separate database connection for each command. This database connectivity can be specified explicitly using 'CONNECT TO database-name' parameter within the db2ts command.

If this is omitted, DB2 Text Search makes an implicit connection to the database specified in the environmental variable 'DB2DBDFT'.

All DB2 Text Search database objects like system catalog tables/views and so on, are created in 'SYSIBMTS' which is a reserved schema.

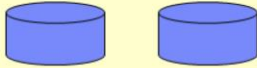
DB2 Text Search administration commands

Instance level



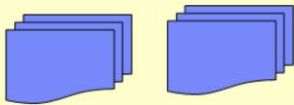
- Start/Stop text search instance services
- Cleanup orphaned text search collections

Database level



- Enable database for text search
- Disable database for text search
- Clear command locks

Text index level



- Create a text index
- Alter a text Index
- Update a text index
- Drop a text index
- Clear events
- Clear command locks
- Reset pending

3

Administration commands

© 2013 IBM Corporation

There are many DB2 Text Search commands that can be administered at Instance, Database and Text Index levels.

At the Instance Level, you can start/stop DB2 Text Search instance services. You can also cleanup orphaned text search collections.

At the Database Level, you can enable or disable the database for Text Search. You can also clear command locks placed on all the text indexes within the database.

At the Text Index Level, you can create, alter, update and drop, a text index. You can also clear events from the event view, clear command locks placed on the index and issue the 'reset pending' command to run set integrity for all dependant staging tables for the index. All these commands are described in detail in the next slides.

DB2 Text Search instance service

- DB2 Text Search instance service
 - Must be started to use any DB2 Text Search functionality
 - Assigned per DB2 instance
 - Configured with TCP/IP port and authentication token to communicate with Text Search Server
- Windows
 - DB2TS - <instance name>[-<partition number>]
- db2ts START FOR TEXT
- db2ts STOP FOR TEXT

DB2 Text Search instance service must be started in order to use any DB2 Text Search administration commands and to search the text columns. It is different from DB2 service and is assigned per DB2 instance. It should be configured with TCP/IP port and authentication token to communicate with the Text Search Server.

On a Windows platform, there is a Windows service associated with DB2 instance for Text Search with the service name: DB2TS - <instance name>[-<partition number>].

DB2 Text Search instance service can be started using 'db2ts START FOR TEXT' command and stopped using 'db2ts STOP FOR TEXT' command.

Enable database

- Enables current database for DB2 Text Search
- Creates required database objects in schema SYSIBMTS
- Verifies parameter values in SYSIBMTS.TSSERVERS view especially PORT and TOKEN
- Command Syntax: Methods to enable database
 - Issue Enable Database command
 - Set environmental variable DB2DBDFT
 - export DB2DBDFT=mydb
 - db2ts ENABLE DATABASE FOR TEXT
 - Use 'CONNECT TO dbname':
 - db2ts ENABLE DATABASE FOR TEXT CONNECT TO dbname
 - Call SYSPROC.SYSTS_ENABLE() procedure
 - CALL SYSPROC.SYSTS_ENABLE('en_US', ?)

The ENABLE DATABASE command enables the current database for DB2 Text Search. This step is mandatory before creating text indexes in the database. It creates the required database objects like catalog tables and views, stored procedures and so on, in the schema SYSIBMTS.

Once the database is enabled, verify the values of PORT and TOKEN in the SYSIBMTS.TSSERVERS view so that the connectivity to the text server can be established.

DB2 Text Search provides both command-line interface and stored procedure interface for executing administrative commands.

The command syntax shows how to enable a database for DB2 Text Search through the Enable Database command and the SYSPROC.SYSTS_ENABLE() procedure.

Disable database

- Disables database for DB2 Text Search
- FORCE option drops all text indexes and associated collections
- Command Syntax: Methods to disable database
 - Issue Disable Database command
 - db2ts DISABLE DATABASE FOR TEXT
 - Call SYSPROC.SYSTS_DISABLE() procedure
 - CALL SYSPROC.SYSTS_DISABLE(' ', 'en_US', ?)

The DISABLE DATABASE command disables the database for DB2 Text Search and drops all database objects created by the 'Enable' command.

When started with the FORCE option, it drops all text indexes and associated collections. The command syntax shows how to disable a database for DB2 Text Search through the Disable Database command and the SYSPROC.SYSTS_DISABLE() procedure.

