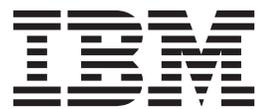


IBM DB2 10.1  
для Linux, UNIX и Windows

*Руководство по текстовому  
поиску*

*Обновлено в январе 2013*

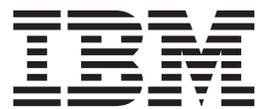




IBM DB2 10.1  
для Linux, UNIX и Windows

*Руководство по текстовому  
поиску*

*Обновлено в январе 2013*



**Замечание**

Перед использованием продукта и этой информации по нему, прочтите общую информацию в разделе Приложение Е, “Замечания”, на стр. 261.

**Замечание по изданию**

Информация в этом документе является собственностью IBM. Она предоставляется в соответствии с лицензионным соглашением и защищена законами об авторском праве. Информация, содержащаяся в этой публикации, не содержит никаких гарантий, и никакое утверждение в этом руководстве не должно рассматриваться как гарантия.

Заказать публикации IBM можно через Интернет или через вашего местного представителя IBM.

- Чтобы заказать публикации через Интернет, откройте Центр публикаций IBM по адресу <http://www.ibm.com/shop/publications/order>
- Чтобы найти вашего местного представителя IBM, откройте страницу IBM Directory of Worldwide Contacts по адресу <http://www.ibm.com/planetwide/>

Чтобы заказать публикации DB2 publications в отделе DB2 Marketing and Sales в Соединенных Штатах или Канаде, позвоните по телефону 1-800-IBM-4YOU (426-4968).

Посылая информацию IBM, вы даете IBM неисключительное право использовать или распространять эту информацию тем способом, каким компания сочтет нужным, без каких-либо обязательств перед вами.

# Содержание

## Глава 1. Текстовый поиск DB2 (DB2 Text Search) . . . . . 1

## Глава 2. Ключевые возможности и понятия DB2 Text Search . . . . . 3

Сценарии внедрения текстового поиска DB2 . . . . .	4
Создание, обновление и изменение свойств индексов текстового поиска . . . . .	7
Текстовый поиск DB2 в среде многораздельных баз данных . . . . .	10
Инкрементные обновления для индексов текстового поиска DB2 . . . . .	12
Лингвистическая обработка для DB2 Text Search . . . . .	14
Сценарий: Индексирование и поиск . . . . .	15
Поддержка документов формата Rich Text Format (RTF) и собственных форматов . . . . .	18

## Глава 3. Планирование решений текстового поиска . . . . . 19

Характеристики документа . . . . .	19
Форматы документов, поддерживаемые для DB2 Text Search . . . . .	19
Поддерживаемые типы данных . . . . .	19
Преобразование неподдерживаемых форматов и типов данных . . . . .	19
Поддерживаемые языки и кодовые страницы . . . . .	20
О размерах документов . . . . .	20
Обзор защиты DB2 Text Search . . . . .	21
Роли пользователей . . . . .	23
Политики доступа и защита связи . . . . .	24
Планирование мощности и оптимизация DB2 Text Search . . . . .	25
Конфигурация сервера текстового поиска DB2 . . . . .	26
Планирование и оптимизация индексов DB2 Text Search . . . . .	30
Настройка системы DB2 Text Search . . . . .	35
Планирование запросов DB2 Text Search . . . . .	36
Аргументы текстового поиска DB2 (DB2 Text Search) . . . . .	36
Множественные предикаты DB2 Text Search . . . . .	37
Локаль и язык текстового поиска DB2 . . . . .	38
Функция SCORE DB2 Text Search . . . . .	38
Функция RESULTLIMIT DB2 Text Search . . . . .	39
Конфигурирование синтаксического анализа для DB2 Text Search . . . . .	39
Пространства имен XML DB2 Text Search . . . . .	40

## Глава 4. Установка и конфигурирование DB2 Text Search. . . 43

Требования к программному и аппаратному обеспечению для DB2 Text Search . . . . .	46
Установка DB2 Text Search с конфигурацией по умолчанию . . . . .	48

Установка и конфигурирование текстового поиска DB2 при помощи мастера по установке DB2 . . . . .	48
Установка и конфигурирование DB2 Text Search при помощи файла ответов . . . . .	48
Установка текстового поиска DB2 при помощи db2_install (Linux и UNIX) . . . . .	49
Установка DB2 Text Search без начального конфигурирования . . . . .	50
Установка серверов баз данных DB2 при помощи мастера по установке DB2 (Windows) . . . . .	50
Установка серверов DB2 при помощи мастера по установке DB2 (Linux и UNIX) . . . . .	53
Обзор установки DB2 с помощью файла ответов (Windows) . . . . .	56
Обзор установки DB2 с помощью файла ответов (Linux и UNIX) . . . . .	57
Установка и конфигурирование автономного сервера текстового поиска . . . . .	58
Требования к памяти для установки автономного сервера . . . . .	58
Установка автономного сервера текстового поиска DB2 . . . . .	58
Установка и конфигурирование автономного сервера в качестве службы Windows . . . . .	59
Деинсталляция автономного сервера текстового поиска DB2 . . . . .	60

## Глава 5. Конфигурирование текстового поиска DB2 . . . . . 61

Начальное конфигурирование интегрированного сервера текстового поиска DB2 . . . . .	63
Обновление информации о сервере текстового поиска DB2 . . . . .	64
Конфигурирование автономного сервера DB2 Text Search . . . . .	66
Обновление файла служб на сервере для соединений TCP/IP . . . . .	67
Установка DB2 Accessories Suite для DB2 Text Search . . . . .	68
Деинсталляция DB2 Accessories Suite для DB2 Text Search . . . . .	69

## Глава 6. Обновление текстового поиска DB2 . . . . . 71

Обновление DB2 Text Search для установок от имени администратора или пользователя root . . . . .	71
Обновление установки DB2 Text Search от имени рядового пользователя (не root) (Linux и UNIX) . . . . .	74
Обновление многораздельного экземпляра без возможности текстового поиска DB2 . . . . .	76
Обновление автономного сервера текстового поиска DB2 . . . . .	76

## Глава 7. Конфигурирование и администрирование индексов текстового поиска . . . . . 79

Инструменты командной строки для DB2 Text Search	79
Вызов команд текстового поиска	79
Поддержка документов формата Rich Text Format (RTF) и собственных форматов	80
Включение поддержки документов в расширенном текстовом формате для DB2 Text Search	80
Отключение поддержки формата RTF и собственных форматов	81
Запуск службы экземпляра текстового поиска DB2	81
Остановка службы экземпляра текстового поиска DB2	82
Включение базы данных для DB2 Text Search	82
Выключение базы данных для DB2 Text Search	84
Удаление бесхозных собраний текстового поиска DB2	84
Словари синонимов для DB2 Text Search	86
Добавление словаря синонимов для DB2 Text Search	86
Удаление словаря синонимов для DB2 Text Search	87
Создание индексов текстового поиска	88
Создание индекса текстового поиска	88
Обслуживание индексов текстового поиска	96
Обновление индекса текстового поиска	98
Очистка событий индекса текстового поиска	101
Изменение индекса текстового поиска	102
Просмотр состояния индексов текстового поиска	103
Изменение положения собрания текстового поиска DB2	103
Резервное копирование и восстановление индексов текстового поиска	104
Отбрасывание индекса текстового поиска	104
Пример: Планирование обновления индекса текстового поиска DB2	106

## Глава 8. Поиск с использованием индексов текстового поиска . . . . . 107

Функции поиска для DB2 Text Search	107
Способы полнотекстового поиска	108
Базовый поиск	109
Поиск неполных соответствий	109
Контекстный поиск	110
Поиск специальных символов	111
Структурный полнотекстовый поиск в документах XML	114
Поиск в индексах тестового поиска при помощи SCORE	116
Синтаксис аргументов DB2 Text Search	117
Синтаксис поиска для документов XML	121
Повышение производительности для полнотекстовых запросов	125

## Глава 9. Встроенные функции поиска SQL и XML . . . . . 127

функция CONTAINS	127
функция SCORE	129
Функция xmlcolumn-contains	132

## Глава 10. Команды управления для DB2 Text Search . . . . . 137

db2ts ALTER INDEX	138
db2ts CLEANUP FOR TEXT	142
db2ts CLEAR COMMAND LOCKS	143
db2ts CLEAR EVENTS FOR TEXT	145
db2ts CREATE INDEX	146
db2ts DISABLE DATABASE FOR TEXT	154
db2ts DROP INDEX	156
db2ts ENABLE DATABASE FOR TEXT	157
db2ts HELP	159
Команда db2ts RESET PENDING	161
db2ts START FOR TEXT	162
db2ts STOP FOR TEXT	164
db2ts UPDATE INDEX	165

## Глава 11. Хранимые процедуры DB2 Text Search . . . . . 169

Процедура SYSTS_ADMIN_CMD - Запуск команд управления текстовым поиском	169
Процедура SYSTS_ALTER - Изменение характеристик обновления индекса	171
Процедура SYSTS_CLEANUP - Удаление недопустимых индексов текстового поиска	177
Процедура SYSTS_CLEAR_COMMANDLOCKS - Удаление блокировок команд для индексов текстового поиска	178
Процедура SYSTS_CLEAR_EVENTS - Удаление событий индексации из таблицы событий индекса	180
Процедура SYSTS_CONFIGURE - Конфигурирование текущей базы данных для текстового поиска	183
Процедура SYSTS_CREATE - Создание индекса текстового поиска для столбца	185
Процедура SYSTS_DISABLE - Выключение текущей базы данных для текстового поиска	195
Процедура SYSTS_DROP - Отбрасывание индекса текстового поиска	197
Процедура SYSTS_ENABLE - Включение текущей базы данных для текстового поиска	199
Процедура SYSTS_UPDATE - Обновление индекса текстового поиска	201
Процедура SYSTS_UPGRADE_CATALOG - Обновление каталога текстового поиска	204
SYSTS_UPGRADE_INDEX - Обновление индексов текстового поиска	206

## Глава 12. Управляющие производные таблицы текстового поиска . . . . . 209

Производная таблица SYSIBMTS.TSDEFAULTS	209
Производная таблица SYSIBMTS.TSLOCKS	210
Производная таблица SYSIBMTS.TSSERVERS	211
Производная таблица SYSIBMTS.TSINDEXES	212
Производная таблица SYSIBMTS.TSCONFIGURATION	214
SYSIBMTS.TSCOLLECTIONNAMES view	215
Производная таблица SYSIBMTS.TSEVENT	215
Производная таблица SYSIBMTS.TSSTAGING	216

**Приложение А. Сравнение DB2 Text Search и Net Search Extender . . . . 219**

**Приложение В. Локали, поддерживаемые DB2 Text Search. . 225**

**Приложение С. Команды DB2. . . . 227**

db2iupgrade - обновление экземпляра . . . . . 227  
db2icrt - Создание экземпляра . . . . . 230  
db2idrop - удаление экземпляра . . . . . 239  
db2iupdt - Обновление экземпляров . . . . . 241

**Приложение D. Обзор технической информации DB2 . . . . . 251**

Техническая библиотека DB2 в печатном виде или в формате PDF . . . . . 251

Вызов справки по SQLSTATE из командной строки . . . . . 254  
Доступ к различным версиям Информационного центра DB2 . . . . . 254  
Изменение Информационного центра DB2, установленного на вашем компьютере или на сервере интранета . . . . . 255  
Обновление Информационного центра DB2, установленного на вашем компьютере или на сервере интранет, вручную . . . . . 256  
Учебные материалы DB2. . . . . 258  
Информация об устранении неисправностей DB2 . . . . . 258  
Положения и условия . . . . . 259

**Приложение Е. Замечания . . . . . 261**

**Индекс . . . . . 265**



---

## Глава 1. Текстовый поиск DB2 (DB2 Text Search)

DB2 Text Search позволяет выполнять поиск по текстовым столбцам, применяя операторы SQL и XQuery для выполнения запросов текстового поиска данных, хранящихся в базе данных DB2.

DB2 Text Search предоставляет большие возможности для поиска данных в текстовых столбцах, хранящихся в таблице DB2. Поисковая система обеспечивает быстрый отклик на запрос и консолидированный и ранжированный набор результатов, что позволяет вам легко и без промедления найти необходимую информацию. При помощи включения функций DB2 Text Search в состав ваших операторов SQL и XQuery можно создать мощные и многоцелевые программы поиска текстов. Кроме того, механизм поиска использует лингвистический анализ, чтобы обеспечить возврат только подходящих результатов запроса. Включив поддержку текстового поиска, можно использовать функции CONTAINS, SCORE и xmlcolumn-contains, встроенные в механизм DB2, для поиска по индексам тестового поиска на основе заданных аргументов поиска.

DB2 Text Search достигает высокой производительности и масштабируемости путем использования потокового режима передачи данных, что исключает высокие ресурсные затраты во время поиска.

Для интеграции с текстовым поиском можно установить в той же системе сервер DB2 Text Search и серверы базы данных DB2. Можно также установить сервер DB2 Text Search и серверы базы данных DB2 в других системах для автономной работы. Сервер DB2 Text Search работает на своей собственной виртуальной Java-машине (Java Virtual Machine, JVM). Службы DB2 Text Search запускаются и останавливаются явным образом после запуска экземпляра DB2.

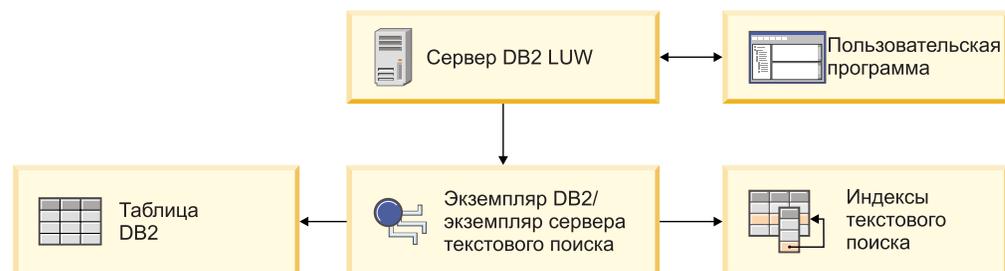


Рисунок 1. Схема внедрения для интегрированного сервера текстового поиска DB2

У DB2 Text Search нет графического пользовательского интерфейса. Вместо этого для таких задач, как конфигурирование и администрирование сервера DB2 Text Search, создание словаря синонимов для собрания и диагностика проблем, доступны инструменты командной строки. Кроме того, можно использовать интерфейс хранимых процедур для различных распространенных задач администрирования.

Вы можете выполнить перенастройку Net Search Extender до DB2 Text Search: создайте и обновите индексы текстового поиска DB2, а затем, когда индексы будут готовы к использованию, переключите индексы в нужное состояние. Подробности смотрите в разделе о перенастройке Net Search Extender до DB2 Text Search.

DB2 Text Search включает в себя следующие основные возможности:

### **Тесную интеграцию с DB2 для Linux, UNIX и Windows**

- Интерфейс хранимых процедур для команд администрирования
- Выполняемые установщиком DB2 установку и конфигурирование
- Скрытая аутентификация
- Коды SQL для обработки ошибок

### **Индексацию документов**

- Быструю индексацию больших объемов данных
- Поддержку pureXML
- Поддержку многих форматов документов
- Инкрементное и асинхронное изменение индексов

### **Технологию расширенного поиска**

- Поддержку SQL, SQL/XML и XQuery
- Функции SQL CONTAINS и SCORE
- Встроенные функциональные возможности SQL в сочетании с DB2 Optimizer
- Функцию XML xmlcolumn-contains
- Фильтрацию XML
- Лингвистическую обработку на всех поддерживаемых языках
- Поддержку весовых коэффициентов, символов подстановки и необязательных термов
- Поддержку словаря синонимов

## Глава 2. Ключевые возможности и понятия DB2 Text Search

DB2 Text Search предлагает быстрый и универсальный метод для поиска в текстовых документах, сохраненных в текстовом столбце баз данных DB2. При поиске можно использовать запросы SQL; при поиске в документах XML можно использовать запросы XQuery.

Необходима возможность уникальной идентификации текстовых документов. DB2 Text Search использует для этого первичный ключ таблицы.

Вместо последовательного поиска в текстовых документах DB2 Text Search использует более эффективный подход, поиск при помощи *индекса текстового поиска*. Индекс текстового поиска состоит из значимых термов, которые извлекаются из текстовых документов.

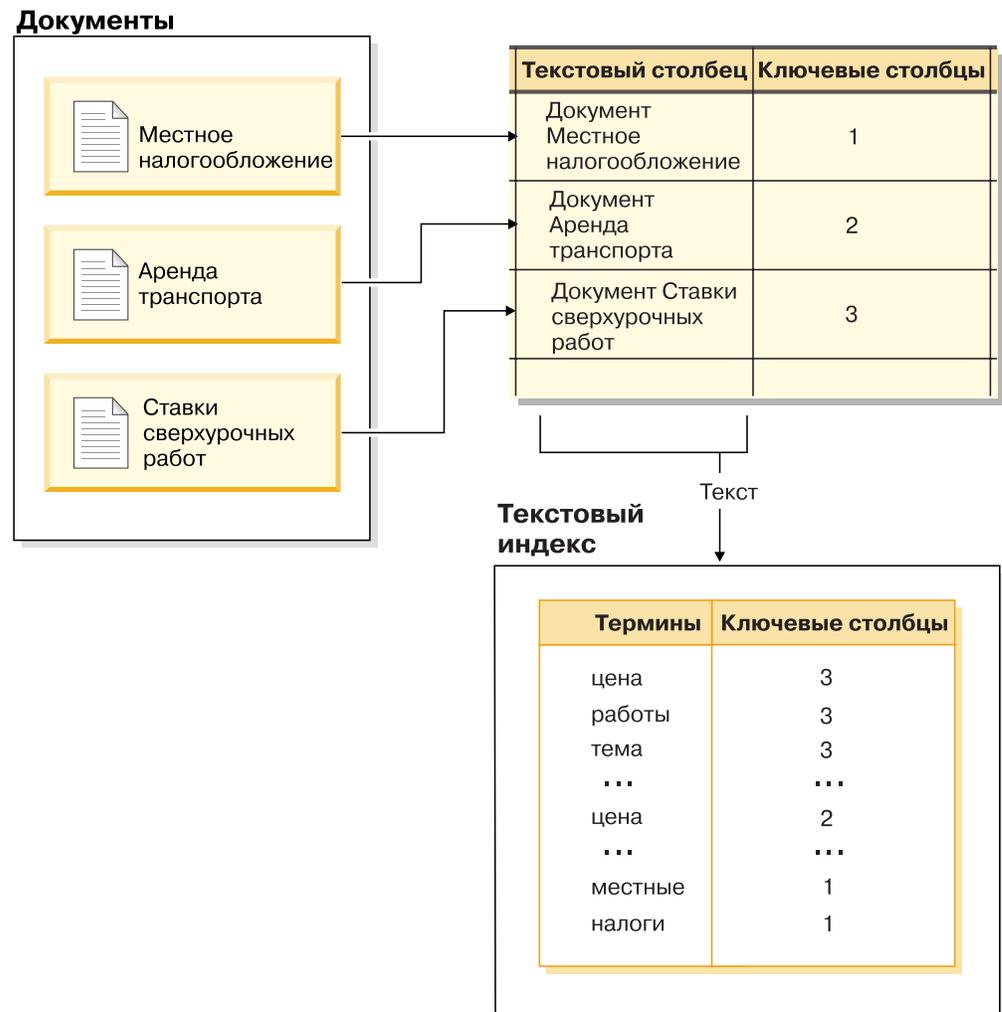


Рисунок 2. Создание индекса текстового поиска

Операция создания индекса текстового поиска определяет свойства индекса, в частности, частоту обновления. Сразу после создания индекс текстового поиска не

содержит никаких данных. При обновлении в индекс текстового поиска добавляются данные о терминах и текстовых документах. При начальном обновлении индекса все текстовые документы добавляются из текстового столбца в индекс. Последующие обновления, так называемые инкрементные обновления, синхронизируют данные в таблице и данные в индексе текстового поиска. DB2 Text Search обеспечивает два метода для синхронизации индекса текстового поиска с таблицей:

- Основной метод синхронизации использует триггеры, которые автоматически сохраняют информацию о новых, измененных, и удаленных документах в промежуточной таблице.
- Расширенный метод синхронизации использует триггер, который сохраняет в промежуточной таблице только информацию об измененных документах, а информация о новых и удаленных документах захватывается через обработку целостности и сохраняется в дополнительной промежуточной таблице.

Подробности смотрите в разделе о создании, обновлении и изменении свойств индексов текстового поиска.

DB2 Text Search работает путем сбора данных из различных источников и их индексирования для последующего быстрого извлечения. DB2 Text Search использует лингвистический анализ для улучшения результатов поиска и поддерживает следующие форматы документов:

- Неструктурированный простой текст.
- Структурированный текст, такой как в документах HTML или XML
- Собственные форматы документов, такие как в документах PDF или Microsoft Office.

Для работы с собственными форматами может потребоваться установка и конфигурирование фильтров.

Текстовый поиск DB2 поддерживает полнотекстовый поиск в среде многораздельных баз данных. Кроме того, можно создать индекс текстового поиска для таблиц с разделением по диапазонам или таблиц с многомерной кластеризацией в среде однораздельных или многораздельных баз данных. Индексы текстового поиска поддерживаются для всех сочетаний возможностей разделения. В среде многораздельных баз данных индекс текстового поиска разделяется в соответствии с разделением таблицы по нескольким разделам базы данных. Другие возможности разделения, такие как разделение таблиц или многомерная кластеризация, не влияют на разделение индекса текстового поиска.

Кроме того, DB2 Text Search поддерживает как интегрированную, так и автономную установку. Конфигурация с автономным сервером DB2 Text Search предпочтительней для многораздельных сред, поскольку предотвращает конкуренцию с сервером баз данных за общие ресурсы. Текстовый поиск DB2 не поддерживается в среде DB2 pureScale.

---

## Сценарии внедрения текстового поиска DB2

DB2 Text Search поддерживает как интегрированную установку сервера текстового поиска, так и автономную, отдельно от продукта базы данных DB2. Автономный сервер текстового поиска (другое название - сервер текстового поиска Enterprise Content Management, ECM) можно установить и управлять им на поддерживаемых платформах хоста. DB2 Text Search не поддерживается возможностью HADR (High availability disaster recovery - Высокая доступность - восстановление после аварий).

Экземпляр базы данных DB2 использует TCP/IP для связи с автономным сервером DB2 Text Search. Поддержка SSL или GSKit недоступна. Однако каналы шифрования

могут использоваться через программу **stunnel** или туннелирование SSH. Ограничьте доступ к своему репозиторию документов и файлам индексов текстового поиска в зависимости от ваших требований к защите. Автономный сервер текстового поиска следует устанавливать на компьютерах с защищенным сетевым соединением за брандмауэром, чтобы предотвратить несанкционированный доступ к индексам текстового поиска. Нужно так сконфигурировать ограничения доступа TCP/IP к автономному серверу текстового поиска, чтобы доступ к нему оставался только у хоста, где установлен сервер базы данных.

Ниже приведены иллюстрации, показывающие на высоком уровне внедрение сервера DB2 Text Search, включая интегрированную и автономную установку. Можно установить и сконфигурировать интегрированный сервер DB2 Text Search, а позднее переключиться на автономный сервер. Однако нет поддержки автоматического перемещения индексов текстового поиска на другой сервер текстового поиска. Поэтому при некоторых установках возникает необходимость отбросить существующие текстовые индексы, когда нужно назначить новый сервер текстового поиска для экземпляра базы данных.

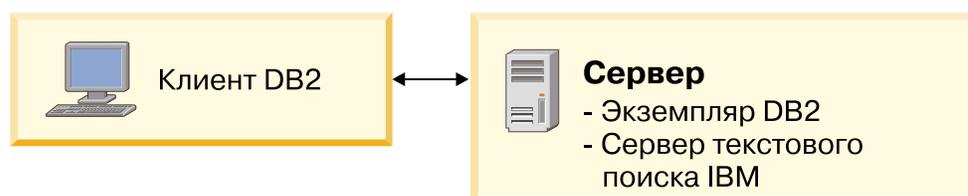


Рисунок 3. Установка интегрированного сервера DB2 Text Search

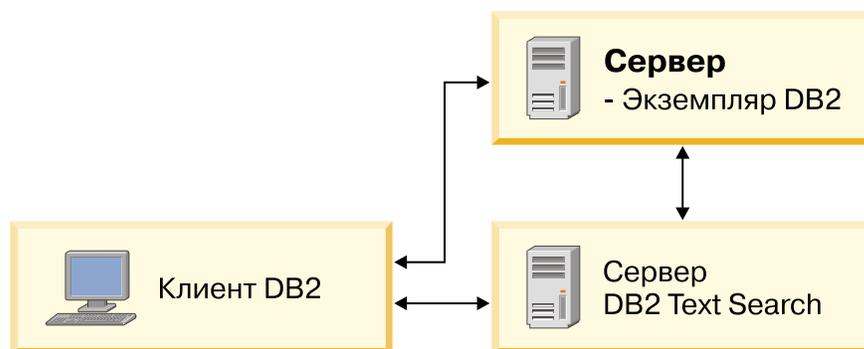


Рисунок 4. Установка автономного сервера DB2 Text Search

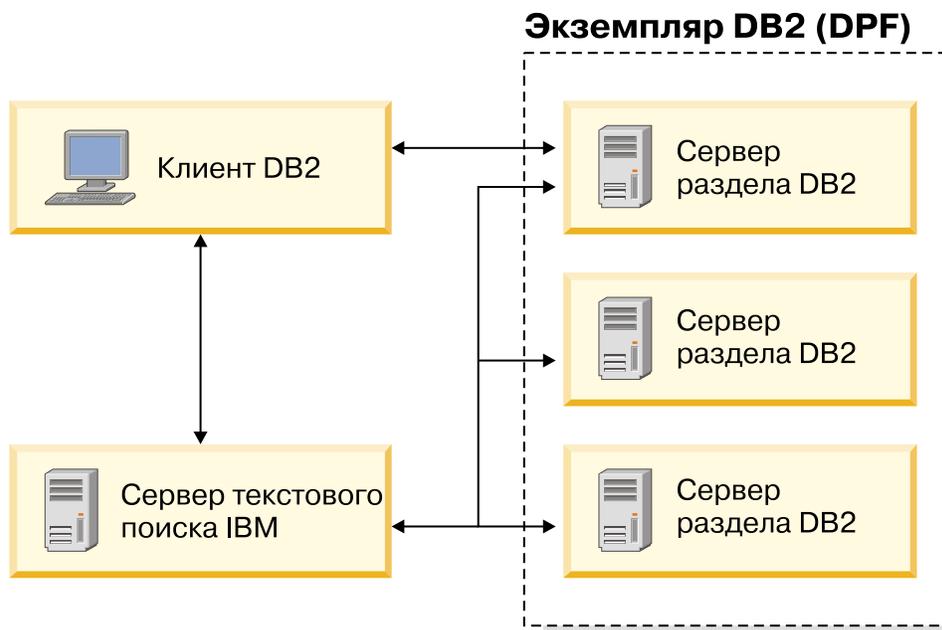


Рисунок 5. Установка автономного сервера DB2 Text Search в среде многораздельных баз данных

**Примечание:** Каталог установки DB2 Text Search зависит от типа внедрения.

- Для интегрированного сервера:
  - `<TS_HOME>` представляет путь `../sql1lib/db2tss` в операционных системах Windows, Linux или UNIX.
- При автономной установке `<ECMTS_HOME>` представляет положение установки сервера текстового поиска.
  - По умолчанию `<ECMTS_HOME>` представляет путь `/opt/ibm/ECMTextSearch` в системах Linux или UNIX.
  - По умолчанию `<ECMTS_HOME>` представляет путь `C:\Program Files\IBM\ECMTextSearch` в системах Windows.

Внедрение автономного сервера текстового поиска имеет смысл по следующим соображениям:

- управление защитой: автономный сервер Text Search дает возможность определить владельца процесса текстового сервера, отличного от владельца экземпляра базы данных.
- управление рабочей нагрузкой: автономный сервер Text Search отделяет ресурсоемкую обработку текстовых поисков от задач сервера базы данных.

Каждый экземпляр базы данных связан с одним сервером Text Search. В среде многораздельной базы данных, содержащей несколько серверов разделов, автономная установка предотвращает концентрацию ресурсоемкой обработки на одном сервере раздела.

