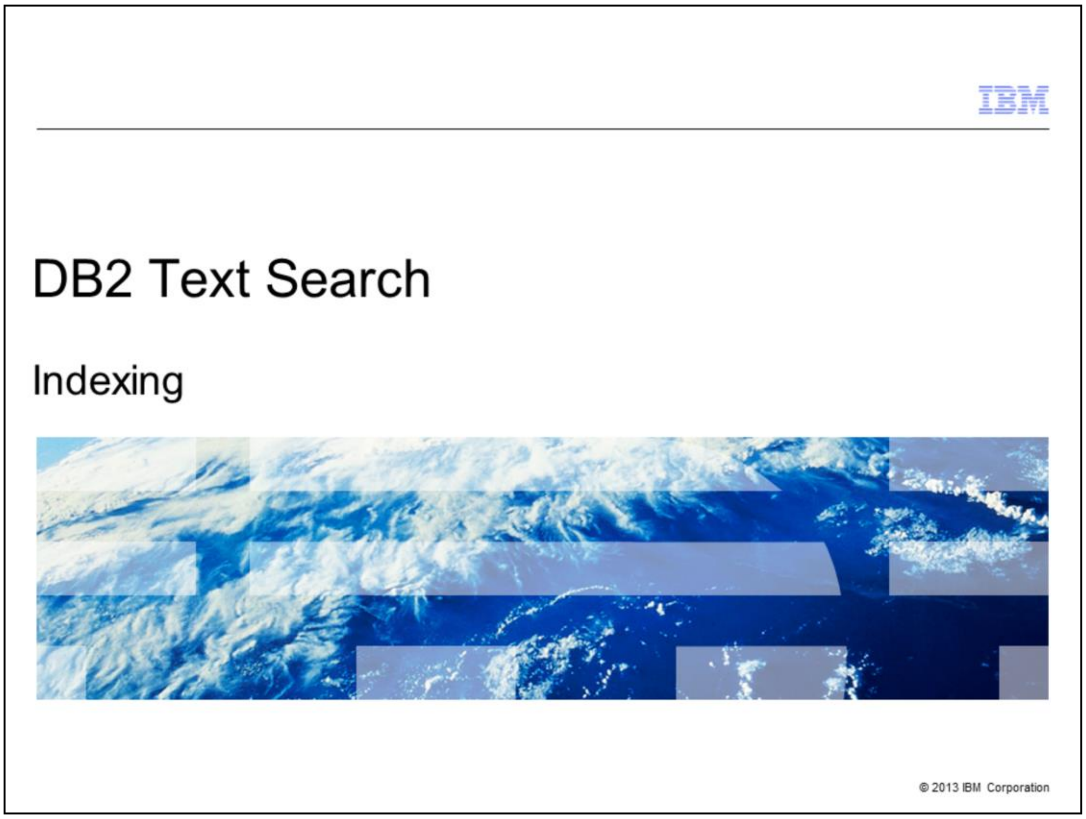
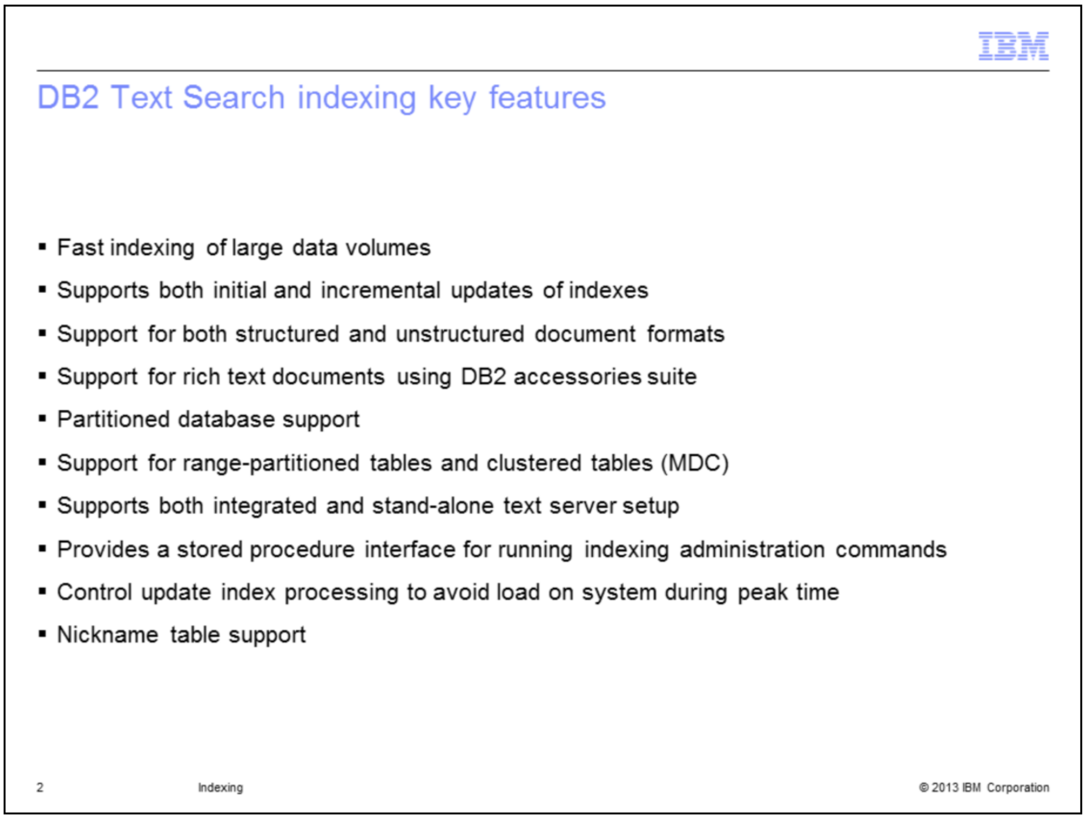


DB2 Text Search

Indexing

This presentation describes DB2® Text Search indexing in detail.

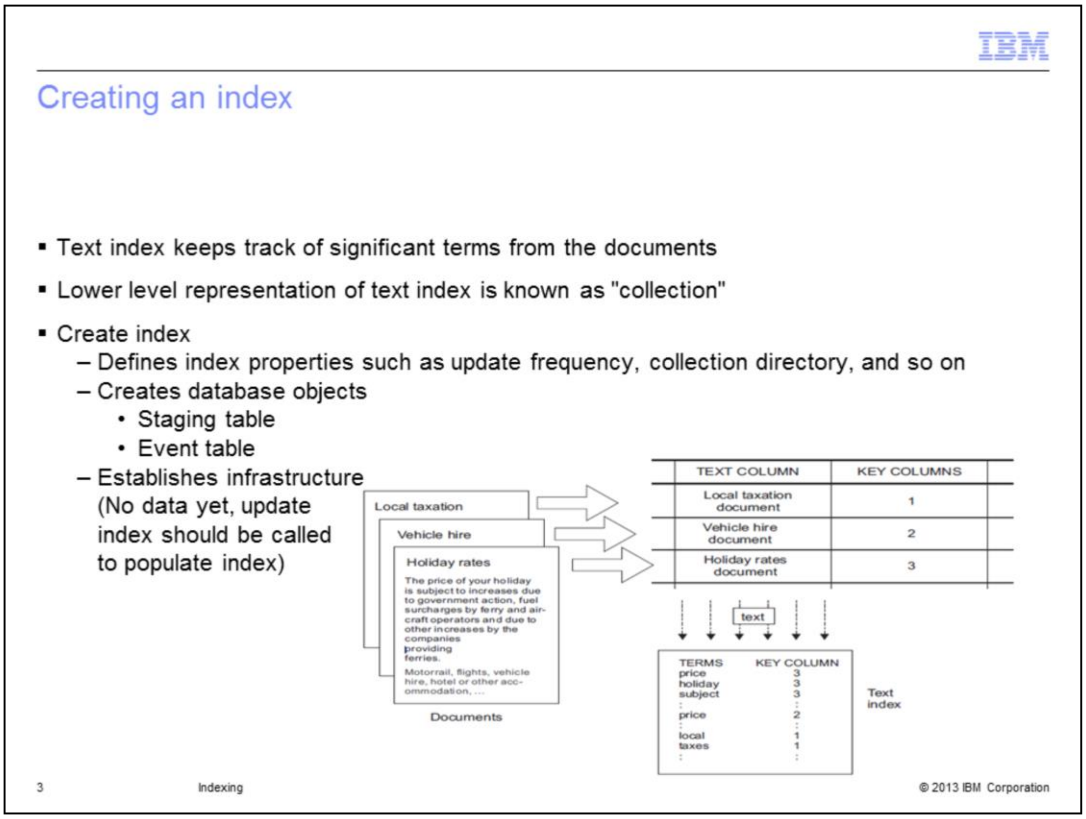


DB2 Text Search indexing key features

- Fast indexing of large data volumes
- Supports both initial and incremental updates of indexes
- Support for both structured and unstructured document formats
- Support for rich text documents using DB2 accessories suite
- Partitioned database support
- Support for range-partitioned tables and clustered tables (MDC)
- Supports both integrated and stand-alone text server setup
- Provides a stored procedure interface for running indexing administration commands
- Control update index processing to avoid load on system during peak time
- Nickname table support