Create index

- Creates full-text index on text column (empty collection)
- Requires primary key on user table
- Database objects created
 - Staging table
 - Event table
 - Auxiliary staging table (only when integrity processing is ON)

The CREATE INDEX command creates a full-text index on a text column, internally an empty collection is created in the file system. It requires a primary key on the user table so that the documents are uniquely identifiable.

During the creation of a text index, other database objects are also created. There is a Staging table to keep track of all the changes that happened on the base table. An Event table to store information about all updates and error and warning messages occurred during the update of the index. Also, an auxiliary staging table is created only when integrity processing is enabled to support range partitioned table or a base table populated with data using the LOAD command.

Example: Create index

- Methods to create index
 - Issue CREATE INDEX command
 - db2ts "CREATE INDEX *index-schema.index-name* FOR TEXT ON *table-name* (*column-name*)"
 - Call SYSPROC.SYSTS_CREATE() procedure
 - CALL SYSPROC.SYSTS_CREATE('index-schema', 'index-name', 'table-name(column-name)', 'options', 'locale', ?)
- Example:
 - db2ts "CREATE INDEX myschema.mytextindex FOR TEXT ON product(name) UPDATE FREQUENCY d(1,3) h(0,12) m(0) UPDATE MINIMUM 5"
 - CALL SYSPROC.SYSTS_CREATE('myschema', 'mytextindex', 'PRODUCT(NAME)', 'UPDATE FREQUENCY D(1,3) H(0,12) M(0)' 'UPDATE MINIMUM 5', 'en_US', ?)

There are two methods to create a text search index. First, you can issue the CREATE INDEX command. Secondly, you can call the SYSPROC.SYSTS_CREATE() stored procedure.

You should provide the table name and column name on which the text index should be created. For example, to specify that the index myschema.mytextindex should be updated after at least five changes have occurred and that the update services should check every Monday and Wednesday at 12 midnight and 12 noon, issue one of the commands as displayed on this slide.

Update index

- Synchronizes text index with base table
- Manual update
- Scheduled update
- Methods to update index
 - Issue UPDATE INDEX command
 - db2ts "UPDATE INDEX *index-schema.index-name* FOR TEXT"
 - Call SYSPROC.SYSTS_UPDATE() procedure
 - CALL SYSPROC.SYSTS_UPDATE('index-schema', 'index-name', '', 'en_US', ?)
- Example
 - db2ts "UPDATE INDEX myschema.myindex FOR TEXT"
 - call sysproc.systs_update('myschema', 'myindex', '', 'en_US', ?)

After the index is created, it is empty and should be populated with data by running the update index command either manually or automatically by way of the scheduler. To update the text search index, either issue the update index command or call the SYSPROC.SYSTS_UPDATE() stored procedure.

Note that index-schema and index-name are case sensitive when a stored procedure is used. The example displayed on this slide shows how to update a text index named 'myschema.myindex'. For more information on the command parameters, look into the Information Center.

Alter index

- Changes properties of index
- Also used to activate/deactivate text search index
- Methods to alter an index
 - Issue ALTER INDEX command
 - db2ts "ALTER INDEX *index-schema.index-name* FOR TEXT *alter-option*"
 - Call SYSPROC.SYSTS_ALTER() procedure
 - CALL SYSPROC.SYSTS_ALTER(*'index-schema'*, *'index-name'*, *'alter-option'*, *'en_US'*, ?)
- Example
 - db2ts "ALTER INDEX myschema.myindex FOR TEXT UPDATE FREQUENCY d(1,2,3,4,5) h(12,15) m(00) UPDATE MINIMUM 100"
 - CALL SYSPROC.SYSTS_ALTER(*'myschema'*, *'myindex'*, *' UPDATE FREQUENCY d(1,2,3,4,5) h(12,15) m(00) UPDATE MINIMUM 100'*, *'en_US'*, ?)

The Alter Index command is used to change the properties of the index, for example, the update frequency. It is also used to activate or deactivate the text search index in case both Net Search Extender (NSE) and Text Search (TS) indexes exist on the same table column.

To alter an index, either issue the ALTER INDEX command or call the SYSPROC.SYSTS_ALTER() procedure.

In this case, *alter-option* is the option used to specify the properties of the index along with new values.

The example displayed on this slide shows how to alter an index so that it is updated from Monday to Friday at 12 noon and 3:00 PM provided that the minimum changes occurred to the indexed column are 100. For more information on the command parameters, look into the Information Center.