Автономный и интегрированный сервер Text различаются только начальной конфигурацией, прежде всего тем, что автономный сервер Text Search уже сконфигурирован для обработки документов формата Rich Text Format (RTF) и собственных форматов.

---

## Создание, обновление и изменение свойств индексов текстового поиска

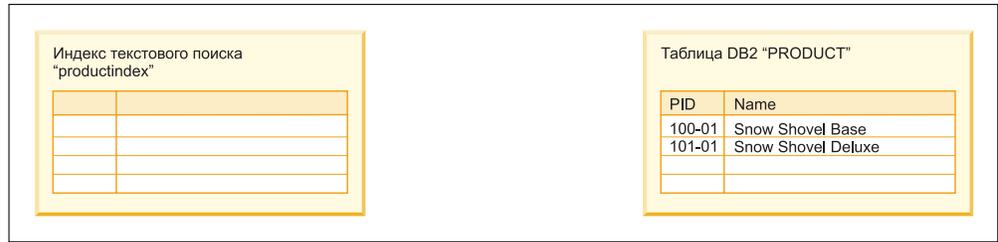
Создание индекса текстового поиска - это процесс определения свойств текстового индекса. После создания индекса текстового поиска надо обновить его, добавив данные из связанной с ним таблицы. Можно также изменить некоторые свойства индекса текстового поиска позже, например, параметры **UPDATE FREQUENCY** и **UPDATE MINIMUM**.

С помощью индекса текстового поиска можно выполнять поиск данных в текстовом столбце, используя функции текстового поиска. Индекс текстового поиска состоит из значимых термов, которые извлекаются из текстовых документов. Первичный ключ табличной строки используется в индексе для идентификации источника термов.

Непосредственно после создания индекс текстового поиска не содержит данных. Данные в индекс текстового поиска добавляются с помощью команды **db2ts UPDATE INDEX** или административной подпрограммы **SQL SYSTS\_UPDATE**. При первом обновлении индекса (другое название - *начальное обновление*) все текстовые документы добавляются в виде текстового столбца в индекс текстового поиска. При последующих обновлениях (другое название - *инкрементные обновления*) данные в базовой таблице синхронизируются с индексом текстового поиска.

В следующем примере пользователь создает индекс текстового поиска с именем **MYSCHEMA.PRODUCTINDEX** для таблицы **PRODUCT** из базы данных **SAMPLE**. Создание индекса текстового поиска и последующее выполнение начальных и инкрементных обновлений показывает, что индекс остается пустым до тех пор, пока пользователь не выполнит начальное обновление, и что по мере добавления данных в таблицу необходимо запускать инкрементное обновление, чтобы добавить новые данные в индекс текстового поиска.

### Создание индекса для текста



### Обновление текстового индекса (начальное обновление)



### Обновление текстового индекса (инкрементное обновление)

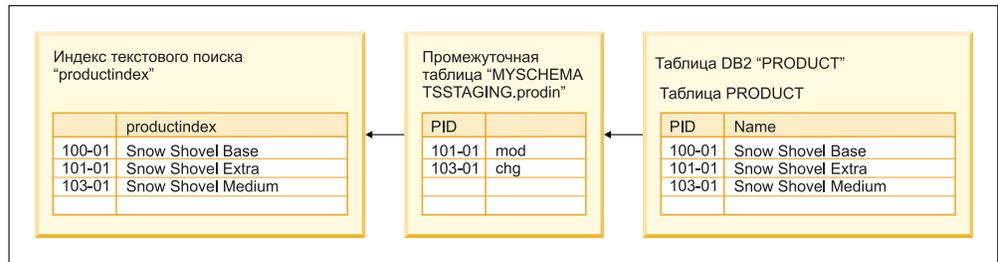


Рисунок 6. Создание индекса текстового поиска и последующее выполнение начальных и инкрементных обновлений

Текстовый поиск DB2 предусматривает два метода синхронизации текстового индекса с его таблицей:

- Основной метод синхронизации использует триггеры, которые автоматически сохраняют информацию о новых, измененных и удаленных документах в промежуточной таблице. Для каждого текстового индекса существует одна промежуточная таблица.

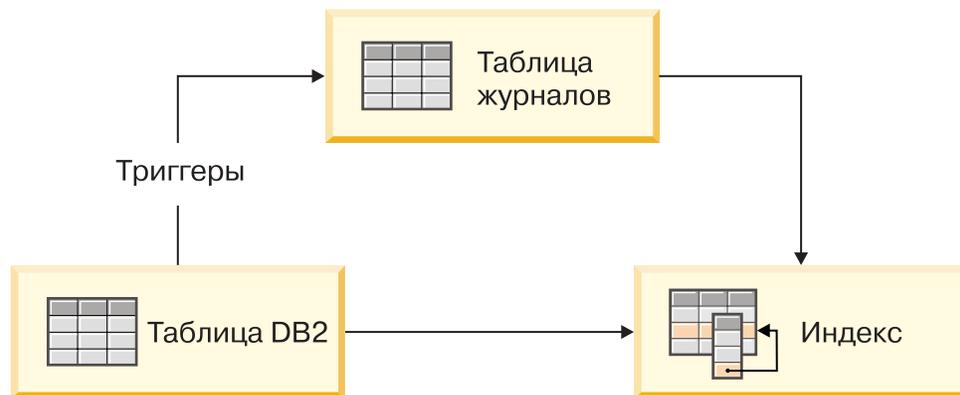


Рисунок 7. Инкрементное обновление при помощи триггеров

Поскольку основной метод использует только триггеры, обновления, не распознанные триггерами, игнорируются, например загрузка данных с помощью команды **LOAD** и подключение или отключение диапазонов таблицы с разделением по диапазонам.

- Расширенный метод синхронизации использует триггеры для хранения информации об измененных документах в промежуточной таблице, но он захватывает информацию о новых и удаленных документах посредством обработки целостности и сохраняет эту информацию в текстовой вспомогательной промежуточной таблице. При подключении раздела или загрузке данных необходимо затем запустить команду **SET INTEGRITY** для базовой таблицы, чтобы сделать доступными данные во вспомогательной промежуточной таблице. В случае, если раздел отключен, для промежуточной таблицы потребуется ввести команду **SET INTEGRITY** еще раз, чтобы сделать данные доступными для обработки. Другой вариант - с помощью команды **RESET PENDING** для базовой таблицы сделать данные доступными во всех ее вспомогательных промежуточных таблицах. При выполнении этой команды базовая таблица полностью доступна для операций чтения и записи. При отключении раздела необходимо запустить команду **RESET PENDING** для базовой таблицы или команду **SET INTEGRITY** для каждой из промежуточных таблиц.

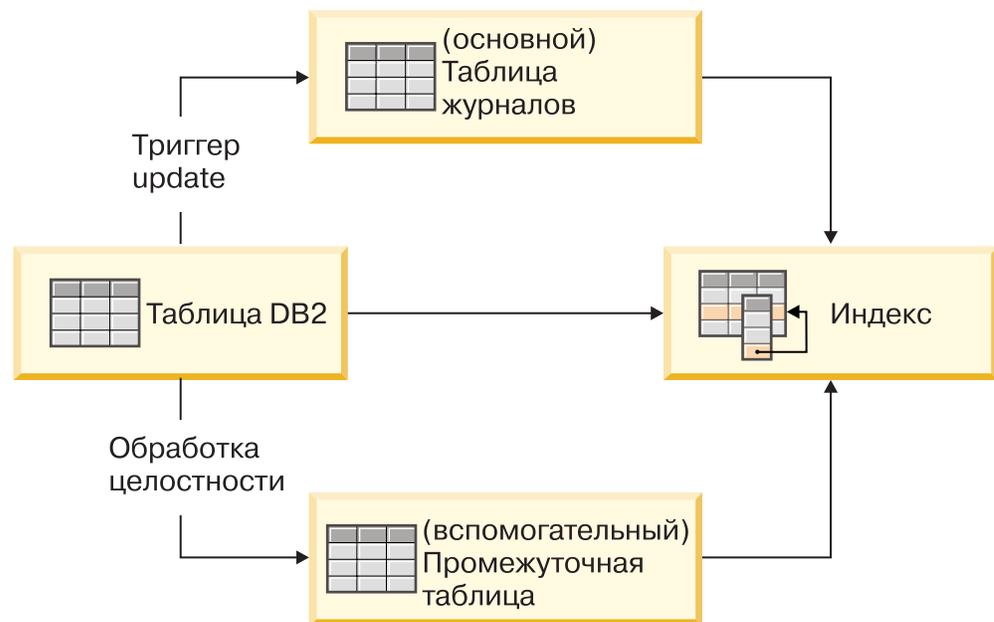


Рисунок 8. Инкрементное обновление с триггерами и обработкой целостности

Некоторые операции базы данных неявно или явно делают индекс текстового поиска недействительным. При явной недействительности состояние индекса текстового поиска в управляющей производной таблице `SYSIBMTS.TSINDEXES` будет `INDSTATUS='INVALID'`, например, если введена команда `ALTER DATABASE PARTITION GROUP`. Неявная недействительность наступает, когда изменения содержимого идут в обход промежуточного механизма, например, когда оператор `LOAD INSERT`, используется без расширенной промежуточной инфраструктуры. При неявной недействительности индекс текстового поиска не помечается как недействительный.

Вы можете обновить текстовый индекс, выбрав автоматическое обновление или обновление вручную. При автоматическом обновлении используется расписание, в котором указаны даты и время. Можно вручную обновить индекс текстового поиска,

запустив команду **UPDATE INDEX FOR TEXT** или процедуру **SYSPROC.SYSTS\_UPDATE**. Текстовый индекс поиска обновляется асинхронно, то есть вне транзакции, выполняющей вставку, изменение или удаление данных в базе данных. Асинхронная обработка обновления индекса текстового поиска улучшает пропускную способность и одновременность, поскольку при этом несколько обновлений могут объединяться в пакет и применяться к копии затронутых сегментов текстового индекса. Таким образом, у индекса текстового поиска блокируется только доступ для чтения на короткое время, пока сегменты обновленного индекса не будут возвращены на исходное место.

Индексы текстового поиска реорганизуются автоматически по мере необходимости, кроме того, можно явным образом запустить реорганизацию при помощи `adminTool` или воссоздать индекс с опцией `ALLROWS` при его обновлении.

---

## Текстовый поиск DB2 в среде многораздельных баз данных

Текстовый поиск DB2 поддерживает полнотекстовый поиск в среде многораздельных баз данных. Индексы текстового поиска распределяются по схеме, соответствующей базовым таблицам, для которых они создавались. Для каждого раздела базы данных создается раздел текстового индекса, так называемое собрание. Такая структура упрощает обслуживание текстового поиска, обеспечивая возможность параллельного обновления индекса на всех разделах.

Промежуточные таблицы, используемые для обновлений многораздельного индекса текстового поиска, создаются не для отдельных собраний, а для всего индекса, и распределяются аналогично базовой таблице. Промежуточные таблицы используют скалярную функцию `DBPARTITIONNUM`, чтобы для всех разделов индекса найти изменения, которые нужно применить к ним при обновлении индекса. Данные от каждого сервера раздела базы данных обновляются в соответствующих разделах текстового индекса во время обновления текстового индекса, что обеспечивает возможность параллельного выполнения обновления.

Каждое обновление текстового индекса может приводить к обновлению нескольких собраний, и это необходимо учесть при планировании ресурсов сервера текстового поиска. При распределении рабочей нагрузки в средах многораздельных баз данных рекомендуется конфигурация с автономным удаленным сервером текстового поиска.

Сервер DB2 Text Search, который устанавливается и конфигурируется отдельно от экземпляра DB2, называется автономной установкой. Удаленную автономную установку, то есть установку на хосте, отдельном от сервера базы данных, можно использовать для однораздельного и многораздельного экземпляров DB2, чтобы снять ресурсоемкую рабочую нагрузку сервера текстового поиска с хоста сервера базы данных.

Интегрированный сервер текстового поиска конфигурируется при создании экземпляра многораздельной базы данных по умолчанию на сервере раздела базы данных с наименьшим номером. Не обязательно конфигурировать сервер текстового поиска при установке, предусмотрены инструменты для управления этим сервером и для его конфигурирования в существующей среде многораздельной базы данных.

На приведенной ниже схеме показан экземпляр DB2 с четырьмя разделами базы данных. Они располагаются на двух выделенных хостах, `Machine1` и `Machine2`, и каждый хост содержит два логических раздела. Все серверы разделов базы данных обслуживаются одним сервером текстового поиска.

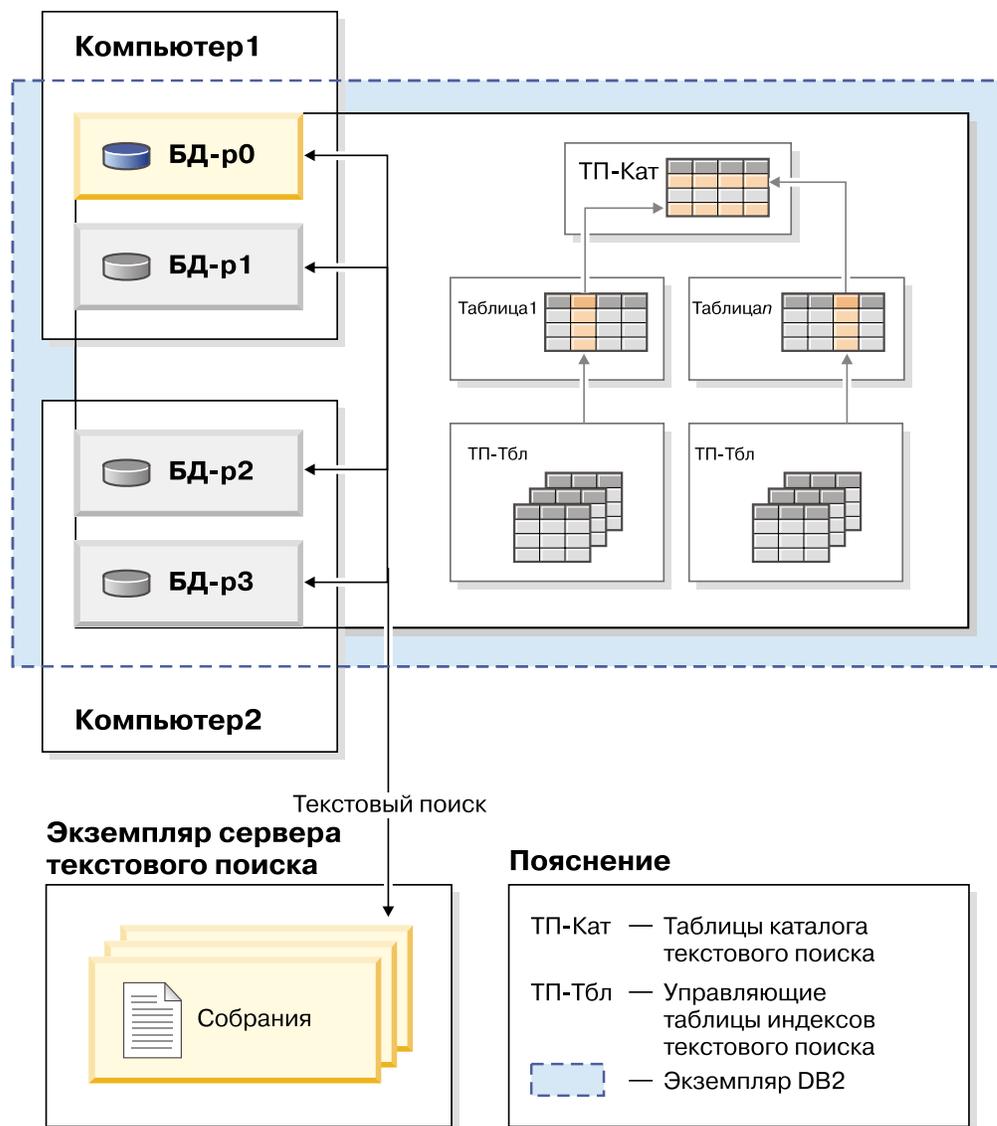


Рисунок 9. Установка сервера DB2 Text Search в среде многораздельной базы данных

Автономные установки призваны помочь в балансировании рабочей нагрузки и избежать совместного использования ресурсов сервером текстового поиска и единственным сервером раздела базы данных.

В средах многораздельной базы данных команда **db2ts START FOR TEXT** с параметрами **STATUS** и **VERIFY** возможна на любом из хостов сервера раздела. Чтобы запустить службы экземпляра, нужна команда **db2ts START FOR TEXT** на компьютере хоста интегрированного сервера текстового поиска. Компьютер хоста интегрированного сервера текстового поиска - это сервер раздела базы данных с наименьшим номером. Если используются пользовательские каталоги собраний, убедитесь, что после установки не были созданы разделы с меньшими номерами. Это ограничение особенно важно для платформ UNIX и Linux. Если вы конфигурируете DB2 Text Search при создании экземпляра, программа конфигурирования вначале выясняет, каков хост интегрированного сервера текстового поиска. Этим хостом всегда должен оставаться хост сервера раздела базы данных с наименьшим номером.

Разделы базы данных в многораздельном экземпляре могут добавляться и удаляться. Обычно это сопровождается перераспределением данных командой **REDISTRIBUTE DATABASE PARTITION GROUP**, чтобы переместить и повторно сбалансировать данные в таблицах. Если индекс текстового поиска размещен в одной из таблиц, затронутых перераспределением данных, потребуется перемещение содержимого разделов текстового индекса, чтобы выровнять их при новом наборе соответствующих разделов базы данных. Для этой цели обычно не подходят инкрементные обновления индекса текстового поиска, и приходится обновлять индекс текстового поиска с опцией **FOR DATA REDISTRIBUTION**. Обратите внимание на то, что при больших рабочих нагрузках это может привести к значительному времени простоя, как при начальном обновлении.

Решая установить DB2 Text Search или выполнить задачи управления в среде многораздельной базы данных, имейте в виду следующее:

- Убедитесь, что завершили установку DB2 согласно документации DB2. Монтировка NFS должна быть сконфигурирована с доступом **root** и **setuid**.
- Если запуск завершится неудачно, необходимо проверить, был ли DB2 Text Search сконфигурирован правильно, и затем повторить команду **db2ts START**.
- Прежде чем вставить или удалить номера разделов в файле **db2nodes.cfg**, остановите службы экземпляра DB2 Text Search. То же относится ко всем командам, выполнение которых может изменить файл конфигурации **db2nodes.cfg**.
- На платформах Windows при использовании DB2 Text Search в среде многораздельной базы данных файл **db2nodes.cfg** не должен использовать IP-адреса, и имена хостов для одного и того же хоста.

При выполнении поиска в среде многораздельной базы данных следует учитывать следующие особенности:

- Значение **RESULT LIMIT** во время поиска оценивается для каждого раздела. Это значит, что если задать для **RESULT LIMIT** значение 3 и использовать 4 раздела, можно получить до 12 результатов.
- Значение **SCORE** отражает степень соответствия документа только по сравнению со значениями **SCORE** других документов того же раздела, даже если запрашивалось несколько разделов.

---

## Инкрементные обновления для индексов текстового поиска DB2

Синхронизация данных в DB2 Text Search основана на триггерах, которые изменяют промежуточную таблицу всякий раз, когда эти триггеры захватывают информацию о новых, измененных и удаленных документах. Для каждого текстового индекса существует одна промежуточная таблица. Применение этой информации в промежуточной таблице к соответствующему текстовому индексу называют выполнением *инкрементного изменения*.

Опция **AUXLOG** параметра **INDEX CONFIGURATION** операции DB2 Text Search **CREATE INDEX** управляет выбором, использовать ли для индекса текстового поиска промежуточную текстовую таблицу. Если для опции **AUXLOG** задать значение **ON**, информация о новых и удаленных документах захватывается через обработку целостности во вспомогательную промежуточную таблицу, поддерживаемую DB2 Text Search, а информация об измененных документах захватывается через триггеры и хранится в промежуточной таблице. По умолчанию для этого параметра задается значение **ON** для таблиц с разделением по диапазонам и значение **OFF** для однораздельных таблиц.

При захвате изменений для инкрементного обновления текстового индекса посредством обработки целостности могут потребоваться дополнительные задачи управления. Например, перед обработкой обновлений индекса текстового поиска может потребоваться команда **RESET PENDING**. Влияние инфраструктуры промежуточного обслуживания текста аналогично влиянию материализованной таблицы запросов (MQT) с отложенным обновлением.

Если таблицы изменяются только командами, влияющими на все строки таблицы, такими как **LOAD REPLACE**, добавление инфраструктуры промежуточного обслуживания текста не даст преимуществ. Вместо этого рекомендуется после изменения таблицы создавать индекс заново.

Как создать текстовый индекс для таблицы и загружать данные при помощи начального и инкрементных обновлений, описано в приведенном ниже примере.

1. Создайте таблицу и добавьте в нее данные.  

```
db2 "create table test.simple (pk integer not null primary key,
comment varchar(48))"
db2 "insert into test.simple values (1, 'blue and red')"
```
2. Создайте индекс текстового поиска.  

```
db2ts "create index test.simpleix for text on test.simple(comment)
index configuration(auxlog on) connect to mydb"
```
3. Обновите индекс и загрузите данные.  

```
db2ts "update index test.simpleix for text connect to mydb"
db2 "load from loaddata4.sql of del insert into test.simple"
```
4. После операции загрузки базовая таблица заблокирована. Например, выбранная операция привела к сообщению SQL0668N Операция для таблицы "TEST.SIMPLE" недопустима из-за кода причины "1". SQLSTATE=57016. Промежуточная таблица доступна, но она еще не содержит информацию о новых данных.
5. Включите обработку целостности.  

```
db2 "set integrity for test.simple immediate checked"
```

Будет возвращено следующее сообщение:

```
SQL3601W Этот оператор вызвал автоматический перевод одной или нескольких таблиц
в состояние Отложенное задание целостности. SQLSTATE=01586
```

6. В данный момент промежуточная таблица заблокирована, и любые операции, изменяющие базовую таблицу, отклоняются. Например, следующий оператор завершится неудачно:  

```
"insert into test.simple values(15, 'green')"
```

Будет возвращено следующее сообщение:

```
DB21034E Данная команда обрабатывалась как оператор SQL, поскольку
она не является допустимой командой процессора командных строк. При
обработке SQL было получено сообщение:
SQL0668N Операция для таблицы "SYSIBMTS" ."SYSTSAUXLOG_IX114555" недопустима
из-за кода причины "1".
SQLSTATE=57016
```

7. Сбросьте состояние таблиц.  

```
db2ts "reset pending for table test.simple for text connect to mydb"
```

После успешной команды **RESET PENDING** промежуточная таблица будет разблокирована, и изменения базовой таблицы снова станут возможными. Разблокируйте промежуточную таблицу либо командой **RESET PENDING** для базовой таблицы, чтобы разблокировать все зависимые от нее промежуточные таблицы обслуживания текста, либо командой **SET INTEGRITY** для конкретной промежуточной таблицы.

8. Теперь промежуточная таблица обслуживания текста содержит изменения, которые нужно применить к индексу текстового поиска. Введите команду обновления для индекса.

```
db2ts "update index test.simpleix for text connect to mydb"
```

---

## Лингвистическая обработка для DB2 Text Search

Сервер DB2 Text Search предоставляет пакеты словарей для поддержки лингвистической обработки документов и запросов. Кроме того, для таких языков, как китайский, японский и корейский, поддерживается сегментация энграмм. Для таких языков, как китайский, японский и корейский, можно вместо разделения по словам на основе словаря выбрать сегментацию энграмм.

Для документа на одном из поддерживаемых языков лингвистическая обработка выполняется на этапе деления текста на лексемы (это означает, что текст разбивается на отдельные слова). Для неподдерживаемых языков документ анализируется при помощи сегментации по пробелам или же сегментации энграмм. Лемматизация (это обнаружение нормализованной формы слова с анализом слов по частям речи) для неподдерживаемых языков не выполняется.

При выполнении поиска с помощью индекса текстового поиска совпадение отмечается, если текст содержит лингвистические словоформы терминов запроса. Эти словоформы зависят от языка запроса.

### Лингвистическая обработка документов на китайском, японском и корейском языках

Для механизма поиска получение хороших результатов поиска большей частью зависит от приемов, используемых для обработки текста. После извлечения текста из документа первым шагом его обработки является идентификация в тексте отдельных слов. Идентификация в тексте отдельных слов называется сегментацией. Во многих языках для опознания границ слов можно использовать пробельные символы (знаки пробела, конца строки и некоторые знаки пунктуации). Однако в корейском, китайском и японском языках пробельные символы для разделения слов не используются, поэтому здесь нужны другие приемы.

DB2 Text Search предоставляет две возможности обработки для китайского, японского и корейского: либо морфологическую сегментацию, то есть разбиение на слова на основе словаря, либо сегментацию энграмм (значение по умолчанию).

Морфологическая сегментация использует словарь конкретного языка, чтобы обнаружить слова в последовательности символов в документе. Эта техника обеспечивает точные результаты поиска, так как для определения границ слов используются словари.

Сегментация энграмм решает задачу определения границ слов путем индексации перекрывающихся пар символов. Так как используются два символа, эта технология также называется сегментацией диграфов. Сегментация энграмм всегда возвращает все документы, содержащие термины поиска. Однако эта техника может возвращать документы, не соответствующие запросу.

### Пример

Чтобы показать, как работают оба типа лингвистической обработки, рассмотрим следующий текст в документе: выборы губернатора префектуры Канагава. На японском этот текст содержит восемь символов. Для этого примера представим эти

восемь символов как A B C D E F G H. Пример запроса, который может ввести пользователь - выборы губернатора; на японском этот текст содержит четыре символа E F G H. (В тексте документа и в примере запроса используются одинаковые символы).

- После индексации документа с использованием морфологической сегментации механизм поиска разобьет текст про выборы губернатора префектуры Канагава на следующие наборы символов: ABC DEF GH.

Пример запроса о выборах губернатора будет разбит на следующие наборы символов: EF GH. Символы EF не встречаются в маркерах текста документа. Документы не содержат EF, он содержит DEF.

Поскольку текст документа содержит DEF, а запрос содержит только EF, вероятность найти документ по данному примеру запроса уменьшилась.

Если выбрать морфологическую сегментацию, результаты будут точнее, но их будет меньше.

- После индексации документа с использованием сегментации энграмм механизм поиска разобьет текст про выборы губернатора префектуры Канагава на следующие наборы символов: AB BC CD DE EF FG GH.

Пример запроса 'выборы губернатора' сегментируется в следующие наборы символов: DE EF FG GH. Если вы выполняете поиск с примером запроса выборы губернатора, документ будет найден в этом запросе, поскольку маркеры и для текста документа, и для запроса, идут в одном и том же порядке.

Если выбрать сегментации энграмм, результатов будет больше, но они будут менее точными. Например, если выполняется поиск на японском языке при помощи запроса Киото, а документ в индексе содержит текст город Токио, запрос Киото вернет документ с текстом город Токио. Причина в том, что в город Токио и Киото подряд идут два одних и тех же иероглифа, хотя в Киото они составляют слово-название города, а в словосочетании "город Токио" "кио" (столица) - часть слова "Токио", а "то" - слово "город".

---

## Сценарий: Индексирование и поиск

После установки и конфигурирования DB2 Text Search надо выполнить четыре действия, прежде чем приступить к поиску.

1. Запустите службы экземпляра DB2 Text Search.
2. Подготовьте базу данных, которую будет использовать DB2 Text Search.  
Включите текстовый поиск в базе данных и используйте процедуру конфигурирования, чтобы завершить связывание сервера Text Search. Включить DB2 Text Search в базе данных нужно только один раз. Процедура конфигурирования нужна в следующих случаях:
  - включение не было завершено
  - для многораздельных баз данных
  - для установок автономного сервера Text Search.

Обратите внимание на то, что нельзя включить базу данных для Net Search Extender, если она уже включена для DB2 Text Search.

3. Создайте индекс текстового поиска для столбца, который содержит или будет содержать текст для поиска.
4. Заполните индекс текстового поиска. При этом данные добавляются в пустой, вновь созданный индекс текстового поиска.

Чтобы задать автоматическое обновление для индексов текстового поиска с заданными частотами обновления, посмотрите раздел о составлении расписаний обновления индексов DB2 Text Search.

Когда индекс текстового поиска будет содержать данные, можно проводить поиск индекса при помощи оператора SQL, а если индекс содержит данные XML - при помощи XQuery.

Как описано в рис. 10, вам нужно вручную или автоматически изменить существующие индексы текстового поиска, чтобы учесть изменения в текстовом столбце, с которым связан индекс.