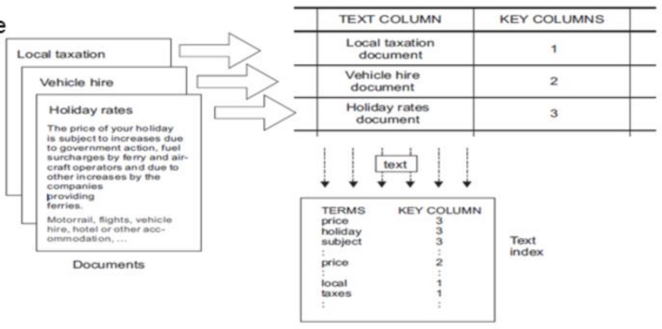
2 Indexing © 2013 IBM Corporation

The key features of DB2 Text Search includes fast indexing of large data volumes, it supports both initial and incremental update of indexes which is done asynchronously so as not to impact performance of search applications and it supports both structured and unstructured document formats. For example plain text, HTML and XML. More key features include the support of rich text documents with the DB2 Accessories Suite, it supports the database partitioning feature, range partitioned tables and clustered tables. DB2 Text Search supports both integrated and stand-alone text server setup. A stand-alone setup is the one where the DB2 database server and the Text Search server resides on different machines to separate out the resource intensive text index processing with that of the database server processing. It provides a stored procedure interface for running indexing administration commands and it controls the update index processing. The administrators can now define a time window for updating the text index, thus, avoiding load on the system during peak time. With the Nickname table support, it can index and search the documents in other relational databases.



Creating an index

- Text index keeps track of significant terms from the documents
- Lower level representation of text index is known as "collection"
- Create index
  - Defines index properties such as update frequency, collection directory, and so on
  - Creates database objects
    - Staging table
    - Event table
  - Establishes infrastructure (No data yet, update index should be called to populate index)


Local taxation
Vehicle hire
Holiday rates
The price of your holiday is subject to increases due to government action, fuel surcharges by ferry and air-craft operators and due to other increases by the companies providing ferries.
Motorrail, flights, vehicle hire, hotel or other acc-ommodation, ...
Documents
TEXT COLUMN
KEY COLUMNS
Local taxation document
1
Vehicle hire document
2
Holiday rates document
3
text
TERMS
KEY COLUMN
price 3
holiday 3
subject 3
price 2
local 1
taxes 1
Text index

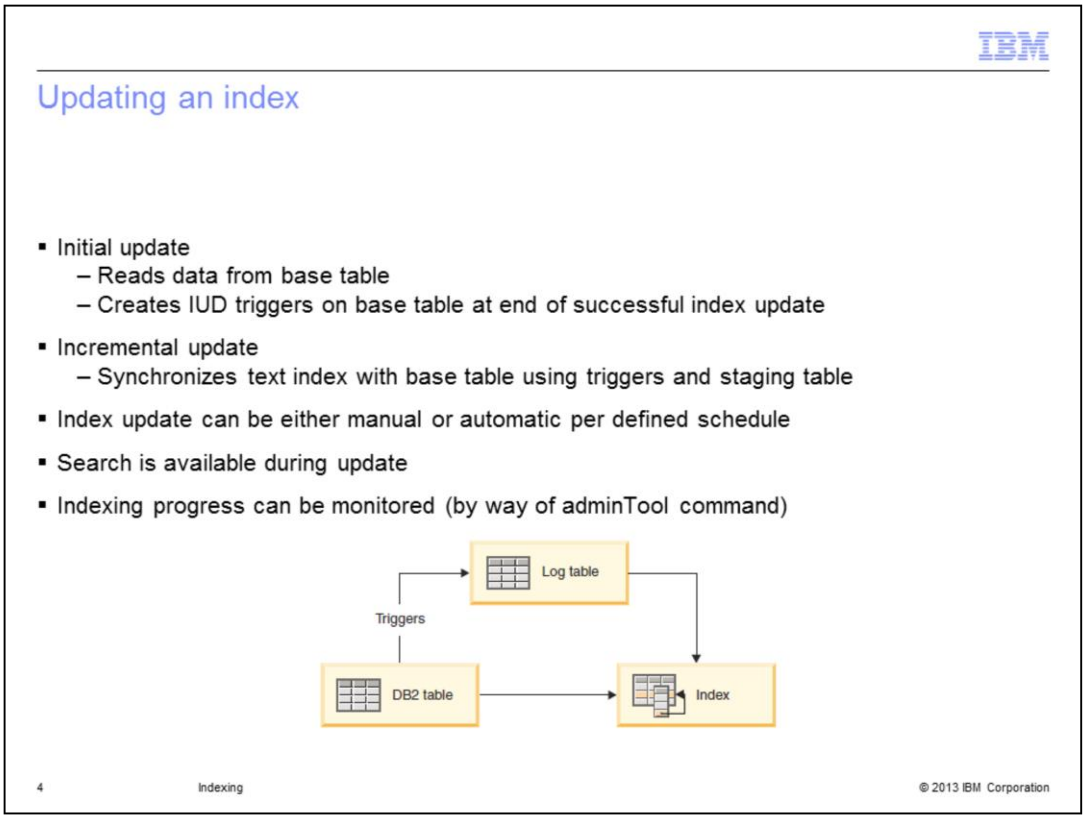
3 Indexing © 2013 IBM Corporation

Basically, DB2 Text Search provides the capability to search the documents that are stored in a table column of your DB2 database. It uses the primary key of the table to uniquely identify the document. A text index should be created in order to efficiently search the documents. As displayed on this slide, DB2 Text Search extracts significant terms from the document, which are used to build a text index. All the updates to the table column are tracked and synchronized with the text index through the next incremental update.

When creating a text index, a collection is automatically created. It is a lower level representation of the text index. The index creation defines the index properties like update frequency to schedule automated updates, collection directory where the text search index collection is stored, and so on. It also creates a staging table to track and record all changes that happened on the base table and the event table to record the status of all updates during the update of the text index.

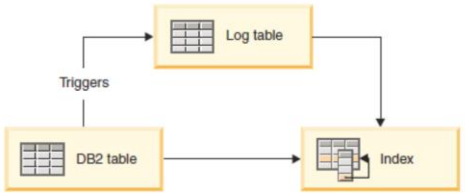
After you create the text search index, it is empty and hence, it is not searchable until you update it.

Refer to the presentation DB2 Text Search - Administration commands for more details on the usage of the create index command.



Updating an index

- Initial update
  - Reads data from base table
  - Creates IUD triggers on base table at end of successful index update
- Incremental update
  - Synchronizes text index with base table using triggers and staging table
- Index update can be either manual or automatic per defined schedule
- Search is available during update
- Indexing progress can be monitored (by way of adminTool command)



4 Indexing © 2013 IBM Corporation

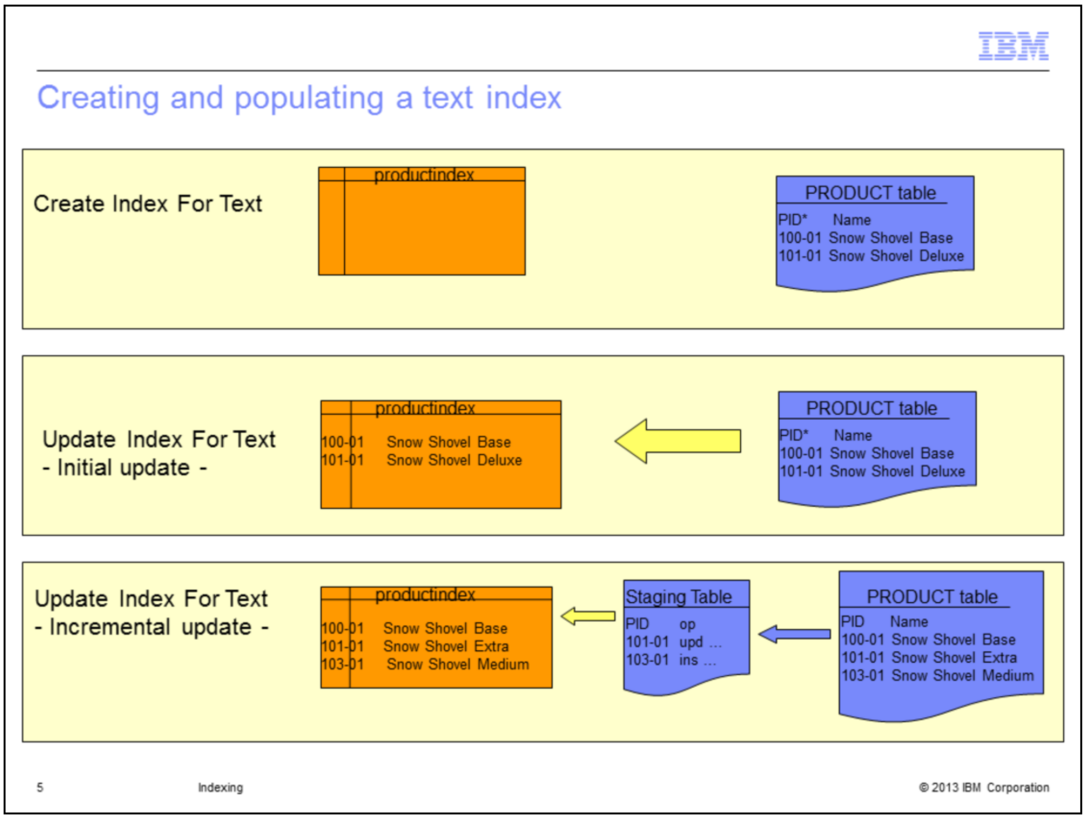
The text index should be updated with the data from the table column to perform searches. Updating the index for the first time, known as the initial update, adds the data about all the documents from the column to the text index and creates insert, update and delete triggers on the base table to capture the changes.