Drop index

- Drops index when no need to perform text searches on column
- Drops all related database objects and collection
- Methods to drop text index
 - Issue DROP INDEX command
 - db2ts "DROP INDEX *index-schema.index-name* FOR TEXT"
 - Call SYSPROC.SYSTS_DROP() procedure
 - CALL SYSPROC.SYSTS_DROP(*'index-schema', 'index-name', 'en_US', ?*)
- Example
 - db2ts "DROP INDEX myschema.myindex FOR TEXT"
 - call sysproc.systs_drop(*'myschema', 'myindex', 'en_US', ?*)

A text index can be dropped when there is no need to perform text searches on the column. It also drops the staging table, event table, triggers associated with the text index and finally deletes the collection in the file system. To drop the text index either issue the DROP INDEX command or call SYSPROC.SYSTS_DROP() procedure. The example displayed on this slide shows how to drop a text index with name 'myschema.myindex'.

Cleanup for text

- Cleans up Text Search orphaned collections
- Possible reasons for orphaned collection
 - Dropping database containing text indexes
 - Dropping table space containing text indexes
 - Disabling database with FORCE option when text search server was not started
 - Any other failure cases where text index goes to an inconsistent state
- Command Syntax: Methods to cleanup the orphaned collections
 - At instance level
 - db2ts CLEANUP FOR TEXT
 - At database level
 - db2ts CLEANUP FOR TEXT CONNECT TO database-name USER userid USING password
 - Call SYSPROC.SYSTS_CLEANUP() procedure
 - CALL SYSPROC.SYSTS_CLEANUP(' ', 'en_US', ?)

The Cleanup for text command cleans up the Text Search orphaned collections within an instance or database. A collection is known as an 'orphaned collection' when a text search index no longer exists but the corresponding collection still exists in the file system. Always drop the text indexes before dropping the database or table space, otherwise you end up in creating orphaned collections.

The example displayed on this slide shows how to cleanup the orphaned collections at the instance level and database level through the CLEANUP FOR TEXT command and the SYSTS_CLEANUP stored procedure

Clear command locks

- Releases all command locks
- Use command cautiously only when there is a need to clear locks explicitly
- Do not release locks while active index update in progress
- Command Syntax: Methods to clear command locks
 - At database level
 - db2ts CLEAR COMMAND LOCKS FOR TEXT
 - At index level
 - db2ts CLEAR COMMAND LOCKS FOR INDEX index-name FOR TEXT
 - Call SYSPROC.SYSTS_CLEAR_COMMANDLOCKS() procedure
 - CALL SYSPROC.SYSTS_CLEAR_COMMANDLOCKS ('index-schema', 'index-name', 'en_US', ?)

In order to prevent conflict with other administration commands, a command lock is created at the beginning of an execution of a command and is removed once it is done.

This command helps to remove all such locks for a specific text index or for all text indexes in a specified database. Use this command more cautiously only when there is a need to clear the locks explicitly, for example, the update process was hung all of a sudden leaving the locks.

Do not release the locks while an active index update is in progress. The example displayed on this slide shows how to clear the command locks for a specific index and for all indexes in a specified database through the CLEAR COMMAND LOCKS command and the SYSTS_CLEAR_COMMANDLOCKS() stored procedure.

Clear events

- Deletes indexing events
- Regularly check event view, correct document errors and resubmit
- Command Syntax: Methods to clear indexing events
 - Issue CLEAR EVENTS
 - db2ts CLEAR EVENTS FOR INDEX index-name FOR TEXT
 - Call SYSPROC.SYSTS_CLEAR_EVENTS() procedure
 - CALL SYSPROC.SYSTS_CLEAR_EVENTS ('index-schema', 'index-name','en_US',
?)

The Clear events command deletes the indexing events from an index's event view. Regularly check the event view in case of scheduled updates, identify the document errors and correct them. Ensure that the next update has successfully re-indexed the erroneous documents and then, clear the events. The example displayed on this slide shows how to clear the events for a specified index through the CLEAR EVENTS command and the SYSTS_CLEAR_EVENTS() stored procedure.

Reset pending

- Calls SET INTEGRITY command on all text maintained staging tables for all text indexes associated to specified table
- Command Syntax
 - db2ts RESET PENDING FOR TABLE table-schema.table-name FOR TEXT

The Reset pending command calls the SET INTEGRITY command on all text maintained staging tables for all text indexes associated to the specified table. Issue this command after executing a command that causes the underlying or dependant tables to go into pending mode such as the LOAD command or the ATTACH/DETACH command. The syntax for this command is displayed on this slide.