Рисунок 10. Конфигурирование индексов текстового поиска для поиска в одностолбчатом экземпляре при использовании интегрированного сервера Text Search

## Основной сценарий

Предположим, что вы хотите сделать продукты в таблице PRODUCT базы данных SAMPLE доступными для поиска DB2 Text Search. В предположении, что вы уже создали базу данных sample (запустив команду **db2samp1**) и задали для переменной среды **DB2DBDFT** значение SAMPLE, можно ввести следующие команды:

```

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

Теперь имена продуктов и описания, содержащиеся в столбце NAME таблицы PRODUCT, проиндексированы и доступны для поиска. Если вы хотите найти ID продуктов всех снеговых лопат (snow shovels), введите следующий запрос поиска:

```

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

## Сценарий совместной работы DB2 Text Search и Net Search Extender

Если в базе данных уже включен поиск Net Search Extender и вы хотите использовать в этой базе данных Text Search, можно запрашивать базу данных с использованием возможности совместного существования индексов.

Запустите базу данных для текстового поиска.

```
db2ts start for text
DB20000I Команда SQL выполнена успешно.
```

Включите Text Search в базе данных, где уже есть индексы Net Search Extender.

```
db2ts enable database for text
CIE00001 Операция завершена успешно
```

Создайте и измените индекс DB2 Text Search для столбца, у которого есть существующий индекс Net Search Extender.

```
db2ts "CREATE INDEX db2ts.title_idx FOR TEXT ON books(title)"
CIE00001 Операция завершена успешно.
```

```
db2ts "UPDATE INDEX db2ts.title_idx FOR TEXT"
CIE00001 Операция завершена успешно.
```

Активируйте новый индекс DB2 Text Search, чтобы переключить обработку запросов с индекса NSE на новый индекс.

```
db2ts "ALTER INDEX db2ts.title_idx FOR TEXT SET ACTIVE"
CIE00001 Операция завершена успешно.
```

Введите запрос для использования индекса DB2 Text Search.

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

```
ISBN          TITLE
123-014014014 Climber's Mountain Tops
Выбрано 2 записи.
```

Запросы, которые пытаются использовать оба типа текстовых индексов, не поддерживаются. Например, здесь у столбца заголовка активный индекс DB2 Text Search, а у столбца bookinfo активный индекс Net Search Extender. Поиск возвратит ошибку, так как все текстовые индексы в одном запросе должны быть одного типа.

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

```
ISBN          TITLE
```

```
SQL20425N Столбец "BOOKINFO" в таблице "BOOKS" был задан как аргумент для
функции текстового поиска, но для этого столбца не существует индекса текстового поиска.
SQLSTATE=38H12
```

Чтобы избежать такой ошибки, создайте индекс DB2 Text Search для столбца bookinfo и активируйте его.

```
db2ts "CREATE INDEX db2ts.bookinfo_idx FOR TEXT ON books( bookinfo )"
CIE00001 Операция завершена успешно.
```

```
db2ts ALTER INDEX db2ts.bookinfo_idx FOR TEXT set active
CIE00001 Операция завершена успешно.
```

---

## Поддержка документов формата Rich Text Format (RTF) и собственных форматов

DB2 Text Search поддерживает индексацию и поиск документов формата Rich Text Format (RTF) и собственных форматов в правильно сконфигурированном экземпляре DB2 Text Search.

DB2 Text Search поддерживает форматы текстового индекса TEXT, XML и HTML для подготовки индексов полнотекстового поиска в текстовых данных. Кроме того, формат INSO позволяет индексировать документы формата Rich Text Format (RTF) или собственных форматов и выполнять поиск в таких документах:

- Документы в формате расширенного текста - это документы, содержащие текст и сведения о форматировании, такие как тип шрифта (жирный, курсив), размер шрифта, отступы и т.д.
- К собственным форматам относится целый ряд обычных офисных продуктов, таких как pdf, doc, ppt, ods.

Дополнительную информацию о включении и конфигурировании функции поддержки формата INSO смотрите в разделе об установке DB2 Text Search для формата Rich Text Format (RTF) и собственных форматов.

---

## Глава 3. Планирование решений текстового поиска

Знакомство с основными понятиями (например, поддерживаемыми типами документов и языками, ролями пользователей) поможет вам лучше использовать преимущества DB2 Text Search.

---

### Характеристики документа

#### Форматы документов, поддерживаемые для DB2 Text Search

Надо задать формат (или тип) текстовых документов, в которых собираетесь выполнять текстовый поиск с использованием DB2 Text Search. Эта информация необходима для индексации текстовых документов.

Данные в текстовом столбце могут быть простым текстом, документами HTML, документами XML, а также документами с расширенным текстом или собственным форматированием. Для документов выполняется синтаксический анализ, чтобы извлечь для индексации и тем самым сделать доступными для поиска только релевантные части. Некоторые элементы, например, теги и метаданные в документах HTML, не индексируются и поэтому недоступны для поиска.

#### Поддерживаемые типы данных

Типы данных в текстовых столбцах, которые вы хотите индексировать для поиска, могут быть двоичными или символьными.

DB2 Text Search поддерживает следующие типы данных:

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

#### Преобразование неподдерживаемых форматов и типов данных

Можно использовать собственную функцию для преобразования неподдерживаемого формата или типа данных в поддерживаемый формат или тип данных.

Создав текстовый индекс с использованием пользовательской функции, можно преобразовать неподдерживаемый формат в поддерживаемый формат, который может быть обработан при индексации, путем фильтрации неподдерживаемых символов.

Этот подход можно использовать также для индексации документов, хранящихся на внешних неподдерживаемых складах данных. В этом случае, когда столбец DB2 содержит ссылки на документы, можно использовать пользовательскую функцию, возвращающую содержимое документов по соответствующей ссылке на документ.

## Поддерживаемые языки и кодовые страницы

При первом создании индекса текстового поиска вы можете указать, что синтаксический анализ текстовых документов будет проводиться для конкретного языка. Можно задать также, что при поиске термины запроса будут интерпретироваться для некоторого языка. Дополнительно при создании индекса текстового поиска для столбца двоичного типа данных можно указать кодовую страницу.

### Спецификация языка

*Локаль* - это сочетание информации о языке и территории (регионе или стране), которое представляется пятисимвольным кодом локали. Вы определяете локаль сообщений для процедуры администрирования текстового поиска, передавая этой процедуре код локали. Уточнение кодов локалей возможно в зависимости от локалей, установленных на сервере DB2.

Есть важное различие между заданием языка при создании индекса текстового поиска и заданием языка при вводе запроса поиска:

- Локаль, заданная в вашей команде **db2ts CREATE INDEX**, определяет язык выполнения лексического или другого анализа документов для индексирования. Если известно, что все документы в столбце для индексирования используют некоторый конкретный язык, укажите применимую локаль при создании индекса текстового поиска. Если вы не указали локаль, для задания значения по умолчанию для параметра **LANGUAGE** будет использована территория базы данных. Чтобы просматривать ваши документы автоматически для определения локали, задайте в производной таблице SYSIBMTS.TSDEFAULTS для атрибута **LANGUAGE** значение AUTO. Производная таблица SYSIBMTS.TSDEFAULTS представляет значения по умолчанию в базе данных для текстового поиска при помощи пар атрибут-значение.
- Заданная вами в запросе поиска локаль используется при лингвистической обработке запроса и помогает определить основные формы термина запроса. После того, как локаль основной формы определена, она не играет роли в самом процессе поиска. Таким образом, можно использовать английский язык для запроса, но получать в результате поиска документы на русском языке, если в этих документах присутствует термин поиска в своей основной форме.

Список поддерживаемых локалей можно найти [здесь](#).

### Спецификация кодовой страницы

Документы можно индексировать, если они используют одну из поддерживаемых кодовых страниц DB2. Хотя указание кодовой страницы при создании текстового поиска необязательно, это поможет идентифицировать кодировку символов в двоичных столбцах. Если для двоичных столбцов кодовая страница не указана, используется кодовая страница из свойств столбца. .

## О размерах документов

DB2 Text Search накладывает ограничения на размер индексируемых документов и на число символов в документе.

Максимальный размер документов, которые могут быть обработаны успешно, задается параметром **MAXDOCUMENTSIZEINMB** в управляющей производной таблице **SYSIBMTS.TSDEFAULTS**. По умолчанию значение этого параметра - 100 Мбайт. Если размер документа превышает этот предел, он отвергается и в таблице событий создается запись с этой информацией, в том числе с первичным ключом для ее идентификации. Обработка продолжается для других документов, участвующих в этой операции изменения.

DB2 Text Search ограничивает число символов Unicode, которые можно индексировать для каждого текстового документа. Иногда это ограничение числа символов приводит к усечению больших текстовых документов в индексе текстового поиска.

Значение по умолчанию для числа символов Unicode, разрешенных для каждого текстового документа, зависит от формата текстового документа:

- Текстовые файлы, размер которых превышает значение *max.text.size* (в символах), усекаются до этого размера перед индексацией. Значение по умолчанию - 60 000 000 символов.
- Файлы XML, размер которых превышает значение *max.xml.text.size* (в байтах), не индексуются. Значение по умолчанию - 60 000 000 байт. При подсчете учитываются имена тегов, названия атрибутов и значения атрибутов; не учитываются директивы и комментарии XML.
- Двоичные файлы, размер которых превышает значение *max.binary.text.size* (в байтах), не индексуются. Значение по умолчанию - 60 000 000 байт. Это ограничение применяется после преобразования документа в текст.

Если размер текстового файла превышает максимальный размер текстового файла (по умолчанию это 60 миллионов символов), текстовый файл усекается до предельного размера перед индексацией. Если на стадии анализа документ усекается, вы получаете предупреждение, что некоторый текст не был обработан правильно или полностью.

Когда размер документа в двоичном формате или в формате XML превышает максимальный размер файла (по умолчанию это 60 миллионов байт), такой документ не индексируется и генерируется ошибка.

Если текст обработан неправильно или не полностью, результаты поиска будут не полны. По мере возможности отрегулируйте предельные размеры или сократите документ, чтобы он был обработан. Подробности об этом предупреждении записываются в таблицу событий, созданную для каждого индекса текстового поиска.

Если вы хотите повысить предельные размеры файлов, нужно соответственно увеличить размер кучи. Вы можете воспользоваться инструментом конфигурирования, чтобы отрегулировать максимальный размер кучи (параметр **startupHeapSize**).

---

## Обзор защиты DB2 Text Search

Текстовый поиск DB2 выполняет административные операции на основе ID авторизации пользователя, выполняющего данную операцию. В отличие от прошлых выпусков, владельцу экземпляра больше не требуются предварительные привилегии для баз данных, а изолированный пользователь может не входить в одну первичную группу с владельцем экземпляра.

Выполнение операций под ID авторизации пользователя улучшает аудиторскую способность и возможности управления текстовым поиском. Для упрощения управления доступом предусмотрены три новые системные роли:

- Администратор текстового поиска (SYSTS\_ADM) - выполняет операции на уровне базы данных
- Менеджер текстового поиска (SYSTS\_MGR) - выполняет операции на уровне индекса
- Пользователь текстового поиска (SYSTS\_USR) - может обращаться к данным каталога текстового поиска

Администратор системы защиты может предоставить или отменить эти роли, как и определяемые пользователем роли, однако роли с префиксом SYSTS управляются системой и их невозможно отбросить или создать.

При создании базы данных эти роли автоматически присваиваются создателю базы данных, а в неограниченных базах данных роль SYSTS\_USR присваивается группе PUBLIC (то есть всем). Чтобы разрешить или запретить текстовый поиск, остальные роли должны явным образом назначать администратор системы защиты, например, SYSTS\_ADM.

В ограниченной установке базы данных администратор системы защиты должен предоставить привилегии выполнения процедур планировщика роли SYSTS\_MGR и привилегии пользователя роли SYSTS\_USR.

Привилегии для таблиц, позволяющие управлять содержимым в таблицах каталога SYSIBMTS или получать доступ к этому содержимому, автоматически предоставляются ролям при включении поддержки DB2 Text Search в базе данных. Точно так же привилегии для таблиц, позволяющие управлять содержимым в управляющих таблицах SYSIBMTS для конкретного индекса текстового поиска или получать доступ к этому содержимому, автоматически предоставляются ролям при создании текстового индекса. Например, чтобы создать текстовый индекс, нужны привилегии для базовой таблицы, соответствующие привилегиям для создания других типов индексов, а также роль SYSTS\_MGR, обеспечивающая привилегии доступа к таблицам SYSIBMTS.

Для некоторых команд уровня индекса требуется соединение с сервером текстового поиска. Соответствующая информация о соединении, включая маркер аутентификации, будет получена из управляющей производной таблицы SYSIBMTS.TSSERVERS. Этот маркер генерируется при конфигурировании сервера текстового поиска и используется инициаторами вызова как механизм идентификации для адресации к нужному серверу текстового поиска. Если используется неправильный маркер, требование об управлении индексом или поиске отклоняется.

В следующей таблице приведена сводка необходимых привилегий роли. Для успешного выполнения операции администратор системы защиты должен был предоставить пользователю соответствующую роль.

Таблица 1. Привилегии роли

	Роль	Операция
Администратор текстового поиска	SYSTS_ADM	Разрешать, запрещать, очищать все блокировки команды, конфигурировать

Таблица 1. Привилегии роли (продолжение)

	Роль	Операция
Менеджер текстового поиска	SYSTS_MGR	Создавать, обновлять, изменять, отбрасывать, очищать события, очищать все блокировки команды для индекса, сбрасывать ожидание
Пользователь текстового поиска	SYSTS_USR	Ограниченный доступ к каталогу SYSIBMTS текстового поиска

## Роли пользователей

Для пользователей DB2 Text Search существуют различные роли и полномочия. Системные роли управляют привилегиями выполнения для административных операций, поэтому для выполнения операций текстового поиска каждому ID авторизации пользователя необходима подходящая роль текстового поиска в дополнение к привилегиям для базы данных или для доступа к таблицам.

Типичные пользователи:

- Администратор сервера текстового поиска
- Администратор текстового поиска
- Менеджер индексов текстового поиска
- Пользователи, выполняющие текстовый поиск при помощи запросов

### Администратор сервера DB2 Text Search

Администратор сервера Text Search конфигурирует опции сервера DB2 Text Search, запускает и останавливает службы экземпляра текстового поиска для интегрированного и автономного внедрения сервера текстового поиска, а также отслеживает работу сервера текстового поиска.

Кроме того, при интегрированной установке сервера текстового поиска эта роль привязана к владельцу экземпляра базы данных.

Владелец экземпляра различным образом определяется в операционных системах UNIX и Windows:

- В операционных системах UNIX принадлежащие владельцу экземпляра имя и ID пользователя экземпляра задаются для команды **db2i crt**.
- В операционных системах Windows владелец экземпляра - это ID пользователя, запускающего службу экземпляра DB2.

В отличие от DB2 Версии 9.7, владельцу экземпляра не нужно владеть полномочиями базы данных. При автономной установке сервера текстового поиска администратору сервера нужен соответствующий доступ к исполняемым файлам сервера текстового поиска, файлам конфигурации и файлам индекса.

### Администратор текстового поиска

Администратор текстового поиска включает и выключает использование DB2 Text Search в базах данных. Еще одна важная задача администратора текстового поиска - очистка блокировок команд.

Администратору текстового поиска требуется роль SYSTS\_ADM, а также полномочия DBADM, что позволяет работать со всеми объектами баз данных, в том числе с индексами текстового поиска.

## **Менеджер индексов текстового поиска**

Менеджер индекса текстового поиска определяет и поддерживает индексы текстового поиска.

Типичные задачи:

- Создание индексов текстового поиска и определение их характеристик
- Обновление индексов текстового поиска
- Изменение характеристик обновления индексов текстового поиска
- Отбрасывание индексов текстового поиска
- Периодическая очистка таблицы событий

Менеджеры индексов текстового поиска имеют роль SYSTS\_MGR и обычно имеют привилегию CONTROL для той таблицы, для которой создан индекс текстового поиска.

## **Пользователь, выполняющий текстовый поиск при помощи запросов**

Пользователи, выполняющие поиск при помощи запросов, могут использовать в запросе SQL функции CONTAINS и SCORE DB2 Text Search для пользовательской таблицы. Они могут использовать также функцию xmlcolumn-contains в XQuery, которая обращается к таблице с индексом текстового поиска.

Специальных полномочий поиска для DB2 Text Search не существует. В зависимости от прав доступа, предоставленных пользователю для таблицы, на которой создан индекс, запрос разрешается или отклоняется. Если пользователи могут ввести оператор SELECT для данной таблицы, они могут также выполнить текстовый поиск для этой таблицы.

Пользователи, выполняющие поиск при помощи запросов, могут включить в свои запросы, например, следующие функциональные возможности:

- Ограничить текстовый поиск конкретным документом (используя SQL или XQuery)
- Возвратить оценку, показывающую, насколько хорошо документ по сравнению с другими документами подходит для данного аргумента поиска (используя SQL)

## **Политики доступа и защита связи**

### **Особенности доступа к файлам для сервера текстового поиска**

Владелец процесса текстового сервера должен иметь доступ для чтения и записи данных конфигурации и данных всех собраний, включая те собрания, которые расположены в пользовательских каталогах собраний.

При интегрированном текстовом сервере владелец процесса - это владелец экземпляра, при автономных текстовых серверах это пользователь, запускающий текстовый сервер командой startup.

Собрания могут содержать конфиденциальные данные, часть которых может быть доступна для чтения, если открыть файл непосредственно. Для предотвращения

несанкционированного доступа проверьте и измените права доступа к каталогам конфигурации и собраний, оставив доступ к файлам только владельцам процесса.

## Политики доступа к промежуточным таблицам

Для идентификации изменений, которые должны быть применены к текстовому индексу, первичный ключ измененных строк (вставленных, обновленных, удаленных) вставляется в промежуточную таблицу.

Первичный ключ может основываться на столбцах данных базовой таблицы, содержащих конфиденциальные данные. По умолчанию у пользователей с ролью SYSTS\_ADM, SYSTS\_MGR и, с некоторыми ограничениями, SYSTS\_USR, есть по меньшей мере права доступа для чтения содержимого промежуточных таблиц. Политики доступа и аудита для базовой таблицы не наследуются для промежуточной таблицы. Если требуется дополнительно ограничить доступ к конкретным промежуточным таблицам, администратор защиты должен отозвать привилегии чтения для определенной таблицы у определенных ролей и предоставить их другому пользователю или пользовательской роли, которым поручается управление определенным текстовым индексом.

## Автономная установка

Экземпляр базы данных DB2 использует TCP/IP для связи с автономным сервером DB2 Text Search. Поддержка SSL или GSKit недоступна, хотя каналы шифрования могут использоваться через программу stunnel или туннелирование SSH. Ограничьте доступ к своему репозиторию документов и файлам индексов текстового поиска в зависимости от ваших требований к защите. Автономный сервер текстового поиска следует устанавливать на компьютерах с защищенным сетевым соединением позади брандмауэра, чтобы предотвратить несанкционированный доступ к индексам текстового поиска. Нужно так сконфигурировать ограничения доступа TCP/IP к автономному серверу текстового поиска, чтобы доступ к нему оставался только у хоста, где установлен сервер базы данных.

---

## Планирование мощности и оптимизация DB2 Text Search

На производительность и использование ресурсов DB2 Text Search влияет ряд факторов. Планируя ресурсы системы для DB2 Text Search, примите во внимание рабочую нагрузку запросов, число параллельных обновлений индексов, ожидаемый размер и темп роста индексов текстового поиска, а также время обработки индексируемых документов.

DB2 Text Search дает возможность выполнять запросы полнотекстового поиска для большинства типов данных в базе данных DB2, включая поддержку документов XML и возможность формата RTF или собственных форматов. Полнотекстовый поиск поддерживается с помощью экземпляра сервера текстового поиска, интегрированного с экземпляром базы данных или в автономной установке, связанной с экземпляром базы данных. Связь между базой данных и сервером текстового поиска осуществляется по протоколу TCP/IP. Полнотекстовая индексация и производительность поиска зависят от конфигурации сервера текстового поиска, доступных ресурсов системы и особых настроек текстового индекса.

## Внедрение и конфигурирование сервера текстового поиска

Отдельный сервер текстового поиска конфигурируется для экземпляра базы данных. Минимальный рекомендуемый объем памяти для сервера текстового поиска - 4

Гбайта для производственного режима; это число увеличивается в соответствии с числом параллельных обновлений индексов.

Обновление индекса текстового поиска - ресурсоемкая операция как в отношении дискового ввода-вывода, так и в отношении ресурсов процессора и памяти. Для управления использованием ресурсов сервера текстового поиска есть несколько параметров конфигурации. Для распределения рабочей нагрузки, например, в среде многораздельных баз данных, рекомендуется автономная установка.

## Размер индексов текстового поиска

В среднем размер индекса текстового поиска составляет приблизительно 50-150% размера исходных данных.

Абсолютного предела размера индекса текстового поиска не существует, но сочетание производительности и времени завершения работы диктует практические пределы размера индекса. Например, при удалении из индекса или добавлении в индекс заметного количества данных структура индекса текстового поиска объединяется для улучшения производительности при выполнении запроса, и время, необходимое для объединения, зависит от размера индекса.

## Факторы, влияющие на пропускную способность

Абсолютная пропускная способность обновления индекса текста зависит от типа данных и формата индекса. Изменение производительности запросов больше всего заметно при наличии большого количества результатов, а не большого размера индекса текстового поиска. Например, запрос с одним предикатом и одним термом поиска по индексу текстового поиска размером 100 Гбайт будет работать почти так же, как и поиск по индексу текстового поиска объемом 800 Гбайт, если они дают одинаковое число результатов.

Оптимально обработка обновлений текстового индекса происходит при объеме 10-100 Кбайт текста на документ. Пропускная способность снижается при размерах текстовых документов больше 1 Мбайта и меньше 1 Кбайта.

## Конфигурация сервера текстового поиска DB2

Можно настраивать конфигурацию текстового поиска DB2, меняя размер очереди, размер кучи, количество потоков индексирования и другие параметры. Сбалансировав эти параметры, можно добиться оптимальной производительности системы.

При конфигурировании сервера DB2 Text Search число потоков индексатора должно быть не больше числа процессоров, а число параллельных обновлений должно быть не больше числа потоков индексатора. Чтобы узнать число параллельных обновлений в многораздельной базе данных, нужно умножить число индексов на число собраний для текстового индекса.

Перед любым изменением конфигурации остановите службы экземпляра DB2 Text Search командой **db2ts STOP FOR TEXT**.

Запустите утилиту configUtility.

- В случае интегрированного сервера текстового поиска она расположена в каталоге `<домашний_каталог_TS>/bin`.
- В случае автономного сервера текстового поиска она расположена в каталоге `<домашний_каталог_ECMTS>/bin`.

Например, чтобы изменить число потоков индексации:

```
configTool configureParams -configPath configPath -numberOfIndexingThreads 3
```

Чтобы изменения вступили в силу, перезапустите процесс текстового поиска DB2.

## Конфигурирование максимального размера кучи

Когда документ получен из потока входных документов, его содержимое помещается в очередь документов. Документы, помещенные в очередь документов, остаются в ней, пока их не проиндексирует активный поток индексирования. В обычных случаях скорость помещения документов в очередь выше, чем скорость их анализа и индексирования. Следовательно, в какой-то момент времени очередь документов будет заполнена и поток приема документов будет заблокирован, пока в очереди не освободится место.

По мере заполнения очереди документов необработанными документами очередь расходует память кучи. Память используется также для обработки документов - синтаксического анализа и индексирования. Общее потребление памяти должно быть меньше максимального размера кучи для процесса. По умолчанию размер кучи сконфигурирован равным 1500 Мбайт.

Кроме того, следует учитывать соотношение между размерами памяти входящей и исходящей очереди и памяти кучи. Размер очереди определяется потреблением памяти документами в этой очереди. Если требуется обрабатывать большие документа, например по 20 Мбайт, и решено увеличить размер памяти очереди, следует также рассмотреть возможность увеличения кучи.

Переменная *startupHeapSize* задает максимальный разрешенный размер кучи для интегрированного или автономного сервера DB2 Text Search. Начальный размер кучи по умолчанию - 1,5 Гбайта. Можно задать объем от 1,5 Гбайт до максимального количества памяти, которое позволяет использовать операционная система и JVM. Рассмотрим следующие примеры:

- В Windows с 32-битной JVM максимальный объем памяти кучи для процесса - 2 Гбайта. Поэтому значение параметра *startupHeapSize* должно быть не больше 2 Гбайт. Например, задайте 1,8 Гбайт.
- В системе AIX с 64-битной JVM максимальный размер кучи ограничен только объемом виртуальной памяти системы. Если требуется обрабатывать непрерывный поток из множества больших документов, средний размер которых 20 Мбайт, параметр *startupHeapSize* следует увеличить примерно до 4 Гбайт.

Вы можете задать максимальный размер кучи при установке или обновлении автономного сервера DB2 Text Search, задав параметр **IA\_STARTUP\_HEAP\_SIZE** в файле ответов. Если задать для максимального размера кучи значение больше 2 Гбайт при установке или обновлении автономного сервера текстового поиска в 64-битной операционной системе, для новых собраний увеличивается предельный размер файлов для текстовых документов, документов XML и двоичных документов. Предельные размеры файлов задаются отдельно для каждого собрания в файле `<домашний_каталог_ECMTS>\config\collections\имя_собрания\parser_config.xml`. Предельные размеры файлов по умолчанию для новых собраний задаются в файле `<домашний_каталог_ECMTS>\config\defaults\parser_config.xml`. Для каждых 8,3 Мбайт памяти кучи сверх 2 Гбайт значения предельных размеров файлов (по умолчанию 60 Мбайт) увеличиваются на 1 Мбайт (до 400 Мбайт).

**Внимание:** Если изменить максимальный размер кучи при помощи инструмента конфигурирования после установки, нужно вручную отрегулировать предельные размеры файлов в файле `parser_config.xml`. Только при установке и обновлении предельные размеры файлов изменяются автоматически после задания параметра **IA\_STARTUP\_HEAP\_SIZE** в файле ответов.

Для изменения максимального размера кучи введите следующую команду:  
`configTool configureParams -configPath <полный_путь_к_папке_конфигурации>  
-startupHeapSize <значение>`

Где *<значение>* - размер кучи, а *<полный-путь-к-папке-конфигурации>* - полный путь к файлу `config.xml` для сервера текстового поиска DB2.

В 32-битной операционной системе типичная конфигурация такая:

- Максимальный размер кучи: 1,8 Гбайт
- Размеры очереди: 90 Мбайт каждая
- Предельные размеры файлов: 60 Мбайт

В 64-битной операционной системе типичная конфигурация такая:

- Максимальный размер кучи: 3 Гбайт
- Размеры очереди: 150 Мбайт каждая
- Предельные размеры файлов: 200 Мбайт

## Потоки индексирования DB2 Text Search

Несколько потоков индексирования работают параллельно над синтаксическим анализом и индексом документов. Обычно это уменьшает общее время обновления индекса текстового поиска.

Потоки индексирования выбирают документы из очереди и управляют процессом индексирования. Они используют потоки предварительной обработки индекса для подготовки содержимого документа к индексированию и записывают результат в собрание текстового индекса.

Потоки предварительной обработки извлекают текст, определяют язык, разбивают текст на элементы и анализируют этот документ.

Обычно конфигурируется равное их количество. Но в некоторых случаях, при обработке больших документов, увеличение количества потоков предварительной обработки может дать выигрыш в производительности.

## Использование потоков индексирования

Если несколько потоков индексирования работают с одной коллекцией, эффект уменьшается из-за координации, необходимой для синхронизации обработки между потоками. При синтаксическом анализе однопоточная работа потока индексирования выгодна, но производительность может снижаться при объединении или записи на диск. Например, четыре потока индексирования, работающие с четырьмя разными текстовыми индексами, покажут лучший результат, чем четыре потока индексирования, работающие с одним текстовым индексом.

## Количество потоков индексирования

Потоков индексирования должно быть не меньше двух, при этом их количество не должно превышать количество доступных процессоров. Максимальное количество параллельных обновлений индекса не должно превышать количество потоков

индексирования во избежание объединения потоков. При слишком большом количестве потоков индексирования или параллельных обновлений индекса общая производительность системы снижается из-за расхода памяти на переключение контекстов процесса.