Subsequent updates, known as incremental updates, synchronize the data in the table and the text index. The incremental updates run asynchronously and can be started manually or automatically through a schedule. The text index update is not in the same transaction as the data update. Instead, for each insert, update and delete that happened on the column, an entry is inserted by way of a trigger into a staging table. This staging table is then used to control the incremental update and to synchronize the text index with the data in the base table.

It is also possible to use operating system functions. For example, crontab to schedule index updates.

You can monitor the index update progress using the DB2 Text Search administration tool.

See the presentation DB2 Text Search - Administration commands for more details on the usage of the update index command.



Creating and populating a text index

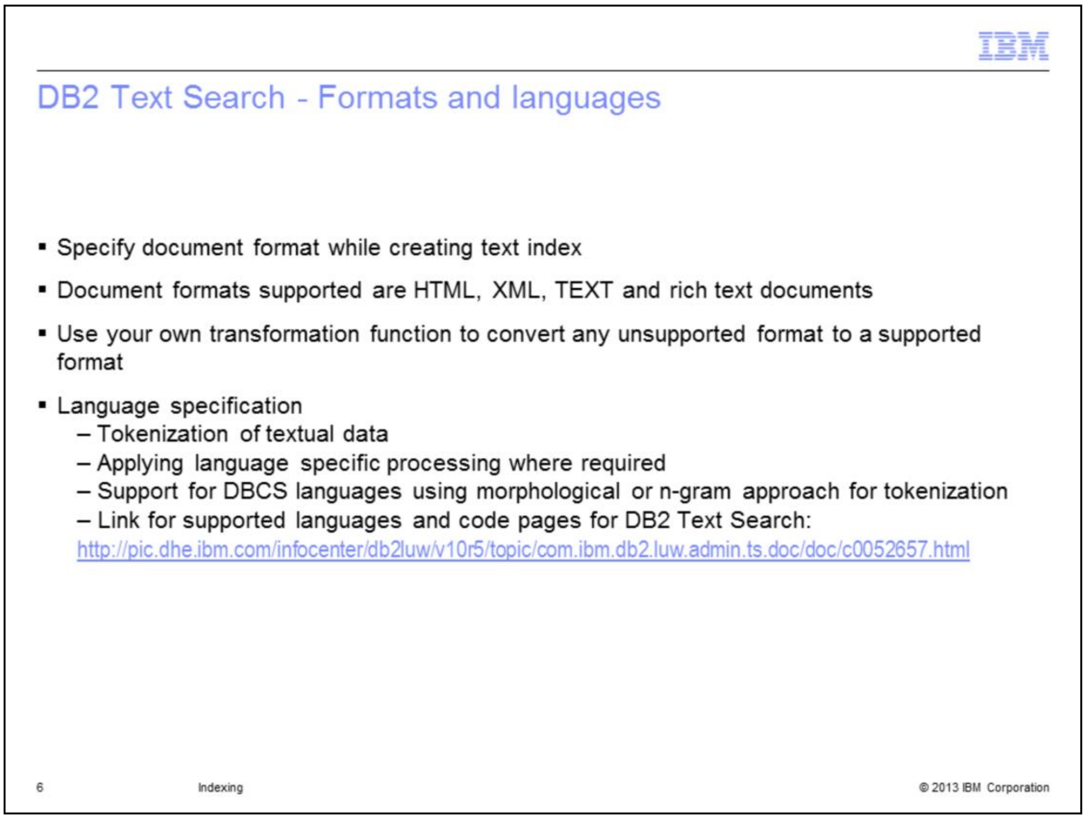
5 Indexing © 2013 IBM Corporation

The figure displayed on this slide illustrates the concepts of initial and incremental update index.

There is a text index as shown by the orange colored block called "productindex" created on the table as shown by the blue colored block called "PRODUCT".

Once the index is created, it is empty, no data is yet available. An empty collection is created by the text search server. As depicted from the figure, the initial index update added data about all the documents in the table column to the text index. At this point, both text index and table are synchronized.

There is a new insert and update operation on the PRODUCT table which is tracked by the triggers and captured into the staging table. An incremental index update is issued and all these changes are applied on the index.



IBM

DB2 Text Search - Formats and languages

- Specify document format while creating text index
- Document formats supported are HTML, XML, TEXT and rich text documents
- Use your own transformation function to convert any unsupported format to a supported format
- Language specification
  - Tokenization of textual data
  - Applying language specific processing where required
  - Support for DBCS languages using morphological or n-gram approach for tokenization
  - Link for supported languages and code pages for DB2 Text Search: http://pic.dhe.ibm.com/infocenter/db2luw/v10r5/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0052657.html

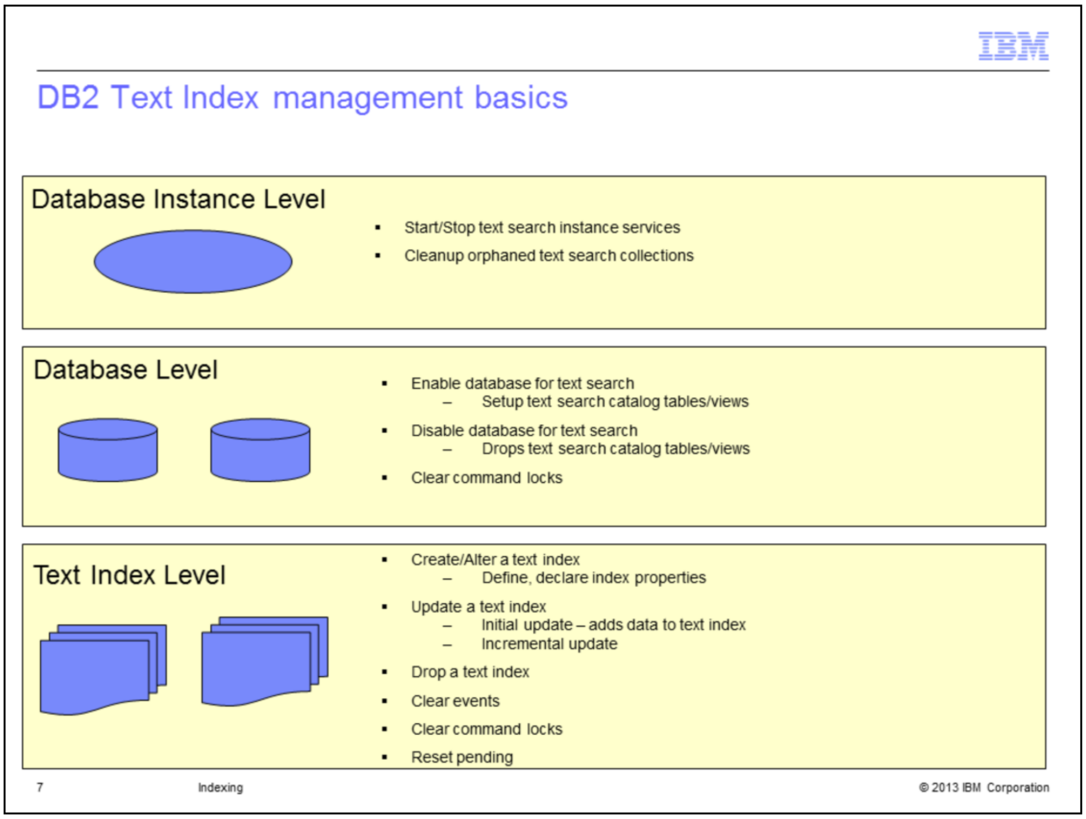
6 Indexing © 2013 IBM Corporation

During the create index, specify the document format being indexed. The document formats supported are Hypertext Markup Language, or HTML, Extended Markup Language, referred to as XML and TEXT, which is plain text. Also, rich text document formats like Microsoft® Office, Open Document, pdf and so on, using the DB2 Accessories Suite, are supported. For more details on the DB2 Text Search Accessories Suite, see the DB2 Text Search - Installation, Configuration and Upgrade presentation.

You can use your own transformation function to convert any unsupported format to a supported format.

The locale specified while creating the text index determines the language to tokenize the document. It also helps to perform linguistic processing on the search term. For DBCS languages, both morphological and n-gram approaches are supported.

For a list of supported languages and code pages for DB2 Text Search, visit the Information Center. The link is displayed on this slide.