Set command locks

- Manually sets a lock on an index or on all unlocked indexes in database
- Command Syntax: Methods to manually set a lock
 - On an index
 - db2ts SET COMMAND LOCKS FOR INDEX index-name FOR TEXT
 - On all unlocked indexes within database
 - db2ts SET COMMAND LOCKS FOR TEXT

The Set command locks is a command that is used to manually set a lock on an index or on all unlocked indexes in a database. Once the lock is applied, it is visible in the SYSIBMTS.TSLOCKS administrative view. The command syntax shows how to apply lock at an index level and database level.

SYSTS_ADMIN_CMD procedure

- To run DB2 Text Search administrative commands using SQL CALL statement
- Supported commands
 - ALTER INDEX
 - CLEAR COMMAND LOCKS
 - CLEAR EVENTS
 - CREATE INDEX
 - DISABLE DATABASE
 - DROP INDEX
 - ENABLE DATABASE
 - RESET PENDING
 - UPDATE INDEX
- Example
 - CALL SYSPROC.SYSTS_ADMIN_CMD ('CREATE INDEX mytextindex FOR TEXT ON mytable (mycolumn)', 'en_US', ?)

The SYSTS_ADMIN_CMD procedure is used by the applications to run DB2 Text Search administrative commands using the SQL CALL statement. Several different commands can be issued through this procedure. Commands include the ALTER INDEX command, the CLEAR COMMAND LOCKS command, the CLEAR EVENTS command and the CREATE INDEX command. There is also the DISABLE DATABASE command, the DROP INDEX command, the ENABLE DATABASE command, the RESET PENDING command and finally, the UPDATE INDEX command.

The example displayed on this slide shows how to create an index 'mytextindex' on the column 'mycolumn' belonging to the table 'mytable'.

SYSTS_CONFIGURE procedure

- Updates text search catalog with text search server connection information
- Must be called in certain cases
 - Incomplete enablement
 - Stand-alone text server setup
 - Partitioned database environment
- Command Syntax:
 - CALL SYSPROC.SYSTS_CONFIGURE(", 'en_US', ?)

The SYSTS_CONFIGURE stored procedure updates the text search catalog view SYSIBMTS.TSSERVERS with the text search server connection information. It should be started initially for incomplete enablement, for example, when there is any error returned while enabling the database, started when doing a stand-alone text server setup, and used on a partitioned database environment.

And later on, this stored procedure must be called whenever there is an update on SYSIBMTS.TSSERVERS catalog view. See the example displayed on this slide.

DB2 Text Search Commands <-> Stored Procedures

Command	SP	Changes
Start		Start text search server, local or remote; on Windows: service
Stop		Stop text search server, local or remote
Enable	X	Create text search catalog tables (schema SYSIBMTS)
Disable	X	Drop text search catalog tables
Create	X	Create empty text search index, type TEXT in SYSINDEXES
Alter	X	Change selected text search index properties
Update	X	Initial and incremental updates; for DPF distributed
Drop	X	Drop text search index and associated collection data
Clear Events	X	Delete entries from event table
Set locks		Create command locks
Clear Locks	X	Remove command locks for cleanup after crash
Cleanup	X	Remove orphaned collections
Configure	X	Update text search server properties
Reset Pending		Convenience command to 'set integrity' for text-maintained MQT

19

Administration commands

© 2013 IBM Corporation

DB2 Text Search provides several administrative stored procedures for running the commands. This slide displays whether a corresponding stored procedure exists or not for each Text Search command. The 'Cross' mark shows there is a corresponding stored procedure available.

Additional resources

- IBM Education Assistant modules
 - DB2 Text Search - [Overview](#)
 - DB2 Text Search - [Indexing](#)
 - DB2 Text Search - [Installation, configuration and upgrade](#)
 - DB2 Text Search - [Search features](#)
 - DB2 Text Search - [Command line tools](#)

- IBM Information Center
 - <http://pic.dhe.ibm.com/infocenter/db2luw/v10r5/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0051296.html>

This slide displays additional IBM Education Assistant modules related to DB2 Text Search and it provides the link to the IBM Information Center.

Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and DB2 are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

Windows, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2013. All rights reserved.