Например, если 40 текстовых индексов часто обновляются на системе с 8 процессорами, не следует использовать больше 8 потоков индексирования. Следует также использовать расписание с чередованием обновлений текстовых индексов для минимизации конфликтов между потоками индексирования.

По умолчанию значение параметра количества потоков индексирования равно 4, такое же значение применяется к потокам предварительной обработки индекса.

Для конфигурирования количества потоков индексирования введите следующую команду:

```
configTool configureParams -configPath <полный_путь_к_папке_конфигурации>  
-numberOfIndexerThreads <значение>
```

Где <значение> - количество потоков, а <полный-путь-к-папке-конфигурации> - полный путь к файлу config.xml для сервера текстового поиска DB2.

Для конфигурирования количества потоков предварительной обработки введите следующую команду:

```
configTool configureParams -configPath <полный_путь_к_папке_конфигурации>  
-numberOfPreprocessingThreads <значение>
```

Где <значение> - количество потоков, а <полный-путь-к-папке-конфигурации> - полный путь к файлу config.xml для сервера текстового поиска DB2.

## Размер памяти очереди DB2 Text Search

Для оптимальной обработки обновлений индексов надо правильно задать размер памяти очереди DB2 Text Search. Управлять выделением памяти очередей можно как для базы данных, так и для текстового сервера.

Память очереди базы данных определяет число документов, которые можно за раз отправить текстовому серверу для обработки обновлений. Чтобы управлять объемом памяти очереди базы данных, обновите производную таблицу управления SYSIBMTS.TSDEFAULTS и задайте значение для параметра **DocumentResultQueueSize**. По умолчанию используется 10000. Это значение используется для ограничения объема памяти базы данных, резервируемой для каждой операции обновления для собрания. Обратите внимание на то, что в многораздельной установке одно обновление текстового индекса, сконфигурированное для параллельного выполнения, зарезервирует память для всех обновляемых собраний.

Второй механизм управления памятью очередей применяется к текстовому серверу. Использование памяти очередей определяется двумя значениями конфигурации.

- *inputQueueMemorySize*:  
Задает объем памяти для входной очереди на сервере индексирования. Входная очередь содержит документы, ожидающие предварительной обработки. Увеличение объема памяти ускоряет работу, но увеличивает потребление ресурсов. Размер по умолчанию - 15 Мбайт.
- *outputQueueMemorySize*:  
Задает объем памяти для выходной очереди на сервере индексирования. Входная очередь содержит документы, ожидающие индексирования после предварительной

обработки. Увеличение объема памяти ускоряет работу, но увеличивает потребление ресурсов. Размер по умолчанию - 15 Мбайт.

Обратите внимание на соотношение между размером входной и выходной очередей и размером памяти кучи. Размер очереди определяется потреблением памяти документами в этой очереди. Если вы собираетесь обрабатывать длинные документы, например, по 20 Мбайт каждый, рассмотрите возможность увеличения объема памяти очередей и размера кучи.

Чтобы изменить, например, размер *inputQueueMemory*, введите команду:  
`configTool configureParams -configPath <полный_путь_к_папке_конфигурации>  
-inputQueueMemorySize <значение>`

где <значение> - объем памяти, а <полный\_путь\_к\_папке\_конфигурации> - полный путь к файлу config.xml для DB2 Text Search.

## Планирование и оптимизация индексов DB2 Text Search

Характеристики источников данных оказывают значительное влияние на производительность.

Время, требуемое для выполнения обновления текстового индекса, зависит преимущественно от следующих факторов:

- число документов для индексирования
- размер документа
- тип индекса
- параллелизм обновления индекса
- конфигурация сервера текстового поиска

Время обработки каждого документа - это сумма приблизительно фиксированного времени и переменного времени. Фиксированное время зависит от типа документа, например, простой текст, XML или INSO. Фиксированное время приблизительно, так как возможны незначительные отклонения во времени, требуемом для выделения или повторного использования памяти. Переменное время определяется в первую очередь размером документа и вариациями лингвистической обработки.

Для индексов документов INSO на время обработки может также повлиять обработка разных типов MIME.

Число документов, которые можно обработать за заданный промежуток времени, больше для документов меньших размеров. Однако общая пропускная способность для небольших документов меньше, чем для больших, из-за фиксированной части стоимости обработки каждого документа.

## Характеристики источников индексов DB2 Text Search

Используйте следующие приемы для повышения производительности индексирования или поиска:

- Для столбцов первичных ключей используйте числовые типы данных, такие как INTEGER, а не тип VARCHAR. Чтобы минимизировать объем передаваемых данных для результатов поиска, избегайте первичных ключей, составленных из нескольких столбцов VARCHAR.
- Убедитесь, что в системе достаточный объем доступной реальной памяти для операций обновления индекса. Для обновлений индексов требуется память сверх

того, что требуется для пулов буферов баз данных. Если памяти недостаточно, операционная система использует пространство подкачки, что значительно снижает производительность поиска.

- Если во время обновлений индекса сервера текстового поиска должны обрабатываться много небольших документов, рассмотрите возможность уменьшения числа параллельных обновлений индекса и увеличения размеров очередей, чтобы увеличить максимальный поток документов к текстовому серверу. Подробную информацию смотрите в разделах о планировании мощности.
- Убедитесь, что индексируемое содержимое доступно и находится в нужном формате, так как если при обновлении индекса в таблицу событий записывается много сообщений об ошибках и предупреждений, производительность может снизиться.

## Асинхронные изменения индексов

Для повышения производительности индекс текстового поиска не синхронизируется с пользовательской таблицей внутри области транзакции DB2, в ходе которой текстовые документы обновляются, удаляются или добавляются в эту таблицу. Индекс текстового поиска обновляется асинхронно.

Для облегчения асинхронного обновления индекса текстового поиска для каждого индекса текстового поиска создается промежуточная таблица (другое название - таблицей журнала). Триггеры для текстовых таблиц захватывают любые изменения, например вставки, удаления или обновления текстового столбца, с которым связан индекс текстового поиска, и записывают их в промежуточную таблицу. Для захвата изменений, выявляемых при обработке целостности, служит вспомогательная промежуточная таблица. Обновления индекса текстового поиска применяются позже во время ручного или автоматического обновления. Обновление вносится в копию очень маленького фрагмента индекса. Во время обновления можно по-прежнему выполнять поиск по индексу, но до окончания синхронизации нельзя получить доступ к обновленному индексу текстового поиска.

При использовании опции `updateautocommit` следует учитывать, что каждый цикл принятия требует значительного использования процессора, что увеличивает общее время выполнения обновления индекса. Использование опции `updateautocommit` особенно полезно для начальных обновлений или обновлений, затрагивающих все строки или большую часть строк, если общее затраченное время велико, чтобы избежать потери сделанной работы в связи с откатом из-за сбоя системы или сервера.

## Оптимизация индекса DB2 Text Search

Оптимизация индекса DB2 Text Search компактифицирует индекс текстового поиска и ускоряет индексирование и поиск. Оптимизация удаляет из индекса текстового поиска удаленные документы и сливает файлы сегментов индекса на диске.

Оптимизацию и индексирование одного и того же индекса нельзя выполнять параллельно. Учитывайте это при составлении расписания сеансов оптимизации и индексирования. Но оптимизацию можно выполнять параллельно с поиском. Во время оптимизации индекса может наблюдаться повышенное потребление дискового пространства, особенно если параллельно выполняется поиск по тому же самому индексу.

Вы можете оптимизировать индекс после полного индексирования набора документов или после инкрементных обновлений индекса. Оптимизация индекса может занять значительное время в зависимости от размера индекса. Если при инкрементных обновлениях часто добавляются новые документы, выполняйте оптимизацию реже, чтобы минимизировать использование процессора на процесс оптимизации.

Чтобы оптимизировать индекс:

1. Из каталога *домашний\_каталог\_ECM TS*/bin запустите инструмент управления командой **optimizeIndex**. Например:  

```
adminTool.bat optimizeIndex -configPath  
"C:\Program Files\IBM\ECMTextSearch\config"  
-collectionName MyCollection
```
2. Чтобы проверить состояние последнего запущенного процесса оптимизации, запустите инструмент управления с командой **optimizeIndexStatus**.

## Потребление дискового пространства

### Размер текстового индекса

Объем дискового пространства, используемого индексом текстового поиска, сильно зависит от характера текста документа. При этом между размером индекса текстового поиска и размером исходных данных существует приблизительно линейная зависимость. Как правило, размер индекса на диске составляет от 50% до 150% размера исходного текста. Например, для таблицы с целочисленным первичным ключом текстовый индекс поиска для 100 000 документов по 20 Кбайт, вероятно, потребует около 1100 Мбайт дискового пространства (100 000 x 20 Кбайт x 55%). Отношение размера индекса текстового поиска к размеру исходных документов зависит от следующих факторов:

- среднего размера документа;
- размера ключа документа (столбцов первичного ключа)
- числа сортируемых полей
- количества и распределения уникальных терминов.

В ходе обновления индекса требуется дополнительное место. В среднем требуемый размер свободного пространства в 2-3 раза больше окончательного размера индекса текстового поиска, если не достигнут предел размера сегмента. Требуемое свободное пространство в 2-3 раза больше максимального размера сегмента. Дисковое пространство резервируется даже после объединения сегментов, если старые сегменты использовались в поиске.

### Файлы журналов

Кроме файла `db2diag.log`, текстовый поиск DB2 создает файлы трассировки и журнала утилиты конфигурирования, содержащие сообщения сервера текстового поиска DB2.

Для интегрированного сервера Text Search файлы журнала расположены по умолчанию в каталоге `db2tss/1log`. Если требуется хранить базу данных DB2 и файлы журнала текстового поиска в одном месте, задайте положение `<домашний-каталог-экземпляра>/sqllib/db2dump/tslog` в UNIX или `<путь-профиля-экземпляра>\<имя-экземпляра>\db2tss\tslog` в Windows.

Для автономной установки журналы сервера DB2 Text Search расположены по умолчанию в каталоге `<домашний_каталог_ECM TS>/1log`. Положение по умолчанию можно изменить при установке, задав параметр **IA\_LOG\_PATH** в файле ответов.

В обоих случаях убедитесь, что в указанном положении достаточно свободного дискового пространства для файлов журнала. Требуется как минимум 100 Мбайт свободного места. Если места недостаточно, служба текстового поиска останавливается и выдает ошибку переполнения диска.

## Административные таблицы

Если при выполнении команды **CREATE INDEX FOR TEXT** не указано табличное пространство для административных таблиц для индекса текстового поиска, административная таблица создается в том же табличном пространстве, что и базовая таблица. При определении нужного расположения учитывайте следующее:

- Промежуточная таблица для текстового индекса

Промежуточная таблица содержит ссылку на строки, которые были обновлены в базовой таблице для инкрементного обновления текстового индекса. Эта таблица автоматически очищается при каждом обновлении:

Размер =  
количество строк обновления индекса \* (длина первичного ключа базовой таблицы + 18)

- Таблица событий для текстового индекса

Таблица событий содержит сведения о состоянии обработки текстового индекса, включая ошибки и предупреждения в ходе обновления индекса. В худшем случае, если каждый документ отклоняется вследствие нефатальной ошибки, количество событий равно количеству документов с прибавлением нескольких начальных и конечных сообщений процесса обновления. Таблица событий не очищается автоматически, и до завершения операции **CLEAR EVENTS FOR INDEX** ее размер продолжает увеличиваться.

Размер таблицы событий =  
количество событий \* (длина первичного ключа базовой таблицы + 1050)

## Положение индексов DB2 Text Search

Следует учитывать, что положение индекса по умолчанию изменилось в этом выпуске.

Для интегрированного сервера Text Search метаданные конфигурации и собраний хранятся в *домашний\_каталог\_экземпляра/sql1lib/db2tss/config* в UNIX или в *путь\_профиля\_экземпляра\instance\_name\db2tss\config* в Windows.

Конфигурация и метаданные собрания для каждого индекса текстового поиска требуют небольшого пространства. Но если не указан пользовательский путь, положение индексов текстового поиска - в подкаталоге каталога *db2tss/config*. Это положение часто ограничено по размеру, и поэтому настоятельно рекомендуется конфигурировать параметр **defaultDataDirectory** для сервера текстового поиска, задавая пользовательское положение с достаточным дисковым пространством, если вы планируете создавать большое число индексов или индексы большого размера при интегрированном сервере Text Search.

Положение данных собрания определяется при создании собрания и сохраняется в файле *collection.xml*. При автономных серверах DB2 Text Search положение файлов конфигурации для собраний определяется параметром **defaultDataDirectory**. По умолчанию каталог конфигурации собрания - *<домашний\_каталог\_ECMTS>\config\collections*, а данные собрания находятся в подкаталоге каталога конфигурации собрания **defaultDataDirectory\имя\_собрания\data\text**.

В любом случае, если вы планируете создать несколько больших индексов, рассмотрите возможность их хранения на отдельных дисковых устройствах или массивах, в особенности если будут выполняться одновременные обновления индексов по расписанию.

## Параметры, специфичные для индексов, для обновлений индексов DB2 Text Search

Для повышения производительности можно сконфигурировать следующие параметры, зависящие от собрания:

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

Чтобы изменить параметры индексации в конкретном собрании, отредактируйте файл *домашний\_каталог\_ECMTS\config\collections\имя\_собрания\collection.xml*. Чтобы изменить параметры по умолчанию для собраний, которые будут создаваться в дальнейшем, задайте значения этих параметров в файле *домашний\_каталог\_ECMTS\config\defaults\collection.xml*.

- Параметр **MaxMergeDocs** определяет самый большой сегмент (по числу документов), который можно слить с другими сегментами в индексе. Надо соблюсти баланс между общей пропускной способностью индексации и временем слияния сегментов.

Если задать для параметра **MaxMergeDocs** маленькое значение (например, 100 000 документов), размер сегментов будет ограничен. В этом случае слияние сегментов проходит быстрее, и индексация выполняется более гладко, без задержек. Однако если вы работаете с большим содержимым, получится слишком много сегментов, что со временем вызовет ухудшение пропускной способности индексации.

Если указать для параметра **MaxMergeDocs** большое значение (например, 100 000 000 или 500 000 000 документов), сегментов будет меньше (если только индекс не будет очень большим), и общая пропускная способность индексации увеличится. Однако слияние сегментов будет отнимать больше времени, и вы можете столкнуться с задержками при индексации.

Обычно значение параметра **MaxMergeDocs** должно быть больше для собраний небольших документов и меньше для собраний больших документов.

- Параметр **MaxMergeMB** определяет самый большой сегмент (по физическому размеру файла), который можно слить с другими сегментами в индексе. Надо соблюсти баланс между общей пропускной способностью индексации и временем слияния сегментов. Если задать для параметра **MaxMergeMB** маленькое значение (например, 500 Мбайт), размер сегментов будет ограничен. В этом случае слияние сегментов проходит быстрее, и индексация выполняется более гладко. Однако если вы работаете с большим содержимым, получится слишком много сегментов, что со временем вызовет деградацию производительности индексации и поиска.

Если указать для параметра **MaxMergeMB** большое значение (например, 50 000 Мбайт или 100 000 Мбайт), сегментов будет меньше (если только индекс не будет очень большим), и общая производительность индексации увеличится. Однако слияние сегментов будет отнимать больше времени, и вы можете столкнуться с задержками при индексации.

- Параметр **MergeFactor** определяет число сегментов, сливаемых за один раз, а также управляет общим числом сегментов, которое можно накопить в одном индексе. Здесь стоит выбор между частыми небольшими слияниями (например, два за один раз) и редкими большими слияниями (например, 10 за один раз). Чтобы избежать задержек, можно указать меньшее значение для параметра **MergeFactor**. Изменение параметра слияния обычно не влияет на производительность.
- Параметр **BufferSize** задает объем оперативной памяти, который можно использовать для буферизации добавляемых документов перед тем, как они будут

сброшены на диск в качестве нового сегмента. Здесь стоит выбор между частыми небольшими сбросами на диск и редкими большими сбросами на диск. В некоторых случаях можно повысить производительность, увеличив значение параметра **BufferSize**. Например, при индексировании одного собрания небольших документов увеличение размера буфера повысит производительность, особенно для первых 100 000 документов в индексе.

## Настройка системы DB2 Text Search

На производительность обновления текстовых индексов и выполнения запросов текстового поиска влияют различные характеристики системы.

Примите во внимание следующее:

- Особенности портов TCP/IP для Windows
- Дескрипторы файлов

### О портах TCP/IP для DB2 Text Search и Windows

В 32-битных операционных системах Windows на возможность обработки больших нагрузок запросов влияет число портов TCP/IP и время ожидания повторного использования порта.

### Назначения портов в Windows (32-битной)

Интегрированная версия DB2 Text Search работает как отдельный процесс на том же хосте, что и сервер баз данных. Сервер баз данных и текстовый сервер связываются через соединение TCP/IP.

На число доступных портов для соединений TCP/IP влияет число портов и время ожидания повторного использования порта после закрытия соединения. Значения этих параметров по умолчанию могут быть недостаточны, чтобы обеспечить достаточное число доступных портов для обслуживания значительной нагрузки запросов. Если у вас слишком мало портов TCP/IP, может возвращаться ошибка CIE00756 Соединение не удалось установить.

Если возникает ошибка CIE00756 Соединение не удалось установить, введите следующую команду, чтобы проверить использование портов на сервере:

```
netstat -n  
netstat -n | c:\windows\system32\find /I <номер_порта>
```

Если выводится много соединений TCP/IP и локальных адресов 127.0.0.1:номер\_порта в состоянии TIME\_WAIT, высока вероятность, что у сервера заканчиваются порты TCP/IP.

Чтобы определить номера портов DB2 Text Search, введите следующую команду:  
configTool printAdminHTTPPort -configPath %INSTPROF%\%DB2INSTANCE%\db2tss\config

где для INSTPROF задано значение переменной реестра **DB2INSTPROF**, применяемой к интегрированным установкам сервера DB2 Text Search.

### Параметры портов

Параметрами портов управляют следующие записи реестра, находящиеся в HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\TCP/IP\Parameters:

- **TcpTimedWaitDelay**

Значение DWORD в диапазоне от 30 до 300, определяющее время в секундах, которое должно пройти до того, как TCP/IP сможет освободить закрытое соединение и повторно использовать его ресурсы. Задайте для значения **TcpTimedWaitDelay** более низкое значение, чтобы уменьшить время, в течение которого гнезда остаются в состоянии TIME\_WAIT.

- **MaxUserPort**

Значение DWORD, определяющее максимальный номер порта, который может быть назначен TCP/IP, когда прикладная программа запрашивает свободный пользовательский порт. Задайте для **MaxUserPort** большее значение, чтобы увеличить общее число гнезд, которые можно соединить с портом.

Система, генерирующая много требований соединения, может заработать быстрее, если задать для **TcpTimedWaitDelay** значение 30 секунд, а для **MaxUserPort** - 32678.

После добавления или изменения этих записей реестра перезагрузите компьютер Windows, чтобы изменения вступили в силу.

## Дескрипторы файлов DB2 Text Search

При обновлении индексов и выполнении запросов DB2 Text Search для обработки множественных требований обновления индекса и поиска используются такие системные ресурсы, как дескрипторы файлов.

Обычно в системе число открытых дескрипторов файлов для каждого процесса может быть ограничено сравнительно небольшим числом, например, 1024, из-за чего у сервера текстового поиска могут закончиться дескрипторы файлов. В этом случае требования поиска и обновления завершаются неудачно.

Для исправления этой ошибки:

- Поищите в журналах сервера исключительную ситуацию со строкой сообщения вроде `too many open files` (слишком много открытых файлов).
- В системах UNIX проверьте ограничения системы с помощью команды `ulimit -a`.

Чтобы увеличить число дескрипторов файлов, выполните следующие действия:

1. Остановите сервер текстового поиска.
2. Увеличьте число дескрипторов файлов, разрешенных для одного процесса, как описано в руководстве по вашей операционной системе. Это увеличение числа дескрипторов файлов должно быть достаточным для всех требований во всех сеансах регистрации.
3. Перезапустите сервер текстового поиска.

---

## Планирование запросов DB2 Text Search

При планировании запроса текстового поиска необходимо принять во внимание несколько аспектов.

### Аргументы текстового поиска DB2 (DB2 Text Search)

Символы подстановки и предел их раскрытия, регистрозависимость аргументов и их опции - это различные типы аргументов текстового поиска, и все они могут повлиять на производительность запросов.

## Символы подстановки

Использование символа подстановки в начале термина поиска замедляет обработку запроса. Старайтесь по возможности не прибегать к поиску вида *\*термин\_поиска* или *?термин\_поиска*.

## Предел раскрытия символов подстановки

Когда термин запроса включает в себя термин подстановки, для получения соответствующих документов для термина запроса выполняется раскрытие. Собрание текстовых индексов может включать в себя больше отдельных терминов совпадения, чем разрешается пределом раскрытия символов подстановки. В этом случае возвращается только то подмножество документов, которое соответствует уже раскрытым терминам. Это ограничение применимо к символу подстановки звездочка (\*).

По умолчанию может быть возвращено 1024 термина. Чтобы изменить этот предел, задайте параметр `queryExpansionLimit` со значением в файле `ECMTS_HOME\config\config.xml`. Например, для задания предела 4096 добавьте в этот файл такую строку: `<queryExpansionLimit>4096</queryExpansionLimit>`

## Учет регистра

Аргументы текстового поиска не зависят от регистра, даже если точный термин или словосочетание задается с помощью двойных кавычек. Например, поиск термина "Hamlet" может вернуть и пьесу Шекспира Hamlet, и слово hamlet, обозначающее маленькую деревню.

## Опции аргумента поиска

Опции аргумента поиска задаются как свойства аргумента поиска. Например, в следующем запросе поиска для слова bank опции аргумента поиска QUERYLANGUAGE могут отличаться:

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

## Множественные предикаты DB2 Text Search

Если запрос содержит несколько предикатов, учитывайте следующие ограничения, зависящие от организации предикатов.

### Операции UNION и OR

Можно повысить производительность запроса, используя для объединения нескольких предикатов UNION, а не OR.

### Использование операции JOIN

Функции текстового поиска могут быть предикатом во внешнем объединении с ограничением для LEFT OUTER JOIN и FULL OUTER JOIN. В этих случаях предикат текстового поиска может применяться только в том случае, если поиск для этого текстового индекса может быть присоединен обратно с помощью первичного ключа базовой таблицы. Например, поддерживается следующий тип запроса:

```
select place.placenum, location.description from place  
LEFT OUTER JOIN location on (location.mgrid = place.ownerid)  
where  
(location.description is null and contains(place.description, 'Paris')=1 )
```

Функции CONTAINS и SCORE не поддерживаются как предикаты в LEFT OUTER JOIN или FULL OUTER JOIN.

## Локаль и язык текстового поиска DB2

Спецификация локали может повлиять также на выполнение запроса текстового поиска.

### Задание локали

При выполнении поиска по индексу текстового поиска в многоязычной среде рекомендуется всегда использовать в запросе поиска опцию QUERYLANGUAGE, чтобы указать локаль (сочетание информации о языке и территории) для интерпретации термина поиска. Например, если есть термин поиска bald, можно указать рассматривать его как английское слово ("лысый"), задав QUERYLANGUAGE=en\_US в поисковом запросе. Аналогично, если требуется интерпретировать это слово как немецкое ("скоро"), можно задать для QUERYLANGUAGE значение de\_DE. Однако необходимо иметь в виду, что возвращаемые результаты в значительной мере зависят от значения LANGUAGE, используемого для индексирования, независимо от QUERYLANGUAGE, заданного в запросе.

Если в поисковом запросе значение QUERYLANGUAGE не задано, используется следующая логика:

- Терм поиска рассматривается как принадлежащий локали, заданной для соответствующего текстового индекса при создании индекса.
- Если при создании индекса для него была задана локаль AUTO, по умолчанию используется английская локаль (en\_US) и поисковый терм рассматривается как английское слово.

Ограничения:

- Если локаль, заданная в поисковых запросах, недопустима (например, QUERYLANGUAGE=Mongolian), запрос будет считаться недопустимым и возникнет исключительная ситуация.
- Значение QUERYLANGUAGE=AUTO в поисковом запросе не поддерживается, результаты такого запроса не определены.

Обратите внимание на то, что локаль, заданная опцией QUERYLANGUAGE, не влияет на локаль сообщений об ошибках, создаваемых для запросов. Локаль сообщений об ошибках зависит от того, запущены ли службы экземпляра текстового поиска. Если службы не запущены, сообщения создаются с использованием локали en\_US; если службы запущены, сообщения создаются с использованием локали среды, в которой запущена команда **START FOR TEXT**.

## Функция SCORE DB2 Text Search

Оценка документа - динамическая, она вычисляется независимо для каждого запроса.

При изменениях в документе, а также при добавлении и удалении других документов в индексе текстового поиска оценка документа для термина поиска может изменяться.

Пусть есть набор документов о транспорте и загрязнении. Если ищутся документы, содержащие оба термина, но термин загрязнение важнее для вас, чем термин транспорт, можно использовать следующую команду:

```

SELECT id_документа
FROM библиотека_документов
WHERE SCORE(содержимое_документа, 'загрязнение') >
SCORE(содержимое_документа, 'транспорт')
and CONTAINS(document_content, 'транспорт загрязнение') = 1

```

Чтобы повысить производительность, запрос можно сформулировать с использованием модификатора значимости (^) так, что функция поиска будет выполнена только один раз:

```

SELECT id_документа
FROM библиотека_документов
WHERE SCORE(содержимое_документа, 'загрязнение^10 транспорт') > 0
ORDER BY SCORE(содержимое_документа, 'загрязнение^10 транспорт') DESC

```

При низкой оценке загрязнения первый запрос не возвращает никаких результатов. Во втором запросе более высокая важность придается термину загрязнение, но тем не менее он возвращает результаты, даже если оценка термина загрязнение во всех документах низка.

## Функция RESULTLIMIT DB2 Text Search

Чтобы несколько экземпляров **RESULTLIMIT** в одном запросе давали предсказуемые результаты, необходимо, чтобы у них был один и тот же аргумент поиска.

### Описание

Если вы включаете в один запрос несколько текстовых поисков с функцией **RESULTLIMIT**, используйте общий аргумент поиска. Если использовать разные аргументы текстового поиска, результаты могут оказаться непредвиденными.

Например, в следующем запросе **RESULTLIMIT** отнюдь не гарантирует, что будет возвращено именно 10 документов:

```

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

```

Вместо этого используйте **RESULTLIMIT** так:

```

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

```

Заметим, что этот метод работает только тогда, когда обе функции CONTAINS применяются к одному и тому же столбцу таблицы. Если они применяются к разным столбцам, попробуйте для повышения производительности использовать FETCH FIRST *n* ROWS.

## Конфигурирование синтаксического анализа для DB2 Text Search

Можно сконфигурировать некоторые параметры, используемые для поиска XML.

Все параметры конфигурации анализатора расположены в файле parser\_config.xml в элементе XML, определяющем анализатор, com.ibm.es.nuvo.parser.xml.XMLParser. Каждый параметр задан элементом Parameter в следующей форме:

```
<Parameter Name="параметр">значение</Parameter>
```

**ParserName: text**

**ParserClass: com.ibm.es.nuvo.parser.text.TextParser**

Класс, который вызывается, если тип содержимого - текстовый.

**required.text.confidence**

Не используется.

**fall.back.parser**

Синтаксический анализатор, активируемый, если анализатор текста не сработал, указанный тип содержимого неизвестен, а определение содержимого показывает, что содержимое - двоичное.

**fall.back.encoding**

Кодировка, используемая, если указана неизвестная или пустая кодировка.

**detection.encoding.buffer.size**

Размер буфера (в байтах), передаваемый механизму обнаружения контента для определения кодировки. Значение по умолчанию - 2000 байт.

**ParserName: xml****titleTagNameList**

Список тегов через запятую, обрабатываемых как поля заголовков.

**maxTextUnicodeChars**

Не используется.

**handleExternalFiles**

Не используется.

**handleSkippedEntities**

Не используется.

## Пространства имен XML DB2 Text Search

Для поиска в пространствах имен XML требуется обходной прием.

Можно индексировать документы XML, содержащие привязки пространств имен, не создавая ошибок, но информация о пространстве имен удаляется из каждого тега. В результате текстовый поиск по документам XML с привязками пространств имен может привести к нежелательным результатам.

Однако для этого ограничения существует обходной прием для запросов с использованием DB2 XQuery. Механизм текстового поиска DB2 не учитывает пространства имен, но можно воспользоваться поддержкой DB2 XQuery для пространств имен, чтобы отфильтровать по пространствам имен нежелательные документы, возвращаемые при текстовом поиске.