DB2 Text Index management basics

Database Instance Level

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

Database Level

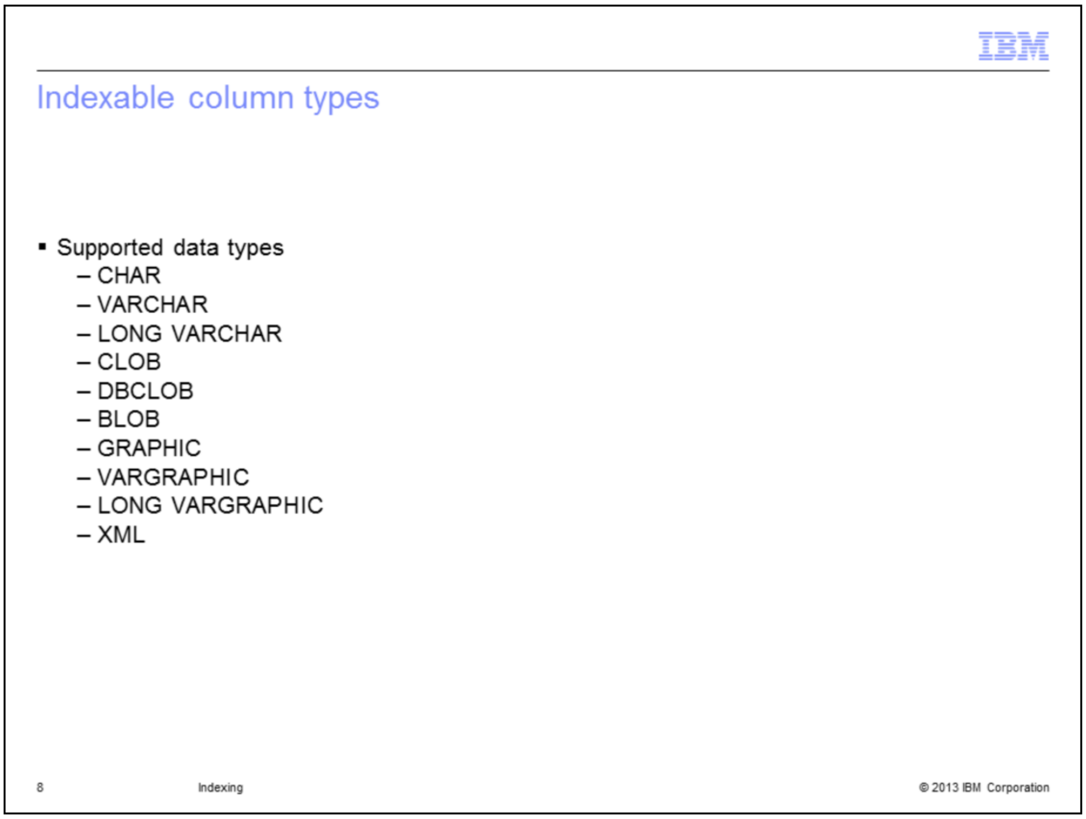
- Enable database for text search
  - Setup text search catalog tables/views
- Disable database for text search
  - Drops text search catalog tables/views
- Clear command locks

Text Index Level

- Create/Alter a text index
  - Define, declare index properties
- Update a text index
  - Initial update – adds data to text index
  - Incremental update
- Drop a text index
- Clear events
- Clear command locks
- Reset pending

This slide illustrates the various administration commands of DB2 Text Search at instance, database and text index level.

At the Instance Level, you can start and stop DB2 Text Search instance services. You can also cleanup orphaned text search collections. At the Database Level, you can enable and disable the database for Text Search, you can also clear command locks placed on all the text indexes with in 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 start the ‘reset pending’ command to run set integrity for all dependant staging tables for the index.



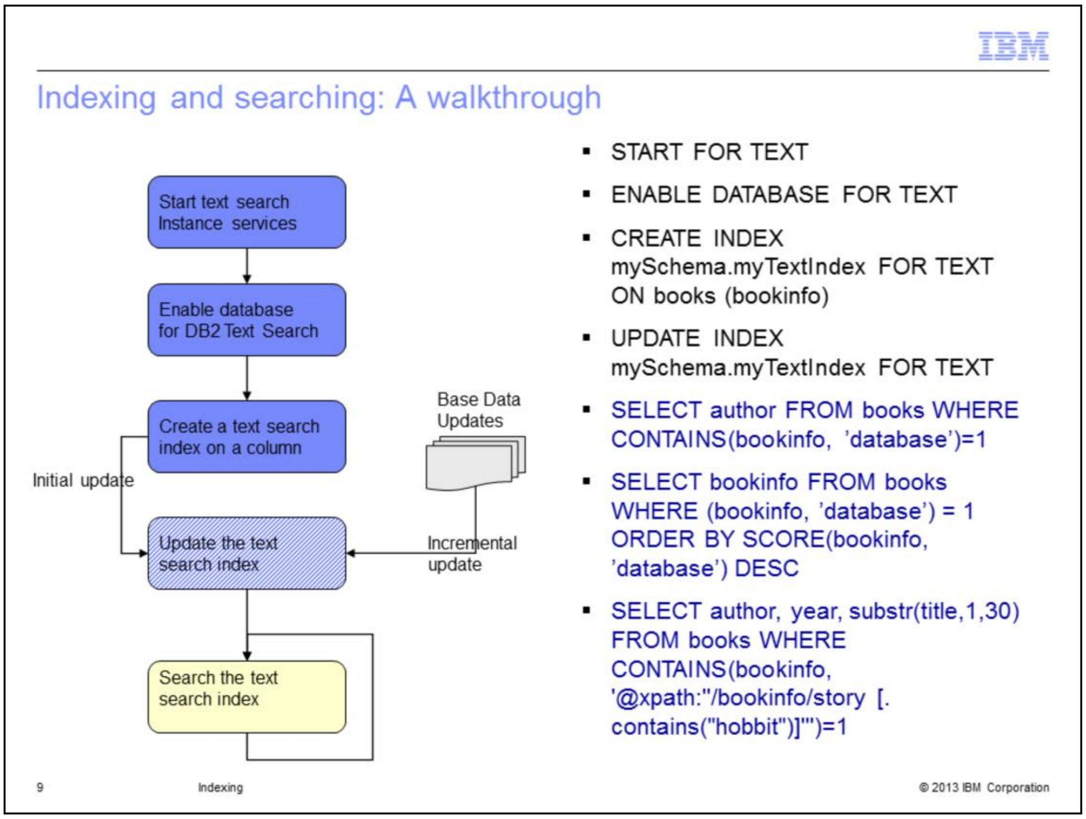
Indexable column types

- Supported data types
  - CHAR
  - VARCHAR
  - LONG VARCHAR
  - CLOB
  - DBCLOB
  - BLOB
  - GRAPHIC
  - VARGRAPHIC
  - LONG VARGRAPHIC
  - XML

8 Indexing © 2013 IBM Corporation

A text index can be created on many different data types, including CHAR, VARCHAR, LONG VARCHAR, CLOB, DBCLOB, BLOB, GRAPHIC, VARGRAPHIC, LONG VARGRAPHIC and XML.

You can use your own transformation function to convert any unsupported data type to a supported data type.


Start text search Instance services
Enable database for DB2 Text Search
Create a text search index on a column
Initial update
Base Data Updates
Update the text search index
Incremental update
Search the text search index

Indexing and searching: A walkthrough

- START FOR TEXT
- ENABLE DATABASE FOR TEXT
- CREATE INDEX mySchema.myTextIndex FOR TEXT ON books (bookinfo)
- UPDATE INDEX mySchema.myTextIndex FOR TEXT
- SELECT author FROM books WHERE CONTAINS(bookinfo, 'database')=1
- SELECT bookinfo FROM books WHERE (bookinfo, 'database') = 1 ORDER BY SCORE(bookinfo, 'database') DESC
- SELECT author, year, substr(title,1,30) FROM books WHERE CONTAINS(bookinfo, '@xpath:"/bookinfo/story [. contains("hobbit")]"')=1

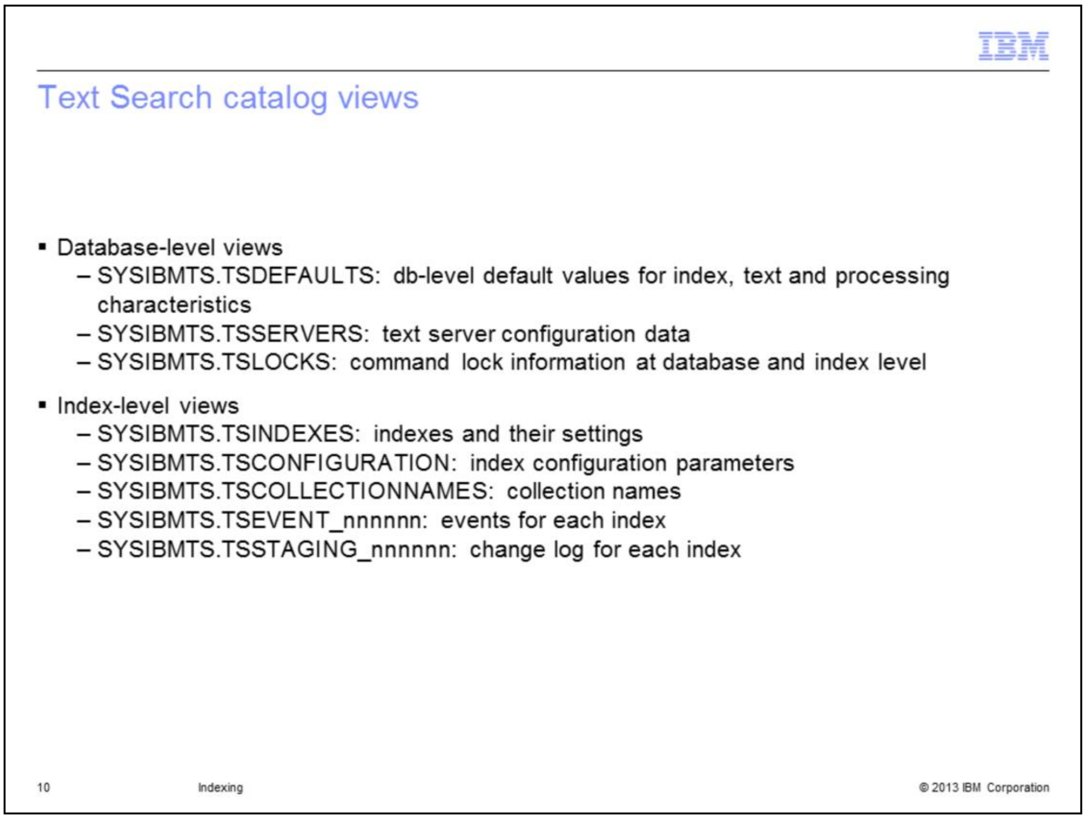
9 Indexing © 2013 IBM Corporation

This slide describes a sample scenario to explain the concept of indexing in DB2 Text Search. First, DB2 Text Search instance service should be started using the command "START FOR TEXT".

Enable the database using the command "ENABLE DATABASE FOR TEXT". It creates system catalog tables and views for storing text index specific information. Next, identify the table column for searching the data and create the index using the "CREATE INDEX" command. In this scenario, text index myschema.mytextindex is created on the column "bookinfo" of table "book". It creates an empty text index and additional infrastructure in the database used to populate the text index.

Update the text index using the "UPDATE INDEX" command. It can be either an initial or an incremental update.

Now that the index is populated and up-to-date, searches can be issued.

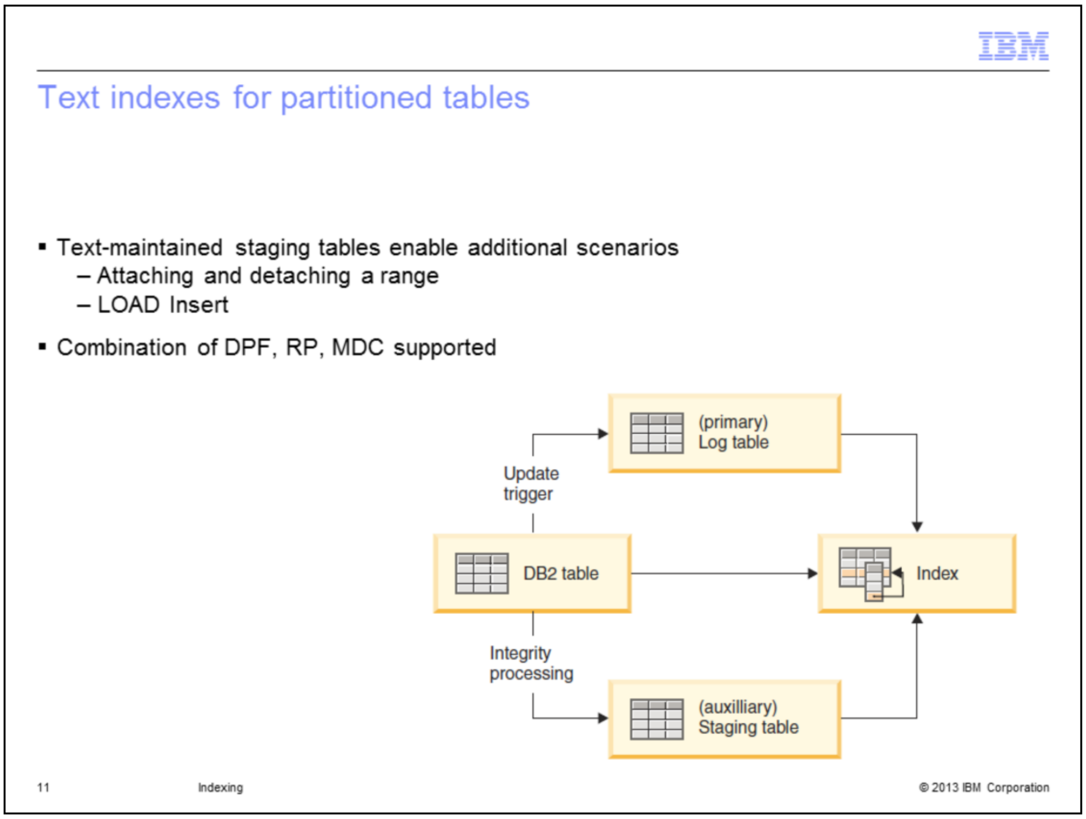

Text Search catalog views

- Database-level views
  - SYSIBMTS.TSDEFAULTS: db-level default values for index, text and processing characteristics
  - SYSIBMTS.TSSERVERS: text server configuration data
  - SYSIBMTS.TSLOCKS: command lock information at database and index level
- Index-level views
  - SYSIBMTS.TSINDEXES: indexes and their settings
  - SYSIBMTS.TSCONFIGURATION: index configuration parameters
  - SYSIBMTS.TSCOLLECTIONNAMES: collection names
  - SYSIBMTS.TSEVENT_nnnnnn: events for each index
  - SYSIBMTS.TSSTAGING_nnnnnn: change log for each index

10 Indexing © 2013 IBM Corporation

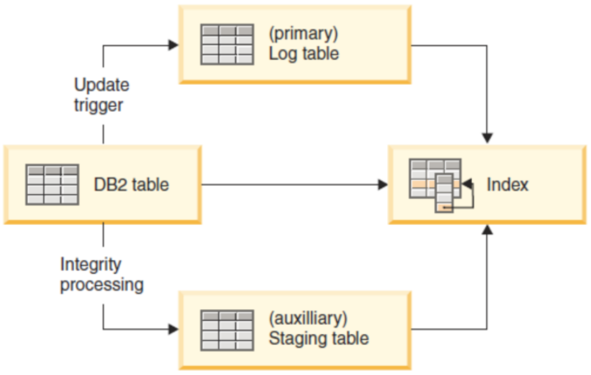
This slide displays the catalog views created in the database when it is enabled for Text Search. The Database-level views are SYSIBMTS.TSDEFAULTS, SYSIBMTS.TSSERVERS and SYSIBMTS.TSLOCKS. SYSIBMTS.TSDEFAULTS shows all the default values for all text indexes in the database. For example, the default values of CODEPAGE, LANGUAGE, FORMAT and so on. SYSIBMTS.TSSERVERS displays the text search server information like host name, port number, token and so on. SYSIBMTS.TSLOCKS shows the command lock information at the database and index level.

The Index-level views are SYSIBMTS.TSINDEXES which shows details about the text indexes and their information in the database. The SYSIBMTS.TSCONFIGURATION, displays the values of index configuration parameters for the indexes defined in the database. SYSIBMTS.TSCOLLECTIONNAMES shows the collection name of each text index. SYSIBMTS.TSEVENT_nnnnnn also known as event view, provides historical information of indexing status along with any errors and warnings returned for each document. SYSIBMTS.TSSTAGING_nnnnnn also known as staging table, captures the changes that happened on the base table, and nnnnnn is the index identifier which is generated automatically.



Text indexes for partitioned tables

- Text-maintained staging tables enable additional scenarios
  - Attaching and detaching a range
  - LOAD Insert
- Combination of DPF, RP, MDC supported

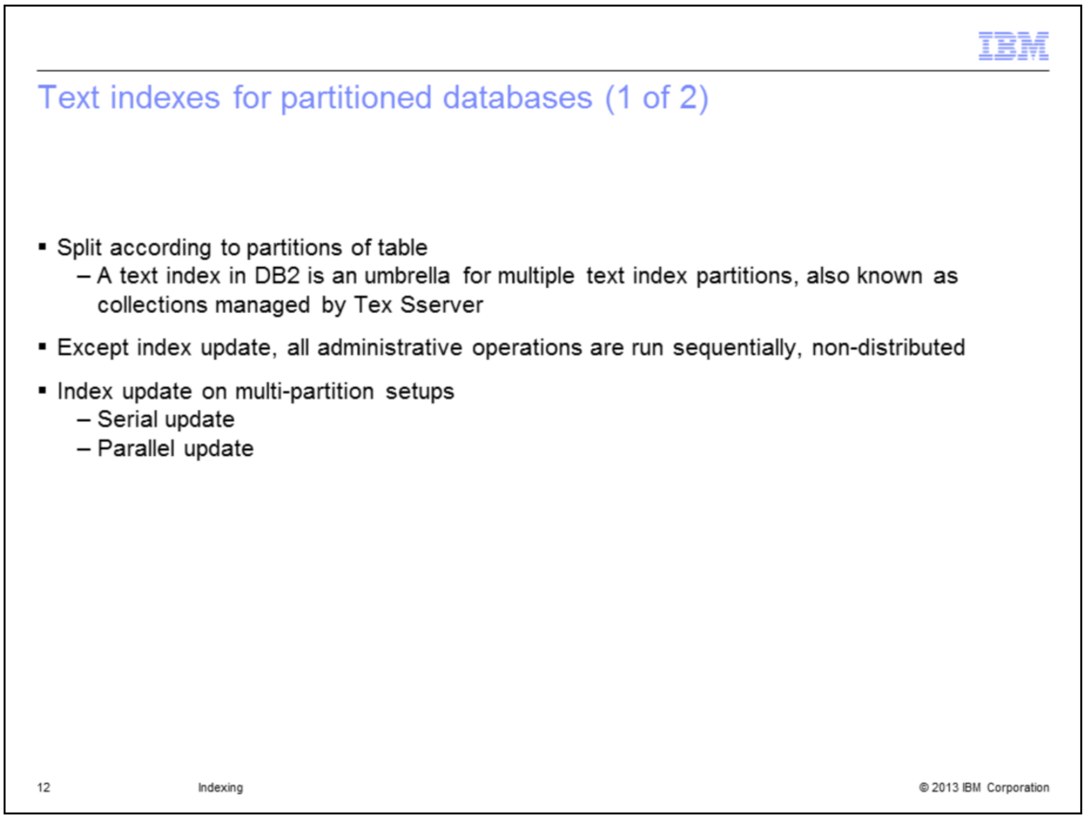


11 Indexing © 2013 IBM Corporation

The basic method of indexing using triggers does not support the scenarios like populating table with LOAD command and attaching and detaching ranges of range partitioned tables. Hence, there is a need to enable an additional staging infrastructure to capture these changes. The information about new and deleted documents is captured in a text maintained auxiliary table using this infrastructure. Whereas, the information about updated documents is captured in the staging, or primary log, table.