Далее приведен пример, в котором переменная среды базы данных по умолчанию имеет значение SAMPLE и индекс текстового поиска с названием prod\_desc\_idx создается в таблице PRODUCT:

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

Новая строка с пространством имен <http://posample.org/wheelshovel> добавляется в таблицу PRODUCT, в которой уже есть два документа XML с пространством имен <http://posample.org>:

```
INSERT INTO PRODUCT VALUES ('100-104-01', 'Wheeled Snow Shovel',
99.99, NULL, NULL, NULL, XMLPARSE(DOCUMENT '<product xmlns=
"http://posample.org/wheelshovel" pid="100-104-01">
```

```
<description><name>Wheeled Snow Shovel</name><details>
Wheeled Snow Shovel, lever assisted, ergonomic foam grips,
gravel wheel, clears away snow 3 times faster</details>
</price>99.99</price></description></product>'))
```

Затем индекс текстового поиска обновляется следующим образом:

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

Следующее выражение XQuery, где как основной элемент задается `http://posample.org`, возвращает все документы с подходящим путем XPath `/product/description/details` со словом `ergonomic`:

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

Возвращаются три документа. Два из них - те, что вы ожидали, поскольку их пространство имен - `http://posample.org`, а один - нет, так как его пространство имен - `http://posample.org/wheelshovel`.

В следующем выражении XQuery применяется выражение пути `/product/..` для использования поддержки поиска XML и пространств имен DB2 XQuery с целью отфильтровать документы, возвращенные механизмом текстового поиска DB2, чтобы возвращались только документы с пространством имен `http://posample.org`:

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

**Примечание:** В запросах SQL можно использовать DB2 XQuery для принудительной фильтрации по пространствам имен. Для приведенного выше примера соответствующее выражение с использованием запроса SQL будет выглядеть следующим образом:

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

Обходной прием:

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

Аналогично, чтобы получить доступ к конкретному элементу в документе (а не просто получить документ, как в предыдущем запросе), можно использовать следующий запрос:

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

**Примечание:** У этого обходного приема есть ограничения, он может не работать, например, если в документе есть несколько элементов продуктов.



---

## Глава 4. Установка и конфигурирование DB2 Text Search

DB2 Text Search - это необязательный устанавливаемый компонент, установка и конфигурация которого полностью интегрированы с установкой всех продуктов серверов баз данных DB2.

Программа установки DB2 может полностью автоматически установить и сконфигурировать DB2 Text Search. Требуемые шаги отличаются для различных платформ. Раздел рис. 11 на стр. 44 описывает процесс установки и конфигурирования в ОС Windows, а раздел рис. 12 на стр. 45 описывает этот процесс для ОС Linux и UNIX.

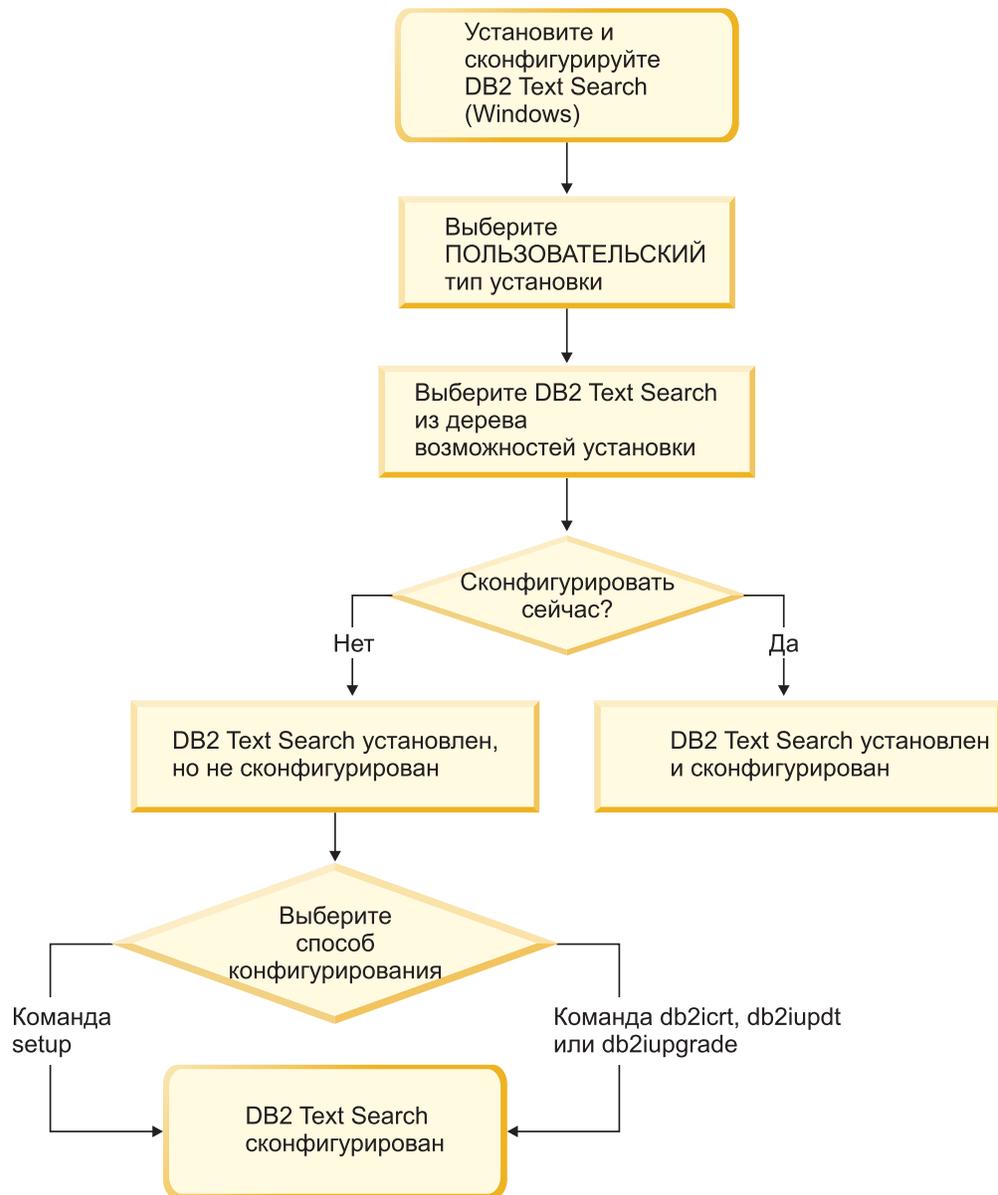


Рисунок 11. Установка и конфигурирование на платформах Windows

В Windows выберите тип установки, решите, требуется ли конфигурирование, и выберите метод конфигурирования.

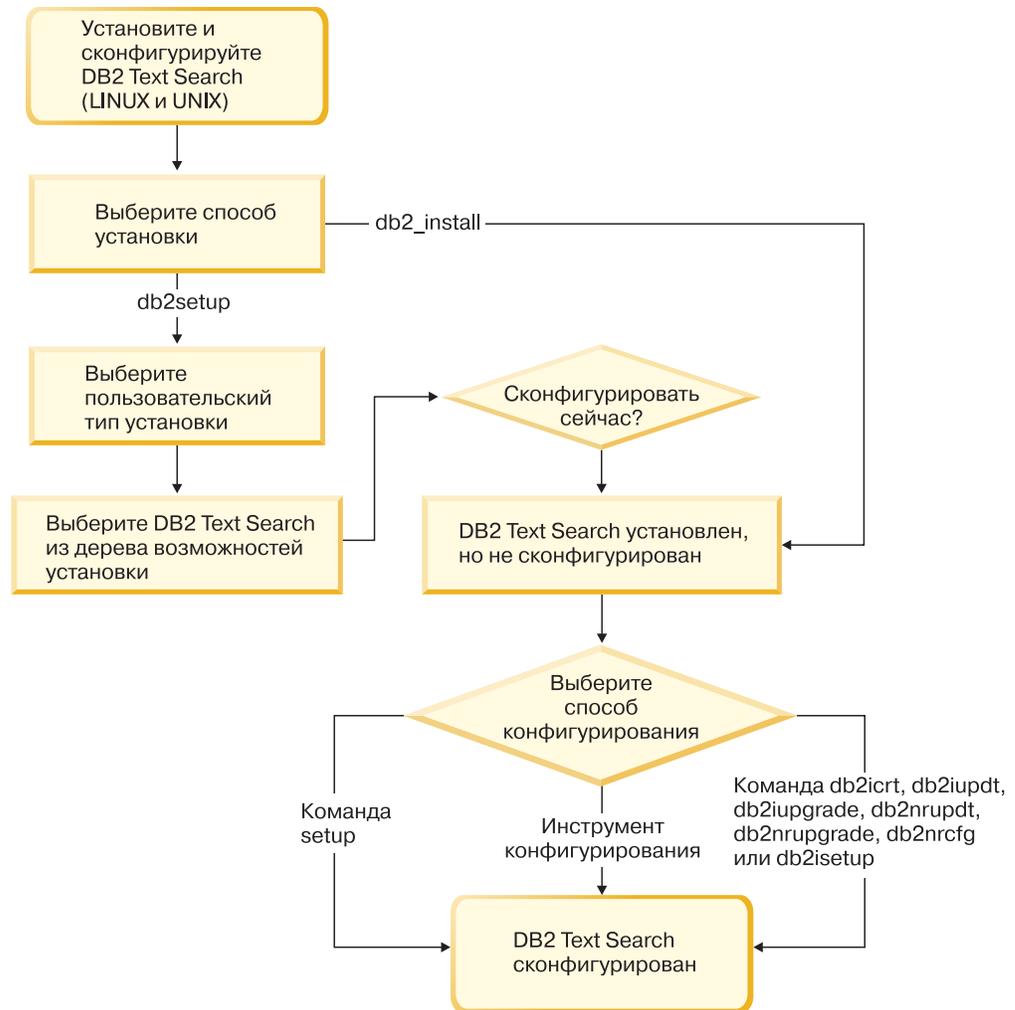


Рисунок 12. Установка и конфигурирование на платформах Linux и UNIX

В Linux и UNIX выберите тип установки, решите, требуется ли конфигурирование, и выберите метод конфигурирования. При запуске **db2setup** от имени пользователя без полномочий root необходимо, чтобы системный администратор (пользователь с полномочиями SYSADM) затем запустил команду **DB2RFE** для резервирования номера порта, который требуется использовать в файле служб.

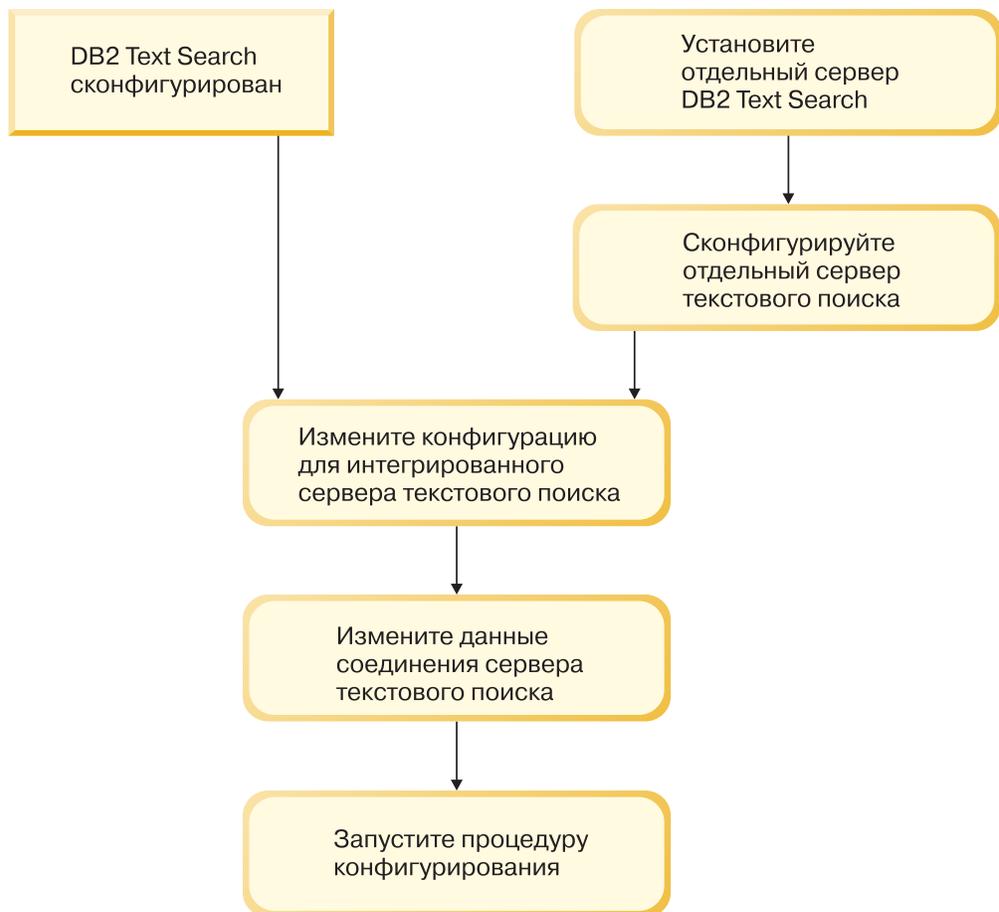


Рисунок 13. Конфигурация автономного сервера текстового поиска DB2

Для автономного сервера текстового поиска DB2 измените конфигурацию интегрированного сервера текстового поиска. Затем обновите данные соединения с сервером и запустите процедуру **CONFIGURE**.

У DB2 Text Search есть следующие ограничения:

- При создании многораздельного экземпляра базы данных с помощью мастера по установке DB2 необходимо находиться на участнике-координаторе или на разделе владельца экземпляра.
- Текстовый поиск DB2 не поддерживается в среде DB2 pureScale.

## Требования к программному и аппаратному обеспечению для DB2 Text Search

### Программные платформы

DB2 Text Search поддерживается на следующих платформах операционных систем:

- AIX Версии 6.1
- HP-UX 11i (платформы HP Integrity Series на основе Itanium)
- Red Hat Enterprise Linux Server 5 (платформы x86 и x64)
- Red Hat Enterprise Linux Server 6 (платформы x86 и x64)
- Solaris 10 (платформы UltraSPARC и x64)

- SUSE Linux Enterprise Server 10 (платформы x86 и x64)
- SUSE Linux Enterprise Server 11 (платформы x86 и x64)
- Windows Server 2003 (платформы x86 и x64)
- Windows Server 2008 (платформы x86 и x64)

**Важное замечание:** В операционных системах Linux должна быть установлена библиотека совместного использования `libstdc++.so.5`.

Автономный сервер DB2 Text Search доступен для всех перечисленных выше платформ, кроме операционных систем HP-UX 11i и Solaris 10 x64. Поддерживается межплатформенное использование, экземпляр базы данных DB2 на этих платформах можно сконфигурировать для использования автономного сервера DB2 Text Search на поддерживаемой платформе.

## Требования к аппаратному обеспечению

Минимальные требования к аппаратному обеспечению для DB2 Text Search:

Таблица 2. Требования к аппаратному обеспечению для DB2 Text Search

Сервер DB2 Text Search	Процессор	ОЗУ / Память	Диск
Конфигурирование интегрированного сервера  В дополнение к требованиям для сервера баз данных DB2)	2 двухъядерный 2.66 ГГц	4 Гбайт	В том числе временное рабочее пространство; каждому индексу текстового поиска требуется объем памяти, примерно в четыре раза превышающий размер всех индексируемых документов. Например, текстовому индексу столбца с миллионом строк по 1 Кбайт текста требуется 4 Гбайт дискового пространства.
Конфигурирование автономного сервера			

Фактическое использование дискового пространства, оперативной памяти и процессора зависит от многих факторов, таких как количество собраний, количество документов в собрании, количество одновременно индексируемых собраний, требуемая пропускная способность индексации и нагрузка запросов. Более подробную информацию смотрите в темах о планировании возможностей DB2 Text Search.

Информацию о рекомендованных ограничениях ресурсов пользовательских процессов операционной системы для операционных систем Linux и UNIX смотрите в теме о требованиях на ограничение пользователей операционных систем. Эти общие требования ограничения ресурсов применяются и к интегрированным, и к автономным серверам DB2 Text Search.

---

## Установка DB2 Text Search с конфигурацией по умолчанию

### Установка и конфигурирование текстового поиска DB2 при помощи мастера по установке DB2

Установить DB2 Text Search можно при помощи мастера по установке DB2 в составе пользовательской установки продукта баз данных DB2.

#### Об этой задаче

Выполните пользовательскую установку продукта баз данных DB2 и выберите в дереве возможностей DB2 Text Search. Возможность DB2 Text Search можно сконфигурировать автоматически либо позднее вручную. Создавать многораздельный экземпляр при помощи мастера по конфигурированию DB2 можно только из участника-координатора экземпляра или из раздела - владельца экземпляра.

#### Процедура

Чтобы выполнить пользовательскую установку DB2 Text Search при помощи **setup** или **db2setup**:

1. Установите сервер DB2, выполнив инструкции для используемой платформы:
  - "Установка серверов DB2 при помощи мастера по установке DB2 (Windows)", *Установка серверов DB2*
  - "Установка серверов DB2 при помощи мастера по установке DB2 (Linux и UNIX)", *Установка серверов DB2*

Вы можете выбрать компонент DB2 Text Search в дереве возможностей. При установке есть возможность сконфигурировать DB2 Text Search для экземпляра по умолчанию. Если вы не хотите конфигурировать DB2 Text Search, пропустите шаг 2.

2. Чтобы сконфигурировать DB2 Text Search самостоятельно, введите правильное имя службы и номер порта, если в этих полях еще нет значений. Конфигурировать возможность DB2 Text Search сразу же после ее установки не обязательно; ее можно сконфигурировать и позднее. Инструкции по конфигурированию после установки смотрите в разделе Глава 5, "Конфигурирование текстового поиска DB2", на стр. 61.

### Установка и конфигурирование DB2 Text Search при помощи файла ответов

Установить и сконфигурировать DB2 Text Search можно в составе пользовательской установки продукта баз данных DB2 в режиме без вывода сообщений. При установке этого типа используется команда **setup** или **db2setup** с файлом ответов.

#### Об этой задаче

Чтобы установить DB2 Text Search, выполните пользовательскую установку продукта баз данных DB2. Для установки и конфигурирования DB2 Text Search в файл ответов нужно добавить ряд ключевых слов.

#### Процедура

Чтобы выполнить пользовательскую установку:

1. Добавьте в файл ответов, используемый для установки продукта баз данных DB2, следующую строку:  
`COMP = TEXT_SEARCH`
2. Чтобы сконфигурировать DB2 Text Search во время установки, добавьте в файл ответов следующую строку:
  - Только для установок с полномочиями root:  
`имя_экземпляра_db2.TEXT_SEARCH_HTTP_SERVICE_NAME = имя_экземпляра_db2_db2j`  
  
где *имя\_экземпляра\_db2* - имя экземпляра DB2, а *имя\_экземпляра\_db2\_db2j* - имя службы.
  - Для установок с полномочиями root и без полномочий root:  
`имя_экземпляра_db2.TEXT_SEARCH_HTTP_PORT_NUMBER = номер-порта`  
Если задать значение для ключевого слова **TEXT\_SEARCH\_HTTP\_SERVICE\_NAME** при установке без полномочий root, будет возвращена ошибка.  
Можно указать любые допустимые свободные имя и номер порта. Если не указать никаких значений, для конфигурации будут использованы значения по умолчанию при условии, что для ключевого слова файла ответов *имя\_экземпляра\_db2.CONFIGURE\_TEXT\_SEARCH* задано значение YES.
3. Установите продукт баз данных DB2, выполнив инструкции для используемой платформы:
  - "Установка продукта DB2 при помощи файла ответов (Windows)" в Установка серверов DB2
  - "Установка продукта (DB2 при помощи файла ответов (Linux и UNIX))" в Установка серверов DB2

## Дальнейшие действия

Конфигурировать возможность DB2 Text Search сразу же после ее установки не обязательно; ее можно сконфигурировать и позднее. Инструкции по конфигурированию после установки смотрите в разделе Глава 5, “Конфигурирование текстового поиска DB2”, на стр. 61.

## Установка текстового поиска DB2 при помощи db2\_install (Linux и UNIX)

При вводе команды **db2\_install** устанавливается также и DB2 Text Search.

### Об этой задаче

**Важное замечание:** Команда **db2\_install** объявлена устаревшей и может быть удалена в следующем выпуске. Используйте вместо нее команду **db2setup**.

Чтобы установить DB2 Text Search, выполните действия, описанные в теме "Установка продукта DB2 при помощи **db2\_install**", *Установка серверов DB2*. Возможность DB2 Text Search будет автоматически установлена в составе установки продукта баз данных DB2.

В случае установки без полномочий root будет создан экземпляр DB2 и установлена возможность DB2 Text Search. В случае установки с полномочиями root нужно будет создать экземпляр DB2 и сконфигурировать DB2 Text Search одним из доступных способов.

Конфигурировать возможность DB2 Text Search сразу же после ее установки не обязательно. Инструкции по выполнению конфигурирования смотрите в разделе

## Установка DB2 Text Search без начального конфигурирования

### Установка серверов баз данных DB2 при помощи мастера по установке DB2 (Windows)

В этом разделе описана процедура запуска мастера по установке DB2 в Windows. Мастер по установке DB2 служит для задания параметров и установки продукта баз данных DB2 в системе.

#### Прежде чем начать

Перед запуском мастера по установке DB2:

- Если вы собираетесь сконфигурировать среду многораздельных баз данных, прочитайте тему "Настройка среды многораздельных баз данных".
- Убедитесь, что в системе выполнены предварительные требования к установке и достаточно оперативной и дисковой памяти.
- Если вы собираетесь использовать LDAP для регистрации сервера DB2 в Active Directory операционных систем Windows, перед установкой расширьте схему каталога, иначе надо зарегистрировать узел и каталогизировать базы данных вручную. Дополнительную информацию смотрите в теме “Расширение схемы Active Directory для служб каталога LDAP (Windows)”.
- Для выполнения установки требуется учетная запись локального администратора с необходимыми правами доступа. На серверах баз данных DB2, где в качестве пользователя экземпляра DAS и DB2 можно использовать LocalSystem и если не используется возможность разделения баз данных, эту установку может выполнить обычный пользователь (не администратор) с расширенными привилегиями.

**Примечание:** Если планируется установка продукта от имени обычного пользователя (не администратора), перед попыткой установки продукта баз данных DB2 надо установить библиотеку времени выполнения VS2010. Библиотека времени выполнения VS2010 должна быть в операционной системе до начала установки продукта баз данных DB2. Библиотеку времени выполнения VS2010 можно получить на сайте загрузки библиотек времени выполнения Microsoft. Существует два варианта: `vcredist_x86.exe` для 32-битных систем и `vcredist_x64.exe` для 64-битных систем.

- Хотя это и не обязательно, рекомендуется закрыть все программы, чтобы программа установки могла изменять любые файлы на компьютере, не требуя перезагрузки.
- Установка продуктов DB2 с виртуального диска или с неотображенного сетевого диска (такого как `\\hostname\sharename` в Windows Explorer) не поддерживается. Перед попыткой установить продукты DB2 products, надо отобразить этот сетевой диск, назначив ему буквы диска Windows (например, Z:).

#### Ограничения

- Под каждой учетной записи пользователя может работать только один экземпляр мастера по установке DB2.
- Имя копии DB2 и имя экземпляра не могут начинаться с цифры. Имя копии DB2 может содержать не более 64 символов; разрешены символы A-Z, a-z и 0-9.
- Имя копии DB2 и имя экземпляра должны быть уникальными среди всех копий DB2.
- Возможности XML можно использовать только для одnorаздельных баз данных.

- Никакой другой продукт баз данных DB2 нельзя установить в тот же самый путь, если в нем уже установлен один из следующих продуктов:
  - IBM® Data Server
  - пакет драйверов IBM Data Server
  - *Информационный центр DB2*
- В поля мастера по установке DB2 нельзя вводить символы национальных алфавитов.
- Если вы включили расширенную защиту в Windows Vista или в Windows 2008, пользователи должны входить в группу DB2ADMNS или DB2USERS, чтобы иметь возможность запускать локальные команды и программы DB2, так как возможность расширенной защиты (User Access Control) ограничивает по умолчанию привилегии локального администратора. Если пользователь не входит в одну из этих групп, у него не будет доступа для чтения локальной конфигурации DB2 или данных прикладных программ.

## Процедура

Для запуска мастера по установке DB2:

1. Войдите в систему с помощью учетной записи локального администратора, созданной для установки DB2.
2. Если у вас есть диск DVD с продуктом баз данных DB2, вставьте его в дисковод. Если включена функция автоматического запуска, появится Панель запуска программы установки DB2. Если функция автоматического запуска не работает, откройте диск DVD продукта баз данных DB2 в Проводнике Windows и дважды щелкните по значку **setup**, чтобы запустить Панель установки DB2.
3. Если вы скачали продукт баз данных DB2 с Passport Advantage, запустите исполняемый файл, чтобы распаковать файлы установки продукта баз данных DB2. Используйте Проводник Windows для просмотра файлов установки DB2 и щелкните дважды по значку **setup** для запуска Панели установки DB2.
4. С Панели запуска установки DB2 можно просмотреть предварительные требования для установки, замечания по выпуску, а также перейти к процедуре установки. Ознакомьтесь со свежей информацией, приведенной в описании предварительных требований к установке и информации о выпуске.
5. Выберите **Установить продукт** - открывается окно Установить продукт со списком доступных для установки продуктов.
 

Если на данном компьютере не установлены продукты баз данных DB2, запустите установку, выбрав **Установить новую**. Выполните установку, следуя подсказкам мастера по установке DB2.

Если на данном компьютере установлен хотя бы один продукт баз данных DB2, можно:

  - Выбрать **Установить новую**, чтобы создать новую копию DB2.
  - Выбрать **Работать с существующей**, чтобы обновить существующую копию DB2, добавить функции к существующей копии DB2, обновить существующую копию DB2 Версии 9.5 и Версии 9.7 или установить дополнительный продукт.
6. Мастер по установке DB2 определяет, какой язык установлен в системе, и запускает программу установки на этом языке. Выполните остальные действия, руководствуясь электронной справкой. Для просмотра электронной справки нажмите кнопку **Справка** или клавишу **F1**. В любой момент можно нажать кнопку **Отмена**, чтобы прервать установку.
7. Примеры панелей при использовании мастера по установке DB2 помогают вам выполнить процесс установки. Смотрите соответствующие ссылки.

## Результаты

Ваш продукт баз данных DB2 устанавливается по умолчанию в каталог *Program\_Files\IBM\sql1lib*, где *Program\_Files* - положение каталога Program Files.

Если при установке в системе этот каталог уже использован, к пути установки продукта баз данных DB2 добавляется *\_xx*, где *xx* - две цифры, начиная с 01 по возрастаную, в зависимости от того, сколько копий DB2 у вас установлено.

Кроме того, вы можете задать свой собственный путь для установки продукта баз данных DB2.

## Дальнейшие действия

- Проверьте свою установку.
- Выполните необходимые задачи после установки.

Информацию об ошибках, возникших во время установки, смотрите в файле журнала установки в каталоге Мои документы\DB2LOG\. Для имени этого файла журнала используется следующий формат: *DB2-краткое\_имя\_продукта-дата\_время.log*, например, *DB2-ESE-Tue Apr 03 17\_04\_45 2012.log*.

Если это новая установка продукта DB2 в 64-битной системе Vista, и вы используете 32-битный провайдер OLE DB, надо вручную зарегистрировать библиотеку DLL IBM DADB2. Чтобы зарегистрировать эту библиотеку DLL, введите команду:  
`c:\windows\SysWOW64\regsvr32 /s c:\Program_Files\IBM\SQLLIB\bin\ibmdadb2.dll`

где *Program\_Files* - положение каталога Program Files.

Чтобы предоставить продукту баз данных DB2 доступ к документации по DB2 на локальном компьютере или на другом компьютере в вашей локальной сети, нужно установить *Информационный центр DB2*. *Информационный центр DB2* содержит документацию по системе баз данных DB2 и связанным с ней продуктами DB2. По умолчанию, если *Информационный центр DB2* не установлен локально, система обращается к информации о DB2 в Web.

IBM Data Studio можно установить при помощи мастера по установке DB2

### Ограничения на память в DB2 Express Edition и DB2 Workgroup Server Edition

При установке DB2 Express Edition максимально допустимый объем памяти для экземпляра - 4 Гбайта.