The combination of range-partitioned tables or tables that use the multidimensional clustering feature in a single-partition or partitioned database environment is supported.

The text-maintained staging infrastructure uses integrity processing to recognize new and deleted data and can be applied to partitioned and non-partitioned tables.



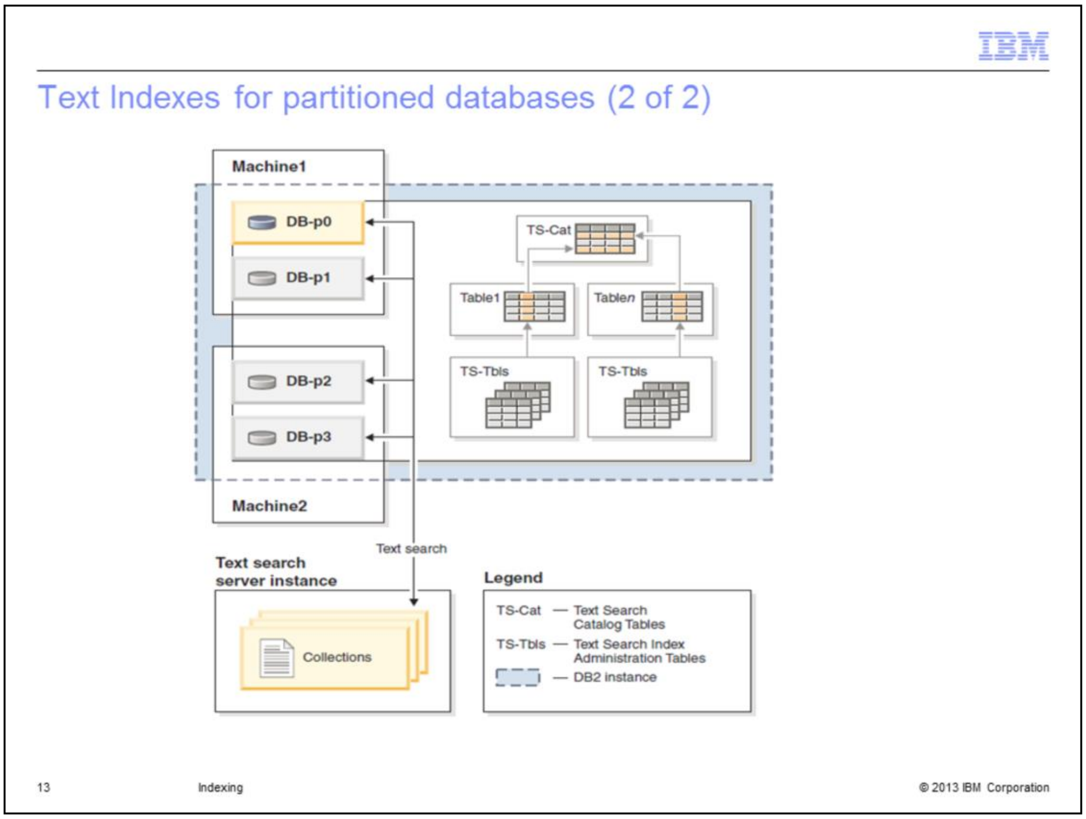
Text indexes for partitioned databases (1 of 2)

- Split according to partitions of table
  - A text index in DB2 is an umbrella for multiple text index partitions, also known as collections managed by Tex Sserver
- Except index update, all administrative operations are run sequentially, non-distributed
- Index update on multi-partition setups
  - Serial update
  - Parallel update

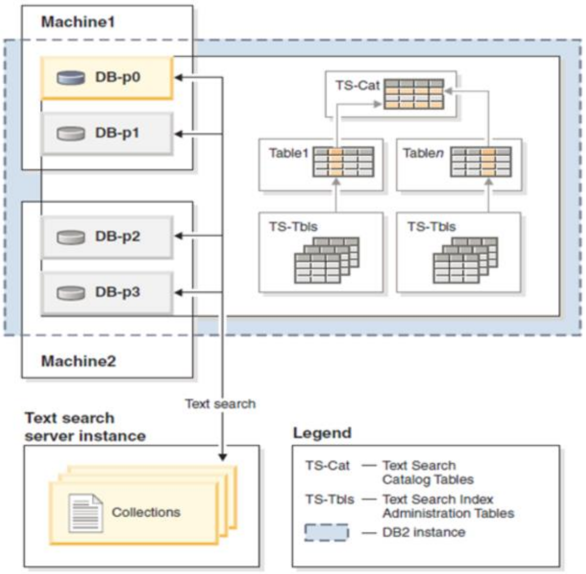
12 Indexing © 2013 IBM Corporation

In a database partitioned environment, text index is also partitioned according to the partitions of the table. For each text index partition, there is a collection created. Except index update, all administrative operations are started sequentially, non-distributed.

The update processing for a partitioned text search index can be run in parallel or in serial mode. In parallel mode, the index update is distributed to the database partitions and run independently on each node. In serial mode, the index update is run without distribution and stops when a failure is encountered. Serial mode indexing typically takes longer but requires less resources.

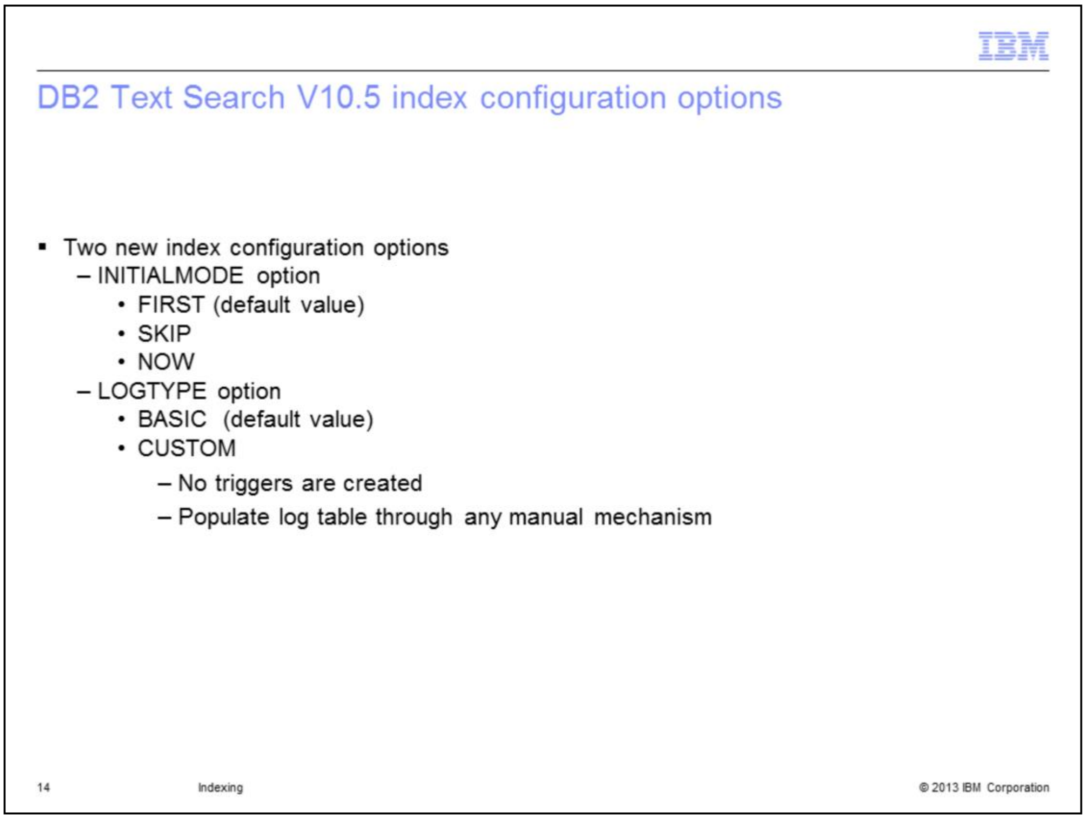


Text Indexes for partitioned databases (2 of 2)



13 Indexing © 2013 IBM Corporation

The figure displayed on this slide depicts a DB2 Text Search instance with four partitions on two physical machines – Machine1 and Machine2 with two logical partitions per host. There is only a single DB2 Text Search server serving all the database partitions.



IBM

DB2 Text Search V10.5 index configuration options

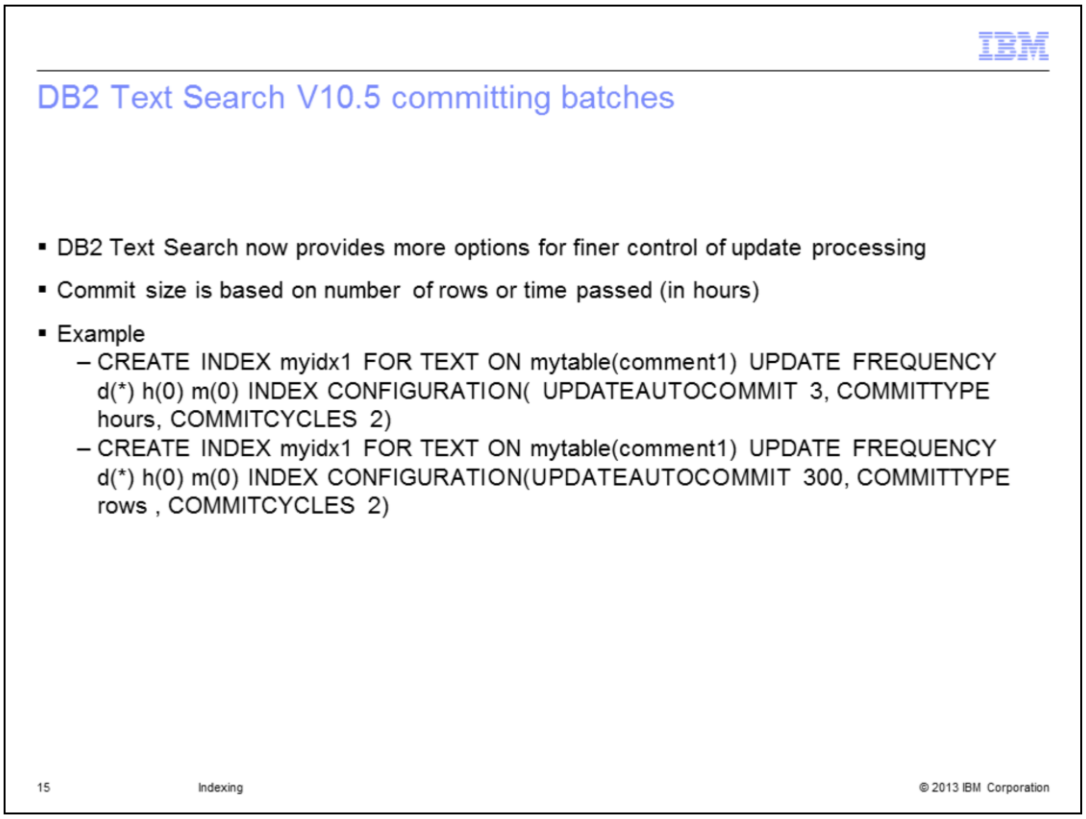
- Two new index configuration options
  - INITIALMODE option
    - FIRST (default value)
    - SKIP
    - NOW
  - LOGTYPE option
    - BASIC (default value)
    - CUSTOM
      - No triggers are created
      - Populate log table through any manual mechanism

14 Indexing © 2013 IBM Corporation

DB2 Text Search has provided two new index configuration options, INITIALMODE and LOGTYPE, to control update index processing.

Valid values for the INITIALMODE option are FIRST, SKIP, and NOW. FIRST is the default value. The initial update happens on the first update. SKIP means an incremental update is activated and provides the first update. NOW is when the initial update is started at the end of creating the index.

Valid values for the LOGTYPE option are BASIC and CUSTOM. BASIC is the default value. When using the CUSTOM value, no triggers are created. You should populate the log table through any manual mechanism. The BASIC option adds triggers to recognize changes.



IBM

DB2 Text Search V10.5 committing batches

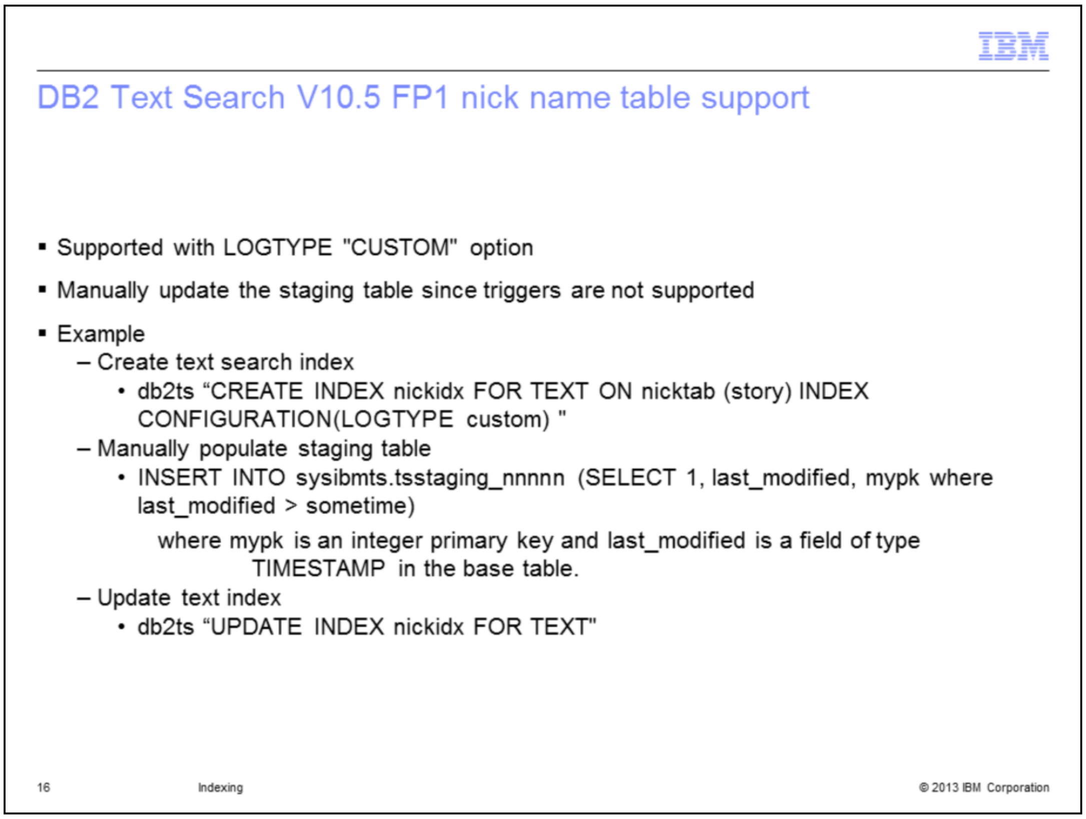
- DB2 Text Search now provides more options for finer control of update processing
- Commit size is based on number of rows or time passed (in hours)
- Example
  - CREATE INDEX myidx1 FOR TEXT ON mytable(comment1) UPDATE FREQUENCY d(*) h(0) m(0) INDEX CONFIGURATION(UPDATEAUTOCOMMIT 3, COMMITTYPE hours, COMMITCYCLES 2)
  - CREATE INDEX myidx1 FOR TEXT ON mytable(comment1) UPDATE FREQUENCY d(*) h(0) m(0) INDEX CONFIGURATION(UPDATEAUTOCOMMIT 300, COMMITTYPE rows , COMMITCYCLES 2)

15 Indexing © 2013 IBM Corporation

Text index update processing already provides a feature to specify the commit size by way of the updateautocommit argument. To provide further control, additional settings are available now with V10.5 to determine whether the commit size should be treated as rows or hours, and how many batches to process.

The sample examples provided on this slide display how to create an index using the new options available with V10.5.

The first example creates an index with a three hour commit cycle for a six hour time window. The second example creates an index with a total of two commit cycles committing 300 rows each.