При установке DB2 Workgroup Server Edition максимально допустимый объем памяти для экземпляра - 64 Гбайт.

Выделяемый для экземпляра объем памяти определяется параметром конфигурации менеджера баз данных **INSTANCE\_MEMORY**.

### Важные замечания при обновлении из Версий 9.5 или 9.7:

- Менеджер автоматической настройки памяти не увеличивает общий предельный объем памяти экземпляров до значения, превышающего лицензионные ограничения.

## Установка серверов DB2 при помощи мастера по установке DB2 (Linux и UNIX)

В этом разделе описана процедура запуска мастера по установке DB2 в операционных системах Linux и UNIX. Мастер по установке DB2 позволяет задать параметры установки и установить продукт баз данных DB2 в системе.

### Прежде чем начать

Перед запуском мастера по установке DB2:

- Если вы собираетесь сконфигурировать среду многораздельных баз данных, посмотрите раздел “Конфигурирование среды многораздельных баз данных” в книге *Установка серверов DB2*
- Убедитесь, что в системе выполнены предварительные требования к установке и достаточно оперативной и дисковой памяти.
- Убедитесь, что у вас установлен поддерживаемый браузер.
- Установку сервера баз данных DB2 можно выполнять с полномочиями root или без полномочий root. Дополнительную информацию об установке без полномочий root смотрите в разделе “Обзор установки без полномочий root (Linux и UNIX)” в книге *Установка серверов DB2*.
- Образ продукта баз данных DB2 должен быть доступен. Вы можете получить образ установки DB2 либо купив физический диск DVD продукта баз данных DB2, либо скачав образ установки с Passport Advantage.
- Для установки неанглийской версии продукта баз данных DB2 нужны соответствующие пакеты для национальных языков.
- Мастер по установке DB2 - это программа установки с графическим интерфейсом. Для запуска мастера по установке DB2 на компьютере должна быть доступна система X windows. Убедитесь, что сервер X windows работает. Убедитесь, что вы правильно экспортировали дисплей. Например, `export DISPLAY=9.26.163.144:0`.
- Если в вашей среде применяется программное обеспечение систем защиты, перед запуском мастера по установке DB2 нужно вручную создать требуемых пользователей DB2.

### Ограничения

- Под каждой учетной записи пользователя может работать только один экземпляр мастера по установке DB2.
- Возможности XML можно использовать только для базы данных, которая определена с кодовым набором UTF-8 и содержит только один раздел базы данных.
- В поля мастера по установке DB2 нельзя вводить символы национальных алфавитов.
- Для HP-UX 11i V2 в системах HP Integrity Series Systems на основе Itanium пользователи, созданные мастером по установке в качестве владельца экземпляра DB2, изолированного пользователя или DAS, недоступны по паролю, заданного в мастере по установке DB2. После завершения работы мастера надо перезадать пароль для этих пользователей. Это не повлияет на создание экземпляра или DAS в мастере по установке, то есть пересоздавать экземпляр или DAS не требуется.

### Процедура

Для запуска мастера по установке DB2:

1. Если у вас есть физический диск DVD с продуктом баз данных DB2, перейдите в каталог, в котором смонтирован диск DVD продукта баз данных DB2, введя команду:

```
cd /dvdrom
```

где */dvdrom* - точка монтирования диска DVD продукта баз данных DB2.

2. Если образ продукта баз данных DB2 загружен по сети, нужно извлечь и распаковать файл tar продукта.

- a. Извлеките файл продукта:

```
gzip -d продукт.tar.gz
```

где *продукт* - имя загруженного вами продукта.

- b. Распакуйте файл tar продукта:

**В операционных системах Linux**

```
tar -xvf продукт.tar
```

**В операционных системах AIX, HP-UX и Solaris**

```
guntar -xvf продукт.tar
```

где *продукт* - имя загруженного вами продукта.

- c. Перейдите в каталог:

```
cd ./продукт
```

где *продукт* - имя загруженного вами продукта.

**Примечание:** Если вы скачали пакет для национальных языков, распакуйте его в тот же каталог. При этом будут созданы подкаталоги (такие как *./nlpack*) в одном каталоге, и программа установки сможет автоматически найти установочные образы, не выводя запрос пользователю.

3. Из каталога, где находится образ продукта баз данных, введите команду **./db2setup** для запуска мастера по установке DB2.
4. Откроется Панель запуска IBM DB2. В этом окне можно просмотреть предварительные требования для установки, информацию о выпуске, а также перейти к процедуре установки. Вы можете также ознакомиться со свежей информацией, приведенной в описании предварительных требований к установке и замечаниях по выпуску.
5. Выберите **Установить продукт** - откроется окно **Установить продукт** со списком доступных для установки продуктов.  
Запустите установку, выбрав **Установить новый**. Выполните установку, следуя подсказкам мастера по установке DB2.
6. Примеры панелей при использовании мастера по установке DB2 помогут вам выполнить процесс установки. Смотрите соответствующие ссылки.  
После запуска программы установки вам потребуется задать необходимые значения на панелях мастера по установке DB2. Выполните остальные действия, руководствуясь справкой по установке. Для просмотра справки нажмите кнопку **Справка** или клавишу F1. В любой момент можно нажать кнопку **Отмена**, чтобы прервать установку.

## Результаты

Для установок без полномочий root продукты баз данных DB2 всегда устанавливаются в каталог *\$HOME/sql1lib*, где *\$HOME* представляет домашний каталог пользователя без полномочий root.

Для установок с полномочиями root продукты баз данных DB2 устанавливаются в один из следующих каталогов:

**AIX, HP-UX и Solaris**

/opt/IBM/db2/V10.1

**Linux** /opt/ibm/db2/V10.1

Если при установке в системе этот каталог уже используется, будет использован путь установки продукта баз данных DB2 с добавлением `_xx` где `xx` - две цифры, начиная с 01 по возрастанию, в зависимости от того, сколько копий DB2 у вас установлено.

Кроме того, вы можете задать свой собственный путь для установки продукта баз данных DB2.

К путям установки DB2 применяются следующие правила:

- Могут содержать латинские буквы нижнего регистра (a-z), верхнего регистра (A-Z) и символ подчеркивания (`_`)
- Не могут быть длиннее 128 символов
- Не могут содержать пробелы
- Не могут содержать символы национальных алфавитов

При установке создаются следующие файлы журналов:

- Файл журнала установки DB2. В этот файл записывается вся информация об установке DB2, включая сообщения об ошибках.
  - Для установок с полномочиями root имя файла журнала установки DB2 - `db2setup.log`.
  - Для установок без полномочий root имя файла журнала установки DB2 - `db2setup_имя-пользователя.log`, где `имя-пользователя` - ID пользователя без полномочий root, под которым выполнена установка.
- Файл журнала ошибок DB2. В этот файл записываются все сообщения об ошибках, возвращаемых Java (например, исключительные ситуации и данные перехвата).
  - Для установок с полномочиями root имя файла журнала ошибок DB2 - `db2setup.err`.
  - Для установок без полномочий root имя файла журнала ошибок DB2 - `db2setup_имя-пользователя.err`, где `имя-пользователя` - ID пользователя без полномочий root, под которым выполнена установка.

По умолчанию эти файлы журналов находятся в каталоге `/tmp`. Можно задать положение файлов журнала.

Файл `db2setup.his` более не используется. Вместо него программа установки DB2 сохраняет копию файла журнала установки DB2 в каталоге `DB2_DIR/install/logs/` и переименовывает ее в `db2install.history`. Если файл с таким именем уже существует, программа установки DB2 переименовывает файл журнала в `db2install.history.xxxx`, где `xxxx` - число от 0000 до 9999, в зависимости от числа установок на данном компьютере.

У каждой копии установки отдельный список файлов хронологии. Если копия установки удаляется, также удаляются и файлы хронологии в ее пути установки. Операция копирования файла журнала в файл хронологии выполняется в конце процесса установки; если программа остановки остановлена или прервана до завершения процесса установки, файл хронологии не создается.

## Дальнейшие действия

- Проверьте свою установку.
- Выполните необходимые задачи после установки.

IBM Data Studio можно установить при помощи мастера по установке DB2

Пакет для национальных языков можно установить также, введя команду `./db2setup` из каталога, в котором находится пакет, после того, как установлен продукт баз данных DB2.

В Linux x86, чтобы предоставить продукту баз данных DB2 доступ к документации по DB2 на локальном компьютере или на другом компьютере в вашей локальной сети, нужно установить *Информационный центр DB2*. *Информационный центр DB2* содержит документацию по системе баз данных DB2 и связанным с ней продуктами DB2.

### Ограничения на память в DB2 Express Edition и DB2 Workgroup Server Edition

При установке DB2 Express Edition максимально допустимый объем памяти для экземпляра - 4 Гбайта.

При установке DB2 Workgroup Server Edition максимально допустимый объем памяти для экземпляра - 64 Гбайт.

Выделяемый для экземпляра объем памяти определяется параметром конфигурации менеджера баз данных `INSTANCE_MEMORY`.

#### Важные замечания при обновлении из Версии 9.5 или 9.7:

- Если объем памяти, сконфигурированный для продукта баз данных DB2 Версии 9.5 или 9.7, превышает допустимый предел, продукт DB2 после обновления в текущую версию может не запуститься.
- Менеджер автоматической настройки памяти не станет увеличивать общий предельный объем памяти экземпляров до значения, превышающего лицензионные ограничения.

## Обзор установки DB2 с помощью файла ответов (Windows)

В Windows можно выполнить установку продукта DB2 с помощью файла ответов на одном или нескольких компьютерах. Установку с помощью файла ответов называют также установкой без вывода сообщений или автоматической установкой.

### Прежде чем начать

Перед началом установки убедитесь, что:

- В системе выполнены требования к памяти, аппаратному обеспечению и программному обеспечению, предъявляемые продуктом DB2.
- У вас есть все необходимые учетные записи пользователей для выполнения установки.
- Все процессы DB2 остановлены.

### Процедура

- Для установки продукта DB2 с помощью файла ответов на одном компьютере:
  1. Создайте и настройте файл ответов, используя один из следующих способов:
    - Изменение примера файла ответов. Примеры файлов ответов находятся в каталоге `db2\Windows\samples`.
    - Использование мастера по установке DB2 для генерации файла ответов.

- Использование генератора файлов ответов.
- 2. Введите команду **setup -u**, задав свой настроенный файл ответов. Например, файл ответов, созданный во время установки:
 

```
setup -u my.rsp
```
- Для установки продукта DB2 с помощью файла ответов на нескольких компьютерах:
  1. Настройте совместный доступ к каталогу.
  2. Создайте файл ответов на основе примера файла ответов.
  3. Установите продукт DB2 при помощи файла ответов.

## Обзор установки DB2 с помощью файла ответов (Linux и UNIX)

В этом разделе описана процедура установки с помощью файла ответов в Linux или UNIX. После начальной установки можно использовать файл ответов для установки дополнительных компонентов или продуктов. Установку с помощью файла ответов называют также установкой без вывода сообщений или автоматической установкой.

### Прежде чем начать

Перед началом установки убедитесь, что:

- В системе выполнены требования к памяти, аппаратному обеспечению и программному обеспечению, предъявляемые продуктом баз данных DB2.
- Остановлены все процессы DB2. При установке продукта баз данных DB2 поверх существующей установки DB2 нужно остановить все прикладные программы DB2, менеджер баз данных DB2 и процессы DB2 для всех экземпляров DB2 и DAS DB2, связанный с существующей установкой DB2.

### Ограничения

При использовании метода установки DB2 с использованием файлов ответов в операционных системах Linux или UNIX необходимо учитывать следующие ограничения:

- Если для любого параметра реестра глобального профиля или экземпляра задано значение BLANK (слово "BLANK"), параметр будет удален из списка заданных в настоящее время параметров.
- Перед установкой убедитесь, что у вас достаточно места на диске. Если этого не сделать, после неудачной установки потребуются ручная очистка.
- При выполнении нескольких установок или установке продукта баз данных DB2 с нескольких дисков DVD рекомендуется производить установку из сетевой файловой системы, а не с дисководов DVD. Установка с сетевой файловой системы требует значительно меньше времени.
- Если вы планируете установку нескольких клиентов, смонтируйте файловую систему на сервере программного кода, чтобы повысить производительность.

### Процедура

Чтобы выполнить установку с помощью файла ответов:

1. Смонтируйте диск DVD продукта баз данных DB2 или перейдите в файловую систему, где хранится установочный образ.
2. Создайте файл ответов на основе примера файла ответов
  - У файла ответов должно быть расширение `.rsp`. Например, `ese.rsp`.

3. Установите DB2 при помощи файла ответов.

---

## Установка и конфигурирование автономного сервера текстового поиска

### Требования к памяти для установки автономного сервера

Для установки автономного сервера текстового поиска требуется по крайней мере 1 Гбайт дискового пространства.

Небольшая часть дискового пространства нужна в дополнение к данным конфигурации для каждого собрания, однако самое большое пространство требуется для данных индекса. Более подробную информацию смотрите в теме об использовании дискового пространства для DB2 Text Search.

Положение файлов данных индекса можно сконфигурировать, используя каталог данных по умолчанию, или задать как параметр каталога собрания при создании текстового индекса.

### Установка автономного сервера текстового поиска DB2

Установку автономного сервера текстового поиска DB2 можно выполнить в режиме без вывода сообщений. Опция установки без вывода сообщений получает входные значения из файла ответов. Установка без вывода сообщений позволяет установить один или несколько серверов. Другое название автономного сервера текстового поиска - сервер текстового поиска ECM.

#### Процедура

Чтобы установить автономный сервер текстового поиска:

1. Создайте пустой каталог установки.
  - В системах Linux или UNIX создайте, например, следующий каталог:  
`/opt/ibm/ECMTextSearch`
  - В системах Windows создайте, например, следующий каталог:  
`C:\Program Files\IBM\ECMTextSearch`Этот каталог называют `<ECMTS_HOME>`.
2. Скачайте DB2 Accessories Suite для вашей платформы с сайта IBM DB2 Accessories Suite для Linux, UNIX и Windows. Извлеките DB2 Accessories Suite во временный каталог.
3. Войдите в систему под именем пользователя с требуемыми полномочиями.
  - В системах Linux и UNIX требуются разрешения на чтение, запись и выполнение для каталога установки.
  - В системах Windows требуются полномочия администратора.
4. Просмотрите лицензию и отредактируйте файл `ecmts_response.txt`, настроив параметры нужным вам образом.
5. Установите автономный сервер текстового поиска при помощи установочного файла `ecmts15_install_<платформа>.exe`. Запустите установку, введя следующую команду:

```
./<ecmts_setup_file_name> -i silent -f ecmts_response.txt
```

В системах Windows введите следующую команду:

```
ecmts15_install_win32.exe -i silent -f ecmts_response.txt
```

6. Убедитесь, что установка выполнена успешно.  
Проверьте файл журнала установки и папки, созданные в каталоге `<ECMETS_HOME>`. Должны существовать как минимум папки `bin`, `lib`, `config` и `resource`.
7. Запустите сервер при помощи сценария запуска из каталога `<ECMETS_HOME>`.
  - На платформах Linux и UNIX:  
`bin/startup.sh`
  - На платформах Windows:  
`bin\startup`
8. Сконфигурируйте свойства сервера текстового поиска. Подробности смотрите в теме о конфигурировании автономного сервера текстового поиска DB2.

## Установка и конфигурирование автономного сервера в качестве службы Windows

Автономный сервер текстового поиска можно установить и сконфигурировать как службу Microsoft Windows.

### Об этой задаче

Служба автономного сервера работает под учетной записью локальной системы, при этом задается автоматический тип запуска. Для службы можно задать имя и указать, будет ли она запускаться после установки автоматически.

### Процедура

Чтобы установить и запускать автономный сервер как службу Windows:

1. Откройте файл ответов `ecmts_response.txt` и задайте следующие параметры:
  - `IA_INSTALL_AS_WINDOWS_SERVICE`  
Задайте для этого параметра значение YES.
  - `IA_WINDOWS_SERVICE_NAME`  
Задайте уникальное имя для службы Windows автономного сервера текстового поиска DB2. Это необязательный параметр.  
Если его значение не задано или задано значение AUTO, этой службе Windows назначается имя по умолчанию ECM Text Search Server. Если указанная служба уже существует, а ее имя не задано, к имени добавляется числовой суффикс, например: ECM Text Search Server1. Если служба с именем, которое вы попытаетесь задать, уже существует, будет возвращено сообщение об ошибке.
  - `IA_START_SERVER`  
Чтобы служба Windows текстового поиска DB2 запускалась после установки автоматически, для параметра `IA_START_SERVER` должно быть задано значение YES. Это необязательный параметр. Значение по умолчанию - YES.
2. Установите автономный сервер текстового поиска. Из каталога, содержащего установочные файлы и файлы ответов, запустите установочный файл, соответствующий вашей операционной системе.
3. Службу Windows текстового поиска можно запустить и остановить с помощью окна служб Microsoft. Чтобы обратиться к окну Службы, откройте панель управления Windows и выберите Администрирование > Службы.

## Деинсталляция автономного сервера текстового поиска DB2

Деинсталляцию автономного сервера текстового поиска DB2 можно выполнить при помощи команды `Uninstall_ECMTextSearch`.

### Прежде чем начать

До начала процесса деинсталляции остановите все службы текстового поиска DB2 и закройте автономный сервер текстового поиска DB2.

### Процедура

Чтобы деинсталлировать автономный сервер текстового поиска DB2:

1. Перейдите в каталог *ECMTS\_HOME*.
2. Запустите деинсталляцию, введя одну из следующих команд, относящихся к конкретным платформам:

- В операционных системах Linux и UNIX:

```
INSTALL_DIR/Uninstall_ECMTextSearch/Uninstall_ECMTextSearch -i silent
```

- В операционных системах Windows:

```
ECMTS_HOME\Uninstall_ECMTextSearch\Uninstall_ECMTextSearch.exe -i silent
```

Программа деинсталляции удаляет не все данные из каталога *ECMTS\_HOME*.

Например, после выполнения программы деинсталляции остается файл `uninstall.log`. Некоторые из следующих каталогов или даже все они могут не быть удалены программой установки, и их нужно удалить вручную:

- *ECMTS\_HOME\config*
- *ECMTS\_HOME\license*
- *ECMTS\_HOME\log*
- *ECMTS\_HOME\resource*
- *ECMTS\_HOME\temp*
- *ECMTS\_HOME\Uninstall\_ECMTextSearch*

**Совет:** Полезно создать резервную копию данных собраний и конфигурирования, хранящихся в каталоге *ECMTS\_HOME\config* для будущего использования.

### Результаты

Сервер текстового поиска DB2 деинсталлирован, и его больше нельзя использовать ни для администрирования индексов текстового поиска, ни для выполнения полнотекстовых запросов. Однако данные собраний и конфигурирования остались нетронутыми.

---

## Глава 5. Конфигурирование текстового поиска DB2

Опции для конфигурирования возможности DB2 Text Search зависят от того, выполняется ли исходное конфигурирование или переконфигурирование, и от используемой платформы.

### Прежде чем начать

Перед переконфигурированием DB2 Text Search остановите службу экземпляра текстового поиска, как описано в разделе “Остановка службы экземпляра текстового поиска DB2” на стр. 82.

Для многораздельных экземпляров при использовании инструмента конфигурирования требуется находиться на участнике-координаторе или в разделе - владельце экземпляра. Это хост экземпляра, где была создана исходная конфигурация интегрированного сервера текстового поиска, с наименьшим номером раздела хоста сервера.

### Процедура

- Определите, сконфигурирована ли возможность DB2 Text Search.  
Запустите инструмент конфигурирования, введя следующую команду:  

```
configTool printAll -configPath абсолютный-путь-к-папке-конфигураций
```

В выводе опции **printAll**, если возможность DB2 Text Search не сконфигурирована, маркер аутентификации - это пустая строка.
- Выполните исходное конфигурирование возможности DB2 Text Search.  
В операционных системах Linux и UNIX используйте для конфигурирования DB2 Text Search один из следующих способов:
  - Повторите установку без вывода сообщений, как описано в разделе “Установка и конфигурирование DB2 Text Search при помощи файла ответов” на стр. 48.
  - Повторите установку из графического пользовательского интерфейса, как описано в разделе “Установка и конфигурирование текстового поиска DB2 при помощи мастера по установке DB2” на стр. 48.
  - Используйте инструмент конфигурирования. Смотрите раздел “Начальное конфигурирование интегрированного сервера текстового поиска DB2” на стр. 63. Учтите, что для выполнения конфигурирования вручную при помощи инструмента конфигурирования большинство параметров нужно сконфигурировать вручную, тогда как при использовании программы установки нужно сконфигурировать всего два параметра.
  - Для конфигурирования текстового поиска DB2 в зависимости от типа экземпляра и операций используйте одну из следующих команд:
    - Для установок с полномочиями root в графическом пользовательском интерфейсе можно ввести команду **db2isetup**, чтобы сконфигурировать существующий экземпляр DB2, выбрав Текстовый поиск DB2 при его конфигурировании. Можно также ввести команду **db2iupdt** с опцией -j, чтобы сконфигурировать интегрированный сервер текстового поиска DB2. Учтите, что при создании экземпляра с помощью команды **db2icrt** с опцией -j будет также сконфигурирован по умолчанию и текстовый поиск DB2.
    - Для установок без полномочий root введите команду **db2isetup**, чтобы сконфигурировать экземпляр в графическом пользовательском интерфейсе (либо введите команду **db2nrupdt** или **db2nrupgrade** с опцией -j).

В операционных системах Windows используйте для конфигурирования возможности DB2 Text Search один из следующих способов:

- Повторите установку без вывода сообщений, как описано в разделе “Установка и конфигурирование DB2 Text Search при помощи файла ответов” на стр. 48.
- Повторите установку из графического пользовательского интерфейса, как описано в разделе “Установка и конфигурирование текстового поиска DB2 при помощи мастера по установке DB2” на стр. 48.
- Введите команду **db2iupdt** с опцией **-j**. Имейте в виду, что при создании экземпляра при помощи команды **db2icrt** с опцией **-j** будет также сконфигурирован по умолчанию и текстовый поиск DB2.

- Определите, используется ли Java Developer Kit разработки IBM.

DB2 Text Search использует для внутренней обработки Java Developer Kit, на положение которого указывает параметр **JDK\_PATH** команды **db2 get dbm cfg**, и этот Java Developer Kit должен быть разработки IBM. Чтобы проверить, используется ли Java Developer Kit разработки IBM, введите следующую команду:  
`JDK_PATH/jre/bin/java -version`

Эта команда выведет информацию о версии Java, и если используется Java Developer Kit разработки IBM, в строке будет указана IBM.

- Переконфигурируйте DB2 Text Search.

Для переконфигурирования уже установленной возможности DB2 Text Search программу установки из графического интерфейса использовать нельзя. Все изменения в конфигурацию нужно вносить вручную.

В операционных системах Linux и UNIX используйте для переконфигурирования DB2 Text Search один из следующих способов:

- Повторите установку без вывода сообщений, как описано в разделе “Установка и конфигурирование DB2 Text Search при помощи файла ответов” на стр. 48.
- Используйте инструмент конфигурирования. Смотрите раздел “Начальное конфигурирование интегрированного сервера текстового поиска DB2” на стр. 63.
- Для переконфигурирования текстового поиска DB2 в зависимости от типа экземпляра и операций, используйте одну из следующих команд:
  - Для установок с полномочиями root в графическом пользовательском интерфейсе можно ввести команду **db2i setup**, чтобы сконфигурировать существующий экземпляр DB2, выбрав конфигурируемый текстовый поиск DB2. Можно также ввести команду **db2iupdt** с опцией **-j**, чтобы сконфигурировать интегрированный сервер текстового поиска DB2.
  - При установках без полномочий root для конфигурирования экземпляра в графическом пользовательском интерфейсе введите команду **db2i setup**, либо введите команду **db2nrupdt** или **db2nrupgrade** с опцией **-j**.

В операционных системах Windows для переконфигурирования возможности DB2 Text Search используйте один из следующих способов:

- Повторите установку без вывода сообщений, как описано в разделе “Установка и конфигурирование DB2 Text Search при помощи файла ответов” на стр. 48.
- Используйте инструмент конфигурирования. Смотрите раздел “Начальное конфигурирование интегрированного сервера текстового поиска DB2” на стр. 63.
- Введите команду **db2iupdt** или **db2iupgrade**, задав, как показано, опцию **-j** нужным вам образом:

- Команда -j "TEXT\_SEARCH" пытается сконфигурировать DB2 Text Search с именем службы по умолчанию и сгенерированным значением для номера порта.
- Команда -j "TEXT\_SEARCH, [имя\_службы]" резервирует указанное имя службы с автоматически сгенерированным номером порта или с одним и тем же номером порта, назначенным для этого имени службы, если имя службы уже зарезервировано в файле служб.
- Команда -j "TEXT\_SEARCH, [номер\_порта]" резервирует порт с именем службы по умолчанию.
- Команда -j "TEXT\_SEARCH, [имя\_службы], [номер\_порта]" резервирует указанные имя службы и номер порта.

**Примечание:** В операционных системах Windows параметр **PATH** в командном окне DB2 указывает на *путь-установки-текущей-копии-по-умолчанию\db2tss\bin*, поэтому, чтобы сконфигурировать экземпляр, находящийся не в текущей копии DB2, переключитесь сначала на командное окно DB2, соответствующее нужной копии.

---

## Начальное конфигурирование интегрированного сервера текстового поиска DB2

Инструмент конфигурирования - это инструмент командной строки, который можно использовать для создания исходной конфигурации DB2 Text Search или для изменения текущей конфигурации.

### Прежде чем начать

Для настройки большинства параметров конфигурации требуется остановить службы экземпляра DB2 Text Search.

### Об этой задаче

Начальное конфигурирование после установки лучше всего выполнить при помощи программы установки DB2. Для начального конфигурирования вручную так же, как и для обновлений конфигурации, нужно использовать инструмент конфигурирования.

### Процедура

Для создания исходной конфигурации сервера текстового поиска DB2 используется следующая процедура. (Дополнительные подробности смотрите в теме об инструменте конфигурирования.)

1. Введите команду **configTool** с опцией **configureParams**, чтобы задать путь конфигурации.
  - Просмотрите следующие опции и измените значения по умолчанию нужным вам образом.
    - defaultDataDirectory**: положение собраний текстовых индексов (каждое собрание хранится в своем собственном подкаталоге).
    - logPath**: положение файлов журналов и трассировки сервера текстового поиска.
    - tempDirPath**: путь к временному каталогу.
    - installPath**: путь к каталогу установки текстового поиска DB2 (каталог DB2PATH\db2tss в Windows и каталог *DB2DIR/db2tss* в Linux и UNIX, где *DB2DIR* - положение копии DB2).

**-startupHeapSize:** максимальный размер кучи сервера текстового поиска.

Например, чтобы сконфигурировать опции **defaultDataDirectory** и **installPath**, введите следующую команду:

```
configTool configureParams -configPath <абсолютный_путь_к_папке_конфигурации>
-defaultDataDirectory dataPath -installPath ipath
```

- В операционных системах Windows введите команду указанным ниже способом. Необходимо задать только **configPath**; все остальные параметры - это пути и значения, назначаемые по умолчанию.

```
configTool
-configPath абсолютный-путь-к-папке-config
```

2. Текстовый поиск DB2 выполняет аутентификацию требований администрирования индексов текстового поиска и требований текстового поиска при помощи маркера аутентификации. Сгенерируйте маркер аутентификации, введя команду **configTool** с опцией **generateToken** следующим образом:

```
configTool generateToken
-configPath абсолютный-путь-к-папке-config
-seed значение
```

3. Задайте порт HTTP, введя команду **configTool** с опцией **configureHTTPListener** следующим образом:

```
configTool configureHTTPListener
-configPath абсолютный-путь-к-папке-config
-adminHTTPPort номер-порта
```

**Примечание:** Можно использовать значение номера порта от 1024 до 65535. Управляющий порт HTTP позволяет устанавливать связь между процессами текстового поиска при помощи TCP/IP. При установке продукта баз данных DB2 или создании экземпляра, если у вас есть полномочия root, можно задать имя службы и порт. Они будут использоваться для обновления файла служб.

4. Обновите файл служб.

Смотрите раздел “Обновление файла служб на сервере для соединений TCP/IP” на стр. 67.

Если для конфигурирования используется инструмент конфигурирования, он не обновляет файл служб. Поэтому вы должны обновить файл служб вручную.

**Примечание:** Обновлять файл служб могут только пользователи root. Пользователи без полномочий root должны сначала попросить администратора системы ввести команду **db2rfe**.

---

## Обновление информации о сервере текстового поиска DB2

Информация о сервере текстового поиска DB2 используется в базе данных для соединения с сервером текстового поиска с целью администрирования индексов текстового поиска и выполнения в них поиска. Поэтому в управляющей производной таблице SYSIBMTS.TSSERVERS каталога текстового поиска должны быть заданы допустимые значения параметров, гарантирующие правильность работы системы.

### Прежде чем начать

Для обновления информации о сервере текстового поиска требуется роль SYSTS\_ADM и привилегии DBADM для заданной базы данных.

## Об этой задаче

В информацию о сервере входит, главным образом, информация о соединении, такая как имя хоста, значение маркера и номер порта сервера, а также характеристики сервера, такие как локаль сервера, указание, включена ли для конфигурации текстового поиска поддержка формата расширенного текста и указание представляет ли собой сервер поиска, используемый экземпляром DB2, интегрированную установку (skonfigурированную DB2 как часть экземпляра DB2) или отдельную автономную установку сервера текстового поиска.

Исходно обновление информации требуется в следующих сценариях:

- при включении для базы данных поддержки текстового поиска появляется сообщение с предупреждением о неполном включении поддержки
- исходное конфигурирование автономного сервера текстового поиска
- многораздельные базы данных
- обновления DB2 Text Search
- использование хранимых процедур для администрирования с компьютера клиента
- и в дальнейшем, после внесения любых изменений в информацию о соединении с сервером текстового поиска.

При включении поддержки базы данных управляющая производная таблица SYSIBMTS.TSSERVERS обновляется исходной информацией для интегрированного сервера, если авторизация, необходимая для обращения к конфигурации, доступна. Просмотрите в SYSIBMTS.TSSERVERS информацию о сервере текстового поиска, обновите ее нужными данными о сервере текстового поиска и вызовите процедуру SYSTS\_CONFIGURE, чтобы применить обновленную информацию. При наличии в экземпляре нескольких баз данных skonfigурируйте каждую базу с учетом информации одного сервера текстового поиска.

Если требуется переконфигурирование, убедитесь, что никакие административные операции текстового поиска не активны, и закройте сервер текстового поиска перед применением каких-либо изменений.

Нужно изменить некоторые аспекты, относящиеся к установке текстового поиска и к конфигурации экземпляра DB2 для текстового поиска. К ним относятся:

- Указание, представляет ли собой сервер поиска, используемый экземпляром DB2, интегрированную установку (skonfigурированную DB2 как часть экземпляра DB2) или отдельную автономную установку сервера текстового поиска.
- Используется указатель, включена ли при конфигурировании текстового поиска поддержка расширенного текста.

## Процедура

Чтобы обновить информацию о сервере текстового поиска DB2:

1. Получите необходимые значения свойств сервера текстового поиска, такие как имя хоста и номер порта, введя команду **configTool** с опцией **printAll**.  
Дополнительные подробности смотрите в теме о **configTool**.
2. Просмотрите записи в управляющей производной таблице SYSIBMTS.TSSERVERS и внесите все необходимые изменения:
  - Если производная таблица пустая, используйте оператор INSERT. Например:

```
INSERT INTO SYSIBMTS.TSSERVERS (HOST, PORT, TOKEN, SERVERSTATUS)
VALUES ('localhost', 55000, '9kfsjg48=', 0);
```

- Если указанная производная таблица содержит строку, используйте оператор UPDATE. Например:

```
UPDATE SYSIBMTS.TSSERVERS SET (HOST, PORT, TOKEN)
= ('tsmach1.ibm.com', 55002, 'k3j4fjk9u=')
```

В случае использования нескольких разделов базы данных или выполнения административных операций с клиента обязательно используйте вместо локального хоста фактическое имя хоста или IP-адрес. Это относится не только к локальным установкам автономного сервера текстового поиска, но и к интегрированным серверам.

3. Выполните процедуру **SYSTS\_CONFIGURE**. Дополнительные подробности смотрите в теме о процедуре SYSTS\_CONFIGURE.
4. Убедитесь, что значения в управляющей производной таблице SYSIBMTS.TSSERVERS совпадают со значениями, возвращаемыми инструментом конфигурирования.
5. Запустите службу текстового поиска, чтобы проверить возможность обращения к серверу тестового поиска.

---

## Конфигурирование автономного сервера DB2 Text Search

Используйте инструмент конфигурирования для настройки некоторых свойств по умолчанию после установки автономного сервера DB2 Text Search. Можно сконфигурировать соответствующие свойства уровня системы и свойства защиты в вашей системе.

Перед тем, как сконфигурировать свойства, убедитесь, что автономный сервер DB2 Text Search закрыт и службы текстового поиска остановлены. Не перезапускайте сервер текстового поиска, пока не завершите конфигурирование автономного сервера текстового поиска и требуемые обновления конфигурации используемых баз данных в соответствующем экземпляре DB2.

Вы можете воспользоваться инструментом конфигурирования, чтобы просматривать свойства сервера текстового поиска, несмотря на то, что этот сервер остановлен.

### Конфигурирование системы

Обязательно пересмотрите и сконфигурируйте при помощи инструмента конфигурирования, как минимум, следующие свойства:

- **configureHTTPListener**: Конфигурирует порт и имя хоста сервера DB2 Text Search
- **generateToken**: Генерирует маркер аутентификации и ключ шифрования
- **defaultDataDirectory**: Конфигурирует параметры для собрания

**Напоминание:** Если значение для *configPath* содержит пробелы, такое значение надо заключить в кавычки.

Подробности и информацию о дополнительном необязательном конфигурировании смотрите в разделе об инструменте конфигурирования для DB2 Text Search.

### Конфигурирование защиты

Каждое Требование API от сервера базы данных DB2 автономному серверу DB2 Text Search аутентифицируется маркером аутентификации. Начальный маркер генерируется при установке автономного сервера текстового поиска.

1. Используйте инструмент конфигурирования, чтобы явным образом задать начальное значение и сгенерировать маркер аутентификации. Максимальная длина строки маркера - 32 байта.
2. Запустите инструмент конфигурирования для экземпляра DB2, чтобы задать соответствующее значение маркера.
3. Сохраните информацию о соединении, включая маркер, в управляющей производной таблице SYSIBMTS.TSSERVER для каждой используемой базы данных.

При помощи инструмента конфигурирования DB2 Text Search можно показать текущие значения маркера аутентификации и ключа шифрования. Но невозможно узнать начальное значение, используемое автономным сервером DB2 Text Search. Сгенерируйте маркер явным образом при помощи утилиты **configTool** и обновите главную конфигурацию в экземпляре DB2, чтобы сконфигурированные значения соответствовали маркеру.

Чтобы сконфигурировать свойства сервера текстового поиска, запустите инструмент конфигурирования командой для соответствующей платформы:

- На платформах Linux и UNIX:  
`configTool.sh команда_конфигурирования -configPath значение_пути [-locale значение_локали] -аргументы_конкретной_команды`
- На платформах Windows:  
`configTool.bat команда_конфигурирования -configPath значение_пути [-locale значение_локали] -аргументы_конкретной_команды`

Например, для печати текущего маркера аутентификации на сервере Linux служит такая команда:

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

**Примечание:** Для автономного сервера DB2 Text Search на платформах Linux и UNIX в команду конфигурирования следует задавать полностью, включая суффикс `.sh`. Только интегрированный сервер DB2 Text Search поддерживает имена сценариев без суффикса.

---

## Обновление файла служб на сервере для соединений TCP/IP

Эта задача - часть большей задачи *Конфигурирование связи TCP/IP для экземпляра DB2*.

### Об этой задаче

В файле служб TCP/IP заданы номера портов, которые могут применяться программами серверов для получения запросов от клиентов. Если в поле **svcsname** файла конфигурации менеджера баз данных было задано имя службы, то в файле служб необходимо задать номер порта или протокол, соответствующий этому имени службы. Если в поле **svcsname** файла конфигурации менеджера баз данных был задан номер порта, то файл служб обновлять *не* нужно.

Укажите в файле `services` порты, через которые сервер принимает входящие требования от клиентов. Положение по умолчанию файла `services` зависит от операционной системы:

**Операционные системы Linux и UNIX**  
`/etc/services`

## Операционные системы Windows

%SystemRoot%\system32\drivers\etc\services

### Процедура

С помощью текстового редактора добавьте запись для соединения в файл служб. Например:

```
db2c_db2inst1 3700/tcp # порт службы соединения DB2
```

где:

**db2c\_db2inst1**

имя службы соединения

**3700**

номер порта соединения

**tcp**

используемый протокол связи

---

## Установка DB2 Accessories Suite для DB2 Text Search

DB2 Accessories Suite включает индексирование и поиск для документов в расширенном текстовом формате и в собственных форматах при помощи DB2 Text Search. Можно запустить новую установку или выполнить установку поверх существующей установки.

### Прежде чем начать

Чтобы установить DB2 Accessories Suite для Linux и UNIX, нужно войти в систему сервера DB2 под именем системного администратора. Для Windows нужно войти в систему как пользователь с полномочиями локального администратора.

Скачайте DB2 Accessories Suite. Ссылку для скачивания смотрите в теме <https://www.ibm.com/services/forms/preLogin.do?source=swg-dm-db2accsuite> Установите самую свежую версию выпуска или пакета Fix Pack DB2 Accessories Suite, чтобы обеспечить правильную работу этой возможности.

Убедитесь, что файл программы установки, файл лицензий и файл информации о выпуске находятся в одном и том же каталоге.

### Процедура

Чтобы установить DB2 Accessories Suite:

1. Остановите службу экземпляра текстового поиска DB2. Для остановки этой службы введите команду **db2ts STOP FOR TEXT**.
2. Войдите в систему сервера баз данных DB2 как пользователь с необходимыми разрешениями, у которого есть привилегия на запись в каталоге установки сервера текстового поиска DB2 (например, на платформе Linux этот каталог находится в каталоге <DB2PATH>/db2tss, где <DB2PATH> - каталог установки сервера баз данных DB2).
3. Есть два режима установки. Один из них - установка с консоли, другой - установка без вывода сообщений.
  - Чтобы выполнить установку с консоли:
    - а. Запустите программу установки фильтров Accessories Suite.
      - Для платформ Linux и UNIX запустите программу установки **installAccSuiteV10.bin** из командной строки.

- На платформе Windows есть два подхода.
  - Запустить программу установки **installAccSuiteV10.exe** из командного окна.
  - Щелкнуть дважды кнопкой мыши по двоичному файлу программы установки.
- b. Приняв условия лицензии, введите положение подкаталога /db2tss в последней копии DB2, когда система запросит путь установки.
- c. Каталог db2tss уже должен существовать. Если он отсутствует, это значит, что DB2 Text Search не был правильно установлен и сконфигурирован.
- d. Проверьте сводку данных и подтвердите установку.
- Чтобы выполнить установку без вывода сообщений:
  - a. Измените файл ответов, задав для параметра **LICENSE\_ACCEPTED** значение **true** и назначив правильный полный путь установки **USER\_INSTALL\_DIR**, который должен содержать каталог db2tss.
  - b. Запустите программу установки фильтров Accessories Suite в режиме без вывода сообщений.
    - На платформе Linux или UNIX введите команду **installAccSuiteV10.bin -i silent -f installer.properties** в командной строке.
    - На платформе Windows введите команду **installAccSuiteV10.exe -i silent -f installer.properties** в командном окне.

## Результаты

Вы успешно установили DB2 Accessories Suite.

## Дальнейшие действия

Теперь вы можете включить поддержку документов в расширенном текстовом формате для DB2 Text Search. Подробности смотрите в теме “Включение поддержки документов в расширенном текстовом формате для DB2 Text Search” на стр. 80.

---

## Деинсталляция DB2 Accessories Suite для DB2 Text Search

Деинсталлировать автономную систему DB2<sup>®</sup> Text Search можно при помощи команды **Uninstall\_DB2AS**.

### Прежде чем начать

Чтобы деинсталлировать DB2 Accessories Suite для Linux и UNIX, необходимо зарегистрироваться на сервере DB2 в качестве системного администратора. Для платформах Windows вы должны зарегистрироваться как пользователь с полномочиями локального администратора.

### Процедура

Чтобы деинсталлировать DB2 Accessories Suite:

1. Остановите службу экземпляра текстового поиска DB2. Для остановки службы введите команду db2ts "STOP FOR TEXT".
2. Зарегистрируйтесь на сервере баз данных DB2 как пользователь с достаточными привилегиями операционной системы.

3. Отключите поддержку расширенных текстовых документов для всех экземпляров текстового поиска, для которых она была включена ранее. Подробности смотрите в теме об отключении поддержки расширенных текстовых документов в DB2 Text Search.
4. Деинсталлируйте программу установки DB2 Accessories Suite. Для деинсталляции программы установки:
  - В операционных системах Linux и UNIX:  
`<DB2DIR>/db2tss/Uninstall_DB2ASV10/Uninstall_DB2AS.bin`  
  
где `<DB2DIR>` - положение последней копии DB2.
  - В операционной системе Windows:  
`<DB2PATH>\db2tss\Uninstall_DB2ASV10\Uninstall_DB2AS.exe`  
  
, где `<DB2PATH>` - это положение, где установлена последняя копия DB2.

## Результаты

Вы деинсталлировали DB2 Accessories Suite.

---

## Глава 6. Обновление текстового поиска DB2

---

### Обновление DB2 Text Search для установок от имени администратора или пользователя root

Чтобы получить самые последние функциональные возможности, обновите свой экземпляр DB2 Text Search. При обновлении экземпляра текстового поиска вам нужно обновить сервер DB2, экземпляр DB2 и все базы данных.

#### Об этой задаче

Перечисленные ниже действия представляют процесс обновления установок от имени пользователя root DB2 Text Search Версии 9.5 или Версии 9.7 для платформ Linux или UNIX или от имени администратора для платформы Windows.

#### Процедура

1. Войдите как пользователь - владелец экземпляра или пользователь с полномочиями SYSADM.
2. Остановите экземпляр баз данных DB2 и службу экземпляра DB2 Text Search.
3. Создайте резервную копию каталога конфигурации DB2 Text Search:
  - Для операционных систем Linux и UNIX это каталог `$INSTHOME/sql1lib/db2tss/config`
  - Для систем Windows это каталог `<INSTPROF>\<INSTNAME>\db2tss\config`

где *INSTHOME* - путь к домашнему каталогу экземпляра.

  - Для систем Windows это каталог `<INSTPROF>\<INSTNAME>\db2tss\config`

где *<INSTPROF>* представляет каталог профиля экземпляра, а *<INSTNAME>* указывает имя экземпляра, который следует обновить.
4. Если для DB2 Text Search включена поддержка документов в расширенном текстовом формате и вы выполняете обновление из DB2 Версии 9.5 Fix Pack 3 или более новой, отключите поддержку документов в расширенном текстовом формате. Дополнительную информацию об отключении поддержки документов в расширенном текстовом формате смотрите в теме об отключении поддержки расширенных текстовых документов в DB2 Text Search.
5. Войдите на сервер DB2 как пользователь с полномочиями root в операционных системах Linux и UNIX или с полномочиями локального администратора в операционных системах Windows. Если вы обновляете многораздельный экземпляр, обновление экземпляра надо выполнять из раздела - владельца экземпляра.
6. Установите новую копию DB2 Enterprise Server Edition для Linux, UNIX и Windows Версии 10.1, выбрав пользовательскую установку, и убедитесь, что выбран текстовый поиск DB2. Текстовый поиск DB2 - это дополнительный компонент, который доступен только при выборе пользовательской установки. Можно также выбрать вариант установки новой копии DB2 Версии 10.1 поверх более ранней копии DB2, выбрав режим Работать с существующим и текстовый поиск DB2 в качестве обновляемого компонента. После установки с применением этого подхода обновлять экземпляры DB2 не требуется.
7. Обновите сервер DB2 Text Search для ваших экземпляров DB2 с помощью команды **configTool upgradeInstance**.

- Для операционных систем Linux и UNIX:  
`$DB2DIR/db2tss/bin/configTool upgradeInstance`  
`-installedConfigPath $DB2DIR/cfg/db2tss/config`  
`-configPath $INSTHOME/sql1lib/db2tss/config`

где *INSTNAME* - домашний каталог экземпляра, а *DB2DIR* - положение только что установленной копии DB2 Версии 10.1.

- Для операционных систем Windows:  
`<DB2PATH>\db2tss\bin\configTool upgradeInstance`  
`-installedConfigPath "<DB2PATH>\CFG\DB2TSS\CONFIG"`  
`-configPath "<INSTPROFDIR>\<INSTANCENAME>\DB2TSS\CONFIG"`

где *<DB2PATH>* - положение только что установленной копии DB2 Версии 10.1, а *<INSTPROFDIR>* - каталог профиля экземпляра.

**Примечание:** Для систем Windows, если экземпляр DB2 ранее не конфигурировался для текстового поиска DB2 и обновляется DB2 Версии 9.7 с пакетом Fix Pack 1 или новее, этот шаг можно пропустить.

Команда **configTool upgradeInstance** заменяет, изменяет и объединяет файлы и каталоги конфигурации и данных.

#### Каталог config

Эта команда копирует в каталог *<ECMTS\_HOME>\config* следующие файлы (если они еще не существуют):

- constructors.xml
- ecmts\_logging.properties
- ecmts\_config\_logging.properties
- mimetypes.xml
- monitoredEventsConfig.xml

Следующие файлы копируются (а все существующие файлы перезаписываются):

- build\_info.properties
- constructors.xsd
- ecmts\_config\_logging.properties

Следующие файлы сливаются. Новые значения параметров добавляются, а существующие значения параметров сохраняются.

- config.xml
- jetty.xml

Следующие файлы не изменяются:

- authentication.xml
- key.txt
- Все файлы в подкаталоге собраний

#### Каталог журналов

Эта команда не изменяет содержание существующего каталога журналов. Однако при генерировании новых файлов журнала они могут заменить существующие файлы журнала.

Команда **configTool upgradeInstance** не обновляет фильтры текстового поиска для интегрированного сервера текстового поиска.

8. Обновите текущий экземпляр DB2, введя команду **db2i upgrade**.

- Для операционных систем Linux и UNIX эта команда находится в каталоге `$DB2DIR/instance`, где `DB2DIR` - положение только что установленного сервера баз данных DB2 Версии 10.1.

```
db2iupgrade -j "TEXT_SEARCH [[имя-службы] | [,номер-порта]]" DB2INST
```

- Для операционных систем Windows файл свойств находится в каталоге `<DB2PATH>\bin`, где `<DB2PATH>` - положение только что установленной копии DB2 Версии 10.1.

```
db2iupgrade DB2INST /j "TEXT_SEARCH [[,имя-службы] | [,номер-порта]]"
```

Дополнительную информацию смотрите в теме о команде **db2iupgrade**.

**Примечание:** Если устанавливается новая копия DB2 Версии 10.1 с опцией обновления, выберите в качестве компонента для обновления текстовый поиск DB2; затем этот шаг можно будет пропустить.

9. Создайте резервные копии значений для всех конфигурируемых свойств использовавшегося в прежнем выпуске текстового поиска DB2, запустив следующий сценарий:

- Для операционных систем Linux и UNIX:  
`$DB2DIR/db2tss/bin/bkuptscfg.sh $INSTNAME`

где `DB2DIR` - положение только что установленной копии DB2 Версии 10.1, а `INSTNAME` - имя обновляемого экземпляра.

- Для операционных систем Windows:  
`<DB2PATH>\db2tss\bin\bkuptscfg.bat <INSTANCENAME> <DB2PATH>`

где `<DB2PATH>` - положение только что установленной копии DB2 Версии 10.1, а `<INSTANCENAME>` - имя обновляемого экземпляра.

Свойства с конфигурируемыми значениями, из которых создана резервная копия, перенаправляются в один файл свойств:

- В операционных системах Linux и UNIX этот файл свойств находится в каталоге `$INSTHOME/sql1lib/db2tss/config/db2tssrvupg.cfg`, где `INSTHOME` - домашний каталог экземпляра.
- В операционных системах Windows этот файл свойств находится в каталоге `<INSTPROFDIR>\<INSTANCENAME>\db2tss\config\db2tssrvupg.cfg`, где `<INSTPROFDIR>` - каталог профиля экземпляра (его можно получить, введя команду **db2set DB2INSTPROF**), а `<INSTANCENAME>` - имя экземпляра для обновления.

**Примечание:** Если экземпляр DB2 не конфигурировался с текстовым поиском DB2 в более ранней копии выпуска DB2, этот шаг можно пропустить.

10. Задайте в переменной среды `DB2INSTANCE` имя текущего обновленного экземпляра.
11. Обновите базы данных с помощью команды **db2 upgrade database**.
12. Для каждой обновленной базы данных проверьте правильность информации в таблице каталога `SYSIBMTS.TSSERVERS` текстового поиска, сравнив ее со значениями свойств, резервная копия которых была создана на шаге 7. Если значение маркера или номера порта в этой таблице каталога пусто или неверно, эту информацию сервера текстового поиска нужно обновить вручную. Подробности о выполнении обновления смотрите в теме по обновлению информации сервера текстового поиска DB2.

13. Просмотрите значения всех конфигулируемых свойств DB2 Text Search. Сравните их со значениями, для которых было выполнено резервное копирование, и убедитесь, что все значения правильны. Введите следующую команду для проверки значений конфигурации:  

```
configTool printAll -configPath <каталог-конфигурации>
```
14. Если вы отключили в текстовом поиске DB2 поддержку документов в расширенном текстовом формате, нужно установить DB2 Accessories Suite Версии 10. Дополнительную информацию смотрите в теме об установке DB2 Accessories Suite.
15. Теперь включите поддержку документов в расширенном текстовом формате. Дополнительную информацию смотрите в теме о включении поддержки документов в расширенном текстовом формате в DB2 Text Search.
16. Проверьте, успешно ли выполнено обновление, запустив службу экземпляра DB2 Text Search. Если вы отключали поддержку текстового поиска, проверьте, что теперь поддержка документов в расширенном текстовом формате включена, запустив запросы текстового поиска и сравнив результаты с полученными до обновления.

---

## Обновление установки DB2 Text Search от имени рядового пользователя (не root) (Linux и UNIX)

Чтобы получить самые последние функциональные возможности, обновите свой экземпляр DB2 Text Search. При обновлении экземпляра текстового поиска вам нужно обновить сервер DB2, экземпляр DB2 и все базы данных.

### Об этой задаче

Перечисленные ниже действия представляют процесс обновления установок от имени рядового пользователя (не root) DB2 Text Search Версии 9.5 или Версии 9.7 для платформ Linux или UNIX.

### Процедура

Для обновления установок текстового поиска DB2 без полномочий root:

1. Войдите как пользователь - владелец экземпляра или пользователь с полномочиями SYSADM. Затем остановите экземпляр DB2 и службу экземпляра DB2 Text Search.
2. Создайте резервную копию прежней копии DB2 в каталоге *<каталог\_резервного\_копирования>*.
3. Если для DB2 Text Search включена поддержка документов в расширенном текстовом формате и вы выполняете обновление из DB2 Версии 9.5 Fix Pack 3 или более новой, отключите поддержку документов в расширенном текстовом формате. Дополнительную информацию об отключении поддержки документов в расширенном текстовом формате смотрите в теме по отключению поддержки расширенных текстовых документов в DB2 Text Search.
4. Зарегистрируйтесь на сервере DB2 как пользователь без полномочий root. Проверьте тип экземпляра базы данных, чтобы убедиться, можно ли ее обновлять как установку от имени рядового пользователя (не root).
5. Установите новую копию DB2 Версии 10.1 с опцией обновления, выбрав для обновления компонент Текстовый поиск DB2. Если вы задали параметр **-f nobackup** и установка продукта базы данных DB2 завершилась неудачно, нужно

вручную установить продукт баз данных DB2, выбрав в дереве компонентов текстовый поиск DB2, а затем обновить экземпляр без полномочий root, введя следующую команду:

```
db2nrupgrade -b <каталог_резервного_копирования> -j "TEXT_SEARCH"
```

где <каталог\_резервного\_копирования> - каталог, в котором хранятся файлы конфигурации от старой версии DB2. Дополнительную информацию о команде **db2nrupgrade** смотрите в теме про эту команду.

6. Создайте резервные копии значений для всех конфигурируемых свойств использовавшегося в прежнем выпуске текстового поиска DB2 перед обновлением базы данных при помощи следующего сценария.

```
$INSTHOME/sqlllib/db2tss/bin/bkuptscfg.sh
```

Свойства с конфигурируемыми значениями, из которых создана резервная копия, перенаправляются в файл свойств *\$INSTHOME*/sqlllib/db2tss/config/db2tssrvupg.cfg.

7. Обновите существующие базы данных с помощью команды **UPGRADE DATABASE**.
8. Для каждой обновленной базы данных проверьте правильность информации в таблице каталога текстового поиска SYSIBMTS.SYSTSSERVERS, сравнив ее со значениями свойств, резервная копия которых была создана на шаге 6. Если значение маркера или номера порта в этой таблице каталога пусто или неверно, эту информацию сервера текстового поиска нужно обновить вручную. Выполнение обновления описано в теме по обновлению информации сервера текстового поиска DB2.
9. Включите возможности, доступные для пользователя root. Вы можете попросить системного администратора с правами доступа root ввести команду **db2rfe**
10. Обновите сервер DB2 Text Search для ваших экземпляров с помощью команды **configTool upgradeInstance**.

- Для операционных систем Linux и UNIX:

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

где *INSTHOME* - домашний каталог экземпляра, а *DB2DIR* - положение копии DB2 V10.1.

11. Просмотрите значения всех конфигурируемых свойств DB2 Text Search. Сравните их со значениями, для которых было выполнено резервное копирование, и убедитесь, что все значения правильны. Введите следующую команду для проверки значений конфигурации:  

```
configTool printAll -configPath каталог-конфигурации
```
12. Если вы отключили в текстовом поиске DB2 поддержку документов в расширенном текстовом формате, нужно установить DB2 Accessories Suite Версии 10. Дополнительную информацию смотрите в теме об установке DB2 Accessories Suite.
13. Теперь включите поддержку документов в расширенном текстовом формате. Дополнительную информацию смотрите в теме о включении поддержки документов в расширенном текстовом формате в текстовом поиске DB2.
14. Проверьте, успешно ли выполнено обновление, запустив службу экземпляра DB2 Text Search. Если вы отключали поддержку текстового поиска, проверьте, что теперь поддержка документов в расширенном текстовом формате включена, запустив запросы текстового поиска и сравнив результаты с полученными до обновления.

---

## Обновление многораздельного экземпляра без возможности текстового поиска DB2

Чтобы получить самые последние функциональные возможности, обновите свой экземпляр DB2 Text Search. При обновлении экземпляра текстового поиска вам нужно обновить сервер DB2, экземпляр DB2 и все базы данных.

### Об этой задаче

Начиная с DB2 V10.1, текстовый поиск поддерживает теперь индексы в среда многораздельных баз данных. Перечисленные ниже действия представляют процесс обновления многораздельного экземпляра DB2 Версии 9.5 или Версии 9.7 для установок с полномочиями root. Текстовый поиск DB2 не должен быть установлен на экземплярах.

### Процедура

1. Войдите как пользователь - владелец экземпляра или пользователь с полномочиями SYSADM.
2. Установите новую копию той версии текстового поиска DB2, до которой вы проводите обновление, и выполните пользовательскую установку. Текстовый поиск DB2 - это дополнительный компонент, который доступен только при выборе пользовательской установки.
3. Обновите ваши экземпляры, введя команду **db2iupgrade**:  
`db2iupgrade /j "text_search [[,имя-службы] | [,номер-порта]]"`
4. Обновите существующие базы данных с помощью команды **DB2 UPGRADE DATABASE**.
5. Для каждой обновленной базы данных обновите вручную информацию текстового сервера. Дополнительную информацию смотрите в теме об обновлении информации сервера текстового поиска DB2.

---

## Обновление автономного сервера текстового поиска DB2

Если автономный сервер текстового поиска DB2 уже установлен, для существующей установки нужно установить исправления, чтобы получить последние поддерживаемые возможности и функции.

### Прежде чем начать

Перед установкой исправлений прочитайте все присоединенные к выпуску примечания, чтобы определить применяемые предварительные требования и процедуры по перенастройке.

### Процедура

1. Перейдите в каталог `<ECMTS_HOME>` для текущей установки автономного сервера текстового поиска DB2.
2. Остановите сервер текстового поиска, запустив сценарий закрытия системы.
  - На платформах Linux и UNIX:  
`<ECMTS_HOME>/bin/shutdown.sh`
  - На платформах Windows:  
`<ECMTS_HOME>\bin\shutdown`

3. Скопируйте выполняемый установочный файл исправления в каталог `<ECMTS_HOME>`. Убедитесь, что в указанном каталоге содержится файл `ecmts_response.txt` для этой установки.
4. Запустите установку исправления, введя одну из следующих команд. Информация в файле ответов должна оставаться без изменений.  
`<имя_файла_установки_ecmts> -i silent -f ecmts_response.txt`

**Примечание:** Установка обновлений не изменяет маркер аутентификации.



---

## Глава 7. Конфигурирование и администрирование индексов текстового поиска

---

### Инструменты командной строки для DB2 Text Search

В DB2 Text Search включены пять инструментов командной строки, помогающие при работе с продуктом.

#### Инструмент конфигурирования

Для выполнения первоначального и последующих конфигурирования DB2 Text Search

#### Инструмент управления

Для выполнения различных административных задач, относящихся к серверу DB2 Text Search

#### Инструмент синонимов

Для добавления словарей синонимов к индексам текстового поиска и удаления словарей синонимов из индексов текстового поиска

#### Инструмент стоп-слов

Для удаления часто встречающихся терминов (так называемых стоп-слов) из запросов текстового поиска

#### Инструмент Log Formatter

Для просмотра и сохранения системных сообщений и сообщений трассировки

---

### Вызов команд текстового поиска

Вы можете передавать на выполнение команды, запустив командную оболочку **db2ts** или вызвав одну из управляющих подпрограмм SQL, представляющую собой хранимую процедуру для DB2 Text Search.

#### Об этой задаче

Для использования командной оболочки **db2ts** строку команды надо передать как параметр. Командная оболочка **db2ts** похожа на командную оболочку DB2 тем, что, если используется удаленная база данных, команда должна содержать информацию о соединении. Однако **db2ts**, в отличие от командной оболочки DB2, не поддерживает сеанс; вместо этого каждая команда представляет собой отдельную единицу и поэтому должна устанавливать соединение отдельно. Если команда запускается локально для базы данных по умолчанию, заданной при помощи переменной среды **DB2DBDFT**, задавать соединение с базой данных не требуется. Задайте переменную среды **DB2DBDFT** на уровне операционной системы. Кроме того, если она задана помощи команды **db2set**, убедитесь, что используется одно и то же значение.

Используя управляющую подпрограмму SQL, можно передавать управляющие вызовы с клиента DB2, на котором не установлен текстовый поиск DB2. Можно вызвать либо универсальную управляющую подпрограмму SQL SYSTS\_ADMIN\_CMD, указав строку команды в качестве параметра, либо конкретную управляющую подпрограмму SQL для нужной команды.

**Примечание:** Сообщения об ошибках, возвращаемые командами **db2ts**, записываются в локали клиента, но сообщения, возвращаемые управляющими

подпрограммами, записываются в локали, заданной аргументом локали сообщений (message-locale), или в локали en\_US, если локаль не задана.

Поскольку некоторые команды, такие как **START FOR TEXT** и **STOP FOR TEXT**, не связаны с конкретной базой данных, их можно вызвать только при помощи командной оболочки **db2ts**.

---

## Поддержка документов формата Rich Text Format (RTF) и собственных форматов

### Включение поддержки документов в расширенном текстовом формате для DB2 Text Search

Поддержка документов в расширенном текстовом формате может быть включена на правильно сконфигурированных серверах DB2 Text Search.

#### Прежде чем начать

Чтобы включить поддержку документов в расширенном текстовом формате для серверов DB2 Text Search, надо от имени владельца экземпляра запустить утилиту **richtextTool** с опцией **enable**.

Перед включением поддержки документов в расширенном текстовом формате каждый сервер DB2 Text Search нужно подготовить для поддержки документов в расширенном текстовом формате. Дополнительную информацию смотрите в разделе “Установка DB2 Accessories Suite для DB2 Text Search” на стр. 68

#### Ограничения

Чтобы ввести команду **richtextTool enable**, необходимо зарегистрироваться как владелец экземпляра.

#### Процедура

1. Зарегистрируйтесь как пользователь - владелец экземпляра.
2. Остановите службу экземпляра текстового поиска DB2. Для остановки службы введите команду **db2ts STOP FOR TEXT**.
3. Запустите утилиту **richtextTool** из командного окна DB2, чтобы отключить поддержку.

- Для операционных систем Linux и UNIX:  
`$INSTHOME/sqllib/db2tss/bin/richtextTool enable DB2DIR`

где *INSTHOME* - домашний каталог экземпляра, а *DB2DIR* - положение последней копии DB2.

- Для операционных систем Windows:  
`DB2PATH\db2tss\bin\richtextTool.bat enable DB2PATH`

, где *DB2PATH* - это положение, где установлена последняя копия DB2.

4. Запустите службу экземпляра текстового поиска DB2. Для запуска службы введите команду **db2ts START FOR TEXT**.

#### Результаты

Вы включили поддержку расширенного текста для сервера DB2 Text Search.

## Отключение поддержки формата RTF и собственных форматов

Поддержку формата RTF (rich text format - формат текста высокого качества) и собственных форматов можно отключить на интегрированных серверах текстового поиска DB2 в любое время.

### Прежде чем начать

Чтобы выключить поддержку документов в расширенном текстовом формате для серверов DB2 Text Search, надо от имени владельца экземпляра запустить утилиту **richtextTool** с опцией **disable**.

#### Ограничения

Чтобы запустить команду **richtextTool disable**, нужно войти в систему под именем владельца экземпляра.

### Процедура

1. Войдите как пользователь - владелец экземпляра.
2. Остановите службу экземпляра текстового поиска DB2. Для остановки службы введите команду **db2ts "STOP FOR TEXT"**. Дополнительную информацию об этой команде смотрите в разделе “Остановка службы экземпляра текстового поиска DB2” на стр. 82.
3. Запустите из командного окна DB2 утилиту **richtextTool**, чтобы отключить поддержку.
  - Для операционных систем Linux и UNIX:  

```
$INSTHOME/sqllib/db2tss/bin/richtextTool disable каталог-установки-DB2
```

где *INSTHOME* - домашний каталог экземпляра.
  - Для операционных систем Windows:  

```
DB2PATH\db2tss\bin\richtextTool.bat disable каталог-установки-DB2
```

, где *DB2PATH* - это положение, где установлена ваша копия сервера баз данных DB2.
4. Запустите службу экземпляра текстового поиска DB2. Для запуска службы введите команду **db2ts "START FOR TEXT"**. Дополнительную информацию об этой команде смотрите в разделе “Запуск службы экземпляра текстового поиска DB2”.

### Результаты

Вы выключили поддержку расширенного текста для сервера DB2 Text Search.

---

## Запуск службы экземпляра текстового поиска DB2

Чтобы можно было создавать текстовые индексы и проводить поиск с ними, необходимо сначала запустить службу экземпляра DB2 Text Search.

### Об этой задаче

Чтобы запустить службу интегрированного экземпляра DB2 Text Search, введите следующую команду:

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

Чтобы начать работу автономного сервера текстового поиска, запустите сценарий пуска из каталога `<ECMTS_HOME>`:

- В Windows:  
`<ECMTS_HOME>\bin\startup`
- В Linux и UNIX:  
`<ECMTS_HOME>/bin/startup.sh`

Проверить состояние сервера Text Search можно с помощью следующей команды:  
`db2ts "START FOR TEXT status"`

---

## Остановка службы экземпляра текстового поиска DB2

При остановке служб экземпляра текстового поиска DB2 сервер текстового поиска закрывает все активные в текущий момент программы.

### Об этой задаче

Активные программы закрываются следующим образом:

- Создание собрания для индекса текстового поиска завершается. Это может означать, что операция CREATE INDEX FOR TEXT завершится неудачно в многораздельной установке, поскольку индекс текстового поиска разделяется на несколько собраний.
- Если уже запущено отбрасывание собрания для безвозвратного удаления файлов, оно будет завершено; в противном случае выполняется откат команды.
- Обновление обрабатывает текущие документы в очереди, но не принимает другие документы; начальное обновление помечается как предпринятое и перезапускается; инкрементное обновление повторяет обработку всех записей в промежуточной таблице.

Новые команды не будут приняты, пока сервер текстового поиска не завершит обработку остановки.

### Процедура

Чтобы остановить сервер текстового поиска DB2:

- Для службы экземпляра интегрированного DB2 Text Search введите следующую команду:  
`db2ts "STOP FOR TEXT"`
- Для автономного сервера текстового поиска запустите сценарий закрытия системы из каталога `<ECMTS_HOME>`, где `<ECMTS_HOME>` - каталог установки автономного сервера текстового поиска.
  - В Windows:  
`<ECMTS_HOME>\bin\shutdown`
  - В Linux и UNIX:  
`<ECMTS_HOME>/bin/shutdown.sh`

---

## Включение базы данных для DB2 Text Search

Включение надо выполнить для каждой базы данных, содержащей столбцы текста, в котором будет нужен поиск. Включить базу данных для DB2 Text Search можно командой `db2ts ENABLE DATABASE FOR TEXT` или хранимой процедурой `SYSPROC.SYSTS_ENABLE`.

## Прежде чем начать

У ID авторизации должна быть роль SYSTS\_ADM и полномочия DBADM.

## Об этой задаче

Включив базу данных, вы сможете использовать следующие производные таблицы для получения информации об индексах текстового поиска в базе данных и о свойствах этих индексов:

### **SYSIBMTS.TSDEFAULTS**

Содержит значения по умолчанию для характеристик индекса, текста и обработки

### **SYSIBMTS.TSLOCKS**

Содержит информацию о блокировках команд, заданных на уровне базы данных и на уровне индекса

### **SYSIBMTS.TSINDEXES**

Содержит все индексы текстового поиска и значения их параметров

### **SYSIBMTS.TSCONFIGURATION**

Содержит параметры конфигурации индекса

### **SYSIBMTS.TSCOLLECTIONNAMES**

Содержит имена собраний для каждого индекса

### **SYSIBMTS.TSSERVERS**

Показывает информацию о соединении сервера Text Search

После включения базы данных для текстового поиска она останется включенной, пока вы не выключите ее явным образом.

Чтобы подготовить базу данных к использованию с DB2 Text Search, воспользуйтесь одним из следующих способов:

- Введите команду:

```
db2ts "ENABLE DATABASE FOR TEXT CONNECT TO имя-базы-данных"
```

Операция включения пытается заполнить информацией о соединении сервера текстового поиска управляющую производную таблицу SYSIBMTS.TSSERVERS. Однако эта информация может быть неполной или недостаточной. После выполнения команды, успешного или с предупреждением о неполном включении, проверьте значения в производной таблице SYSIBMTS.TSSERVERS и при необходимости измените данные.

Это действие нужно выполнить только один раз для каждой базы данных. Не нужно включать базу данных каждый раз, когда вы останавливаете и перезапускаете службы экземпляра.

Например, чтобы включить базу данных с именем SAMPLE, введите следующую команду:

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

- Вызовите одну из подпрограмм SQL администрирования, как показано ниже:
  - CALL SYSPROC.SYSTS\_ADMIN\_CMD ('ENABLE DATABASE FOR TEXT', 'en\_US', ?)
  - CALL SYSPROC.SYSTS\_ENABLE('en\_US', ?)

---

## Выключение базы данных для DB2 Text Search

Если вы больше не собираетесь выполнять текстовые поиски в базе данных, отключите текстовый поиск для нее.

### Об этой задаче

При выключении базы данных для текстового поиска из схемы SYSIBMTS отбрасываются таблицы каталога и управляющие производные таблицы.

### Процедура

Чтобы выключить базу данных для текстового поиска, воспользуйтесь одним из следующих способов:

1. Отбросьте все определенные в базе данных индексы текстового поиска при помощи команды **DROP INDEX**.
2. Чтобы выключить базу данных для текстового поиска, воспользуйтесь одним из следующих способов:
  - Введите команду **DISABLE DATABASE FOR TEXT**:  
`db2ts "DISABLE DATABASE FOR TEXT CONNECT TO имя-базы-данных"`
  - Вызовите процедуру **SYSPROC.SYSTS\_DISABLE**:  
`CALL SYSPROC.SYSTS_DISABLE('en_US', ?)`

**Примечание:** Индексы текстового поиска можно также отбросить при помощи опции **FORCE**. Однако при этом после выключения базы данных могут остаться некоторые данные, в частности, относящиеся к собранию текстового поиска. Это возможно, поскольку опция **FORCE** позволяет отбросить индексы текстового поиска даже при отключенном сервере текстового поиска DB2. Так, оставшееся собрание надо удалить явно при помощи операции **CLEANUP**.

---

## Удаление бесхозных собраний текстового поиска DB2

Бесхозные собрания можно удалить при помощи команды **db2ts CLEANUP FOR TEXT** или используя следующий процесс для идентификации и удаления бесхозных собраний с использованием инструмента администрирования.

### Об этой задаче

Для баз данных без разделов или с одним разделом индекс текстового поиска связан с одним собранием, а для многораздельных баз данных - с *n* собраниями, где *n* - это число соответствующих разделов. Хотя команды и процедуры **db2ts** работают с индексами текстового поиска, инструменты текстового поиска работают с собраниями текстового поиска. Существующее собрание текстового поиска, соответствующее индексу текстового поиска, который больше не существует, называется бесхозным собранием.

Собрание становится бесхозным при следующих сценариях:

- отбрасывание базы данных, содержащей текстовый индекс
- использование опции **FORCE** с операцией индекса **DISABLE** или **DROP**

Эти операции успешно выполняются, даже если сервер Text Search недоступен.

Собрание может стать бесхозным или перейти в недопустимое состояние также при некоторых аварийных сценариях. Например, поломка диска может привести к несогласованности метаданных текстового индекса.

Чтобы определить, существуют ли бесхозные собрания:

1. Используйте инструмент администрирования для получения сведений о всех собраниях текстового поиска. Введите команду:  
`adminTool status -configPath <абсолютный-путь-к-папке-конфигураций>`
2. Запросите управляющую производную таблицу SYSIBMTS.TSCOLLECTIONNAMES, чтобы получить все индексы текстового поиска в текущей базе данных:  
`SELECT имя-собрания FROM SYSIBMTS.TSCOLLECTIONNAMES`

Передайте этот запрос на выполнение для всех баз данных с включенной поддержкой DB2 Text Search и объедините результаты в список.

Инструмент администрирования выводит список всех собраний текстового поиска, в то время как запрос управляющей производной таблицы SYSIBMTS.TSCOLLECTIONNAMES - список индексов текстового поиска только для текущей базы данных.

3. Сравните списки, возвращенные инструментом администрирования и оператором SELECT. Все собрания текстового поиска, которые были возвращены инструментом администрирования, но не были возвращены оператором SELECT - это бесхозные собрания. Единственное исключение из этого правила - собрание по умолчанию, которое создается при запуске сервера DB2 Text Search.

Удалите бесхозные собрания текстового поиска при помощи следующей команды:

```
adminTool delete -configPath <абсолютный-путь-к-папке-конфигураций>
-collectionName имя-собрания
```

**Важное замечание:** Действие, выполняемое командой **adminTool delete**, необратимо и эквивалентно отбрасыванию индекса или переработке индекса в несогласованный.

## Пример

В настоящее время у вас есть продукт DB2 Text Search, включенный для базы данных с именем DBCP1208, запущенной в системе UNIX. Для определения, существуют ли какие-либо бесхозные собрания текстового поиска, используйте инструмент администрирования и оператор SELECT:

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

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

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

```
COLLECTIONNAME
```

```
-----  
tigertail_DBCP1208_TS542717_0000
```

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

Выбрано 13 записей.

Из сравнения двух выводов видно, что у собрания текстового поиска tigertail\_DBCP1208\_TS867530\_0000 нет соответствующего ему индекса текстового поиска. Используйте инструмент администрирования для удаления бесхозного собрания:

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

---

## Словари синонимов для DB2 Text Search

Словарь синонимов содержит слова - синонимы друг друга. Словарь синонимов можно использовать для поиска синонимов терминов вашего запроса в индексе текстового поиска, что повышает результативность запросов поиска.

Используя словарь синонимов, можно искать специфичные для вашей организации слова, например, сокращения и технические жаргонизмы.

По умолчанию словарь синонимов для поиска не используется. Чтобы включить его использование, надо непосредственно добавить этот словарь в конкретный индекс текстового поиска. До использования словаря синонимов индекс текстового поиска должен быть по крайней мере однажды изменен. После добавления словаря синонимов его можно изменять сколь угодно часто по необходимости.

Словарь синонимов состоит из групп синонимов, определяемых вами в файле XML, как это показано в следующем примере:

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

## Добавление словаря синонимов для DB2 Text Search

При помощи инструмента синонимов можно легко добавить словарь синонимов в индекс текстового поиска.

## Прежде чем начать

- Для возможности добавления словаря синонимов в индекс текстового поиска сначала нужно активировать службу экземпляра DB2 Text Search.
- Индекс текстового поиска уже должен быть обновлен хотя бы один раз.
- Должен также существовать файл XML синонимов, в котором задаются группы синонимов.

## Процедура

Чтобы добавить словарь синонимов:

1. Скопируйте указанный файл XML в произвольный каталог на сервере DB2 Text Search.
2. Определите имя собрания текстового поиска, связанного с индексом текстового поиска, куда вы хотите добавить словарь синонимов. При помощи инструмента администрирования можно получить отчет обо всех собраниях текстового поиска:  
`adminTool status -configPath абсолютный-путь-к-папке-config`

3. С помощью инструмента синонимов добавьте словарь синонимов в конкретный индекс текстового поиска. Синонимы можно добавлять в режиме **append** или **replace**, то есть добавлять их к существующим синонимам, определенным для данного индекса текстового поиска, или же заменять их.

```
synonymTool importSynonym -synonymFile абсолютный-путь-к-файлу-синонимов  
-collectionName имя-собрания -replace true или false  
-configPath абсолютный-путь-к-папке-config
```

**Примечание:** Если формат XML недопустим или файл XML пуст, будет возвращена ошибка.

## Пример

Например, чтобы добавить файл синонимов `synfile.xml` в режиме добавления (`append`), введите следующую команду:

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

## Удаление словаря синонимов для DB2 Text Search

Словарь синонимов надо удалять последовательно, удаляя его для каждого собрания по отдельности, поэтому инструмент синонимов нужно использовать для всех собраний, существующих для индекса текстового поиска.

### Об этой задаче

Чтобы удалить словарь синонимов, введите следующую команду:

```
synonymTool removeSynonym -collectionName имя-собрания  
-configPath абсолютный-путь-к-папке-config
```

Где *имя-собрания* задает собрание текстового поиска, а *абсолютный-путь-к-папке-config* - абсолютный путь к папке конфигурации текстового поиска.

---

## Создание индексов текстового поиска

Индекс текстового поиска - это компиляция значимых терминов, извлеченных из текстовых документов. Для каждого термина сохраняется информация о том, из какого документа он извлечен.

Индекс тестового поиска создается один раз для каждого столбца, содержащего текст для поиска. При создании индекса текстового поиска одновременно создаются следующие объекты:

### Промежуточная таблица

Сохраняет данные всех измененных строк в пользовательской таблице.

### Дополнительная промежуточная таблица (необязательно)

Отслеживает вставки и обновления в пользовательской таблице путем обработки целостности.

### Таблица событий

Собирает информацию о состоянии команды обновления индексов или обо всех ошибках, встреченных при ее обработке. Если ошибки происходят во время индексирования, к таблице событий добавляются *события обновления индекса*.

### Триггеры для пользовательской таблицы

Триггеры добавляют информацию к промежуточной таблице при всяком добавлении, удалении или изменении документа в столбце. Эта информация необходима для синхронизации индекса при следующем индексировании.

**Примечание:** Если вы используете команду **LOAD** для заполнения документов, триггеры не активируются и инкрементное индексирование загруженных документов не работает. Используйте вместо этого команду **IMPORT**, которая активирует триггеры. Другой вариант - можно добавить вспомогательную инфраструктуру для обработки целостности, которая будет распознавать изменения, внесенные, например, командой **LOAD INSERT**.

После создания индекса текстового поиска он еще пуст и непригоден для поиска, пока вы его не измените. При создании индекса текстового поиска вы можете указать частоту, с которой планировщик должен периодически проверять, не требуется ли обновить индекс текстового поиска, и при необходимости запускать команду обновления.

## Создание индекса текстового поиска

После включения DB2 Text Search в базе данных можно создать индексы текстового поиска для столбцов, содержащих текст, в котором нужно выполнять поиск.

### Прежде чем начать

Для создания индекса текстового поиска требуется один из следующих уровней полномочий:

- Привилегия CONTROL для выбранной таблицы
- Привилегия INDEX для таблицы индекса плюс либо полномочия IMPLICIT\_SCHEMA для базы данных, либо привилегия CREATEIN для схемы таблицы индекса
- Полномочия DBADM с DATAACCESS

Для этой таблицы должен существовать первичный ключ. Если первичный ключ не существует, его надо создать до создания индекса.

## Об этой задаче

Если вы не хотите вручную применять изменения документов из таблицы к индексу текстового поиска, можно задать параметр UPDATE FREQUENCY, чтобы запланировать автоматические изменения. Используйте параметр UPDATE MINIMUM для управления, запускать ли изменение только в том случае, если в таблице произведено некоторое минимальное число изменений. Например, чтобы задать, что MYSCHEMA.MYTEXTINDEX нужно обновлять после минимум пяти изменений, и что службы обновления должны выполнять проверку каждый понедельник и среду в полночь и в полдень, введите следующую команду:

```
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', ?)
```

Создавая индекс, вы можете задать так называемую локаль (то есть язык и территорию), используя опцию **LANGUAGE**. Чтобы выполнить автоматический просмотр документов для определения его локали, задайте для **LANGUAGE** значение AUTO. Если параметр **LANGUAGE** не задается, используется значение по умолчанию. Это значение по умолчанию будет взято при помощи DEFAULTVALUE из SYSIBMTS.TSDEFAULTS при DEFAULTNAME='LANGUAGE'. (В этом случае DEFAULTVALUE задается тогда, когда для базы данных включается текстовый поиск. Это значение вычисляется по территории базы данных, если эту территорию удалось отобразить в одну из поддерживаемых локалей документов. Если определить поддерживаемую локаль документов по территории базы данных не удалось, для DEFAULTVALUE задается значение AUTO.)

### Ограничения

- Текстовый столбец в индексе должен быть одного из следующих поддерживаемых типов:
  - CHAR
  - VARCHAR
  - LONG VARCHAR
  - CLOB
  - GRAPHIC
  - VARGRAPHIC
  - LONG VARGRAPHIC
  - DBCLOB
  - BLOB
  - XML

- Объекты, связанные с текстовым поиском, должны, помимо правил именования DB2, подчиняться следующему ограничению на используемые символы:

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

Это ограничение применяется к следующим именам:

- имена схем, содержащих индексы текстового поиска
- имена таблиц, с которыми связаны индексы текстового поиска
- имена текстовых столбцов
- имена индексов текстового поиска

## Процедура

Создайте индекс текстового поиска одним из следующих методов:

- Введите команду **CREATE INDEX**:  
`db2ts "CREATE INDEX имя-индекса FOR TEXT ON имя-таблицы (имя-столбца)"`
- Вызовите хранимую процедуру SYSPROC.SYSTS\_CREATE:  
`CALL SYSPROC.SYSTS_CREATE('схема-индекса', 'имя-индекса', 'имя-таблицы  
(имя-столбца)', 'опции', 'локаль', ?)`

**Примечание:** При использовании хранимой процедуры имя схемы и имя индекса регистрозависимы.

## Пример

Например, пусть таблица PRODUCT в базе данных SAMPLE содержит столбцы для ID продукта, имени, цены, описания и так далее. Чтобы создать индекс текстового поиска с именем MYSCHEMA.MYTEXTINDEX для столбца NAME, введите следующую команду или вызовите хранимую процедуру, как описано ниже:

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

Аналогичным образом, чтобы создать индекс текстового поиска с именем MYSCHEMA.MYXMLINDEX для столбца XML DESCRIPTION (описание), введите следующую команду:

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

или

```
CALL SYSPROC.SYSTS_CREATE('MYXMLINDEX', 'MYXMLINDEX',
  'PRODUCT (DESCRIPTION)', '', 'en_US', ?)
```

## Создание индекса текстового поиска для двоичных типов данных

При создании индекса текстового поиска можно задать кодовую страницу для двоичного столбца. Это помогает механизму текстового поиска DB2 идентифицировать кодировку символов.

### Об этой задаче

Чтобы задать при создании индекса текстового поиска кодовую страницу, введите следующую команду:

```
db2ts "CREATE INDEX имя-индекса FOR TEXT ON имя-таблицы  
CODEPAGE кодовая-страница"
```

Данные, хранящиеся в столбце с двоичным типом данных, таким как BLOB или FOR BIT DATA, не преобразуются. Это значит, что для указанных документов сохраняются их исходные кодовые страницы, что может привести к ошибкам при создании индекса текстового поиска, поскольку у вас могут быть две разные кодовые страницы. Поэтому необходимо определить, используется ли кодовая страница базы данных или кодовая страница, заданная для команды db2ts **CREATE INDEX**. Если вы не знаете, какая кодовая страница была использована для создания индекса текстового поиска, это можно выяснить, введя следующий оператор:

```
db2 "SELECT CODEPAGE FROM SYSIBM.TSINDEXES where INDSHEMA='имя-схемы'  
and INDNAME='имя-индекса'"
```

## Создание индекса текстового поиска для неподдерживаемых типов данных

Если документы хранятся в столбце неподдерживаемого типа данных, например, пользовательского типа данных, задайте функцию, которая принимает пользовательский тип на входе и преобразует его на выходе в один из поддерживаемых типов.

### Об этой задаче

Текстовый столбец в индексе должен быть одного из следующих поддерживаемых типов:

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

Чтобы преобразовать тип данных столбца в один из допустимых типов, используйте один из следующих методов:

- Введите команду **db2ts CREATE INDEX** с именем функции преобразования.  

```
db2ts "CREATE INDEX имя-индекса FOR TEXT ON  
      имя-таблицы (имя-функции(имя-текстового-столбца))"
```
- Используйте пользовательскую функцию, указываемую как *имя-функции*, которая обращается к текстовым документам в столбце с неподдерживаемым текстовым поиском типом данных, выполняет преобразование типа для значения из этого столбца и возвращает значение в одном из поддерживаемых типов данных.

### Пример

В следующем примере таблица UDTTABLE содержит столбец пользовательского типа (UDT) с именем "COMPRESSED\_TEXT", который определен как CLOB(1M). Чтобы создать индекс для этого типа данных, создайте сначала пользовательскую функцию с именем UNCOMPRESS, которая получает значения типа COMPRESSED\_TEXT. Затем создайте индекс текстового поиска так:

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

## Пример: Создание энграмм и морфологических индексов для простого текста

### Об этой задаче

Следующие инструкции позволяют сконфигурировать и синхронизировать индексы текстового поиска DB2 для индексации энграмм и морфологической индексации в базе данных SAMPLE. Выполняется поиск лингвистически значимых слов на китайском языке.

## Процедура

1. Создайте для морфологической индексации и индексации энграмм две таблицы. Эти таблицы содержат столбцы для названия, автора, аннотации, международного стандартного номера и года публикации книги.

```
db2 "CREATE TABLE morphobooks (  
  isbn VARCHAR(18) not null PRIMARY KEY,  
  bookname VARCHAR(30),  
  author VARCHAR(30),  
  story blob(1G),  
  year integer  
)"
```

```
db2 "CREATE TABLE ngrambooks (  
  isbn VARCHAR(18) not null PRIMARY KEY,  
  bookname VARCHAR(30),  
  author VARCHAR(30),  
  story blob(1G),  
  year integer  
)"
```

2. Введите команду **CREATE INDEX**, чтобы создать индекс текстового поиска на столбце STORY таблицы MORPHOBOOOKS. Имя этого индекса текстового поиска - MORPHOINDEX.

```
db2ts " CREATE INDEX db2ts.morphoindex FOR TEXT  
ON morphobooks (story) LANGUAGE zh_TW  
INDEX CONFIGURATION (CJKSEGMENTATION