DB2 Text Search V10.5 FP1 nick name table support

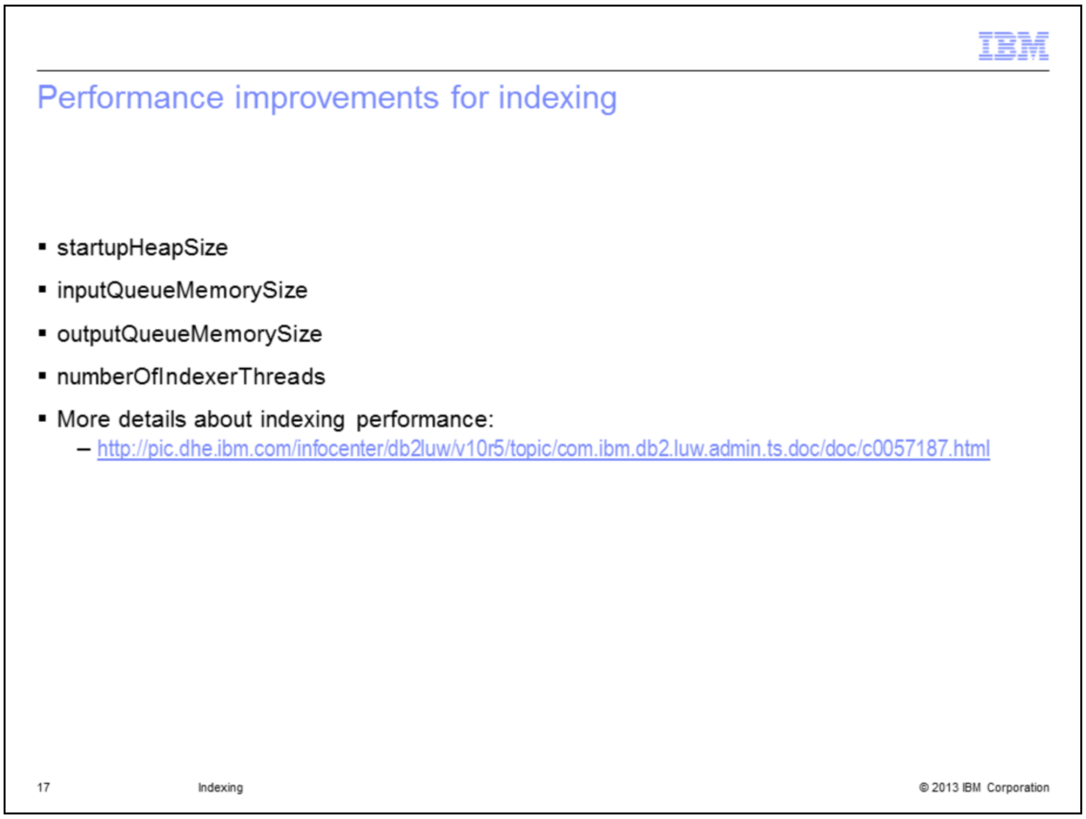
- Supported with LOGTYPE "CUSTOM" option
- Manually update the staging table since triggers are not supported
- Example
  - Create text search index
    - db2ts "CREATE INDEX nickidx FOR TEXT ON nicktab (story) INDEX CONFIGURATION(LOGTYPE custom) "
  - Manually populate staging table
    - INSERT INTO sysibmts.tsstaging_nnnnn (SELECT 1, last_modified, mypk where last_modified > sometime)

      where mypk is an integer primary key and last_modified is a field of type TIMESTAMP in the base table.
  - Update text index
    - db2ts "UPDATE INDEX nickidx FOR TEXT"

16 Indexing © 2013 IBM Corporation

DB2 Text Search supports creation of text indexes on nickname tables for a federated database since V10.5 FP1. Only the LOGTYPE index configuration parameter with a value of "CUSTOM" is supported. Triggers cannot be created on nicknames, hence, the staging table must be populated manually.

A sample usage scenario is provided on this slide for setting up the text index on a nickname table.



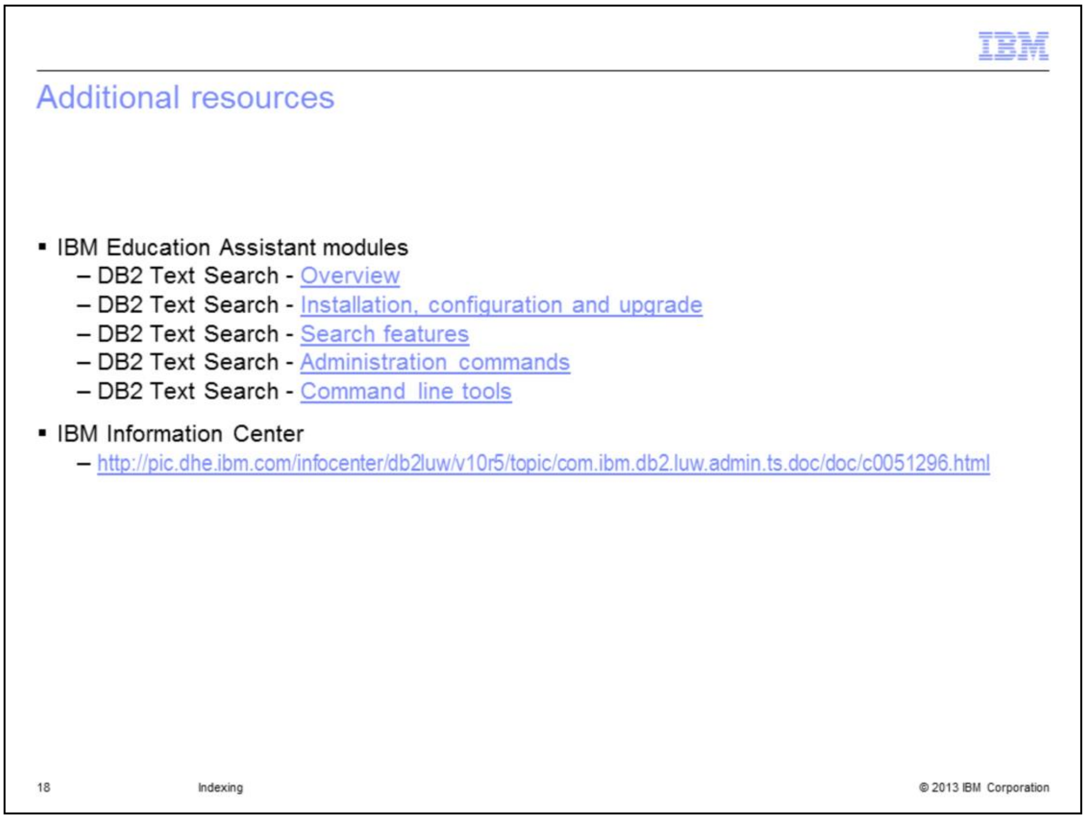
Performance improvements for indexing

- startupHeapSize
- inputQueueMemorySize
- outputQueueMemorySize
- numberOfIndexerThreads
- More details about indexing performance:
  - http://pic.dhe.ibm.com/infocenter/db2luw/v10r5/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0057187.html

17 Indexing © 2013 IBM Corporation

There are several configuration parameters available including heap size, queue size, number of indexing threads, and so on to achieve optimal indexing performance.

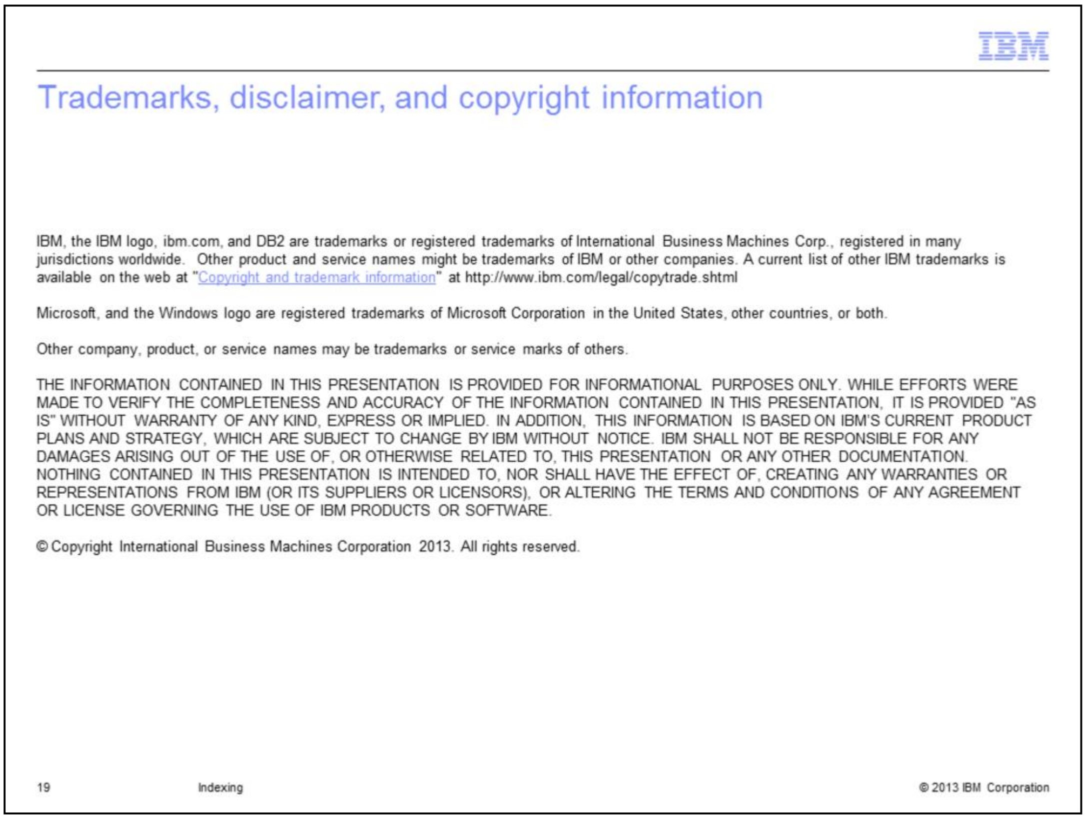
For more details visit the Information Center. The link is displayed on this slide.


Additional resources

- IBM Education Assistant modules
  - DB2 Text Search - Overview
  - DB2 Text Search - Installation, configuration and upgrade
  - DB2 Text Search - Search features
  - DB2 Text Search - Administration commands
  - 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

18 Indexing © 2013 IBM Corporation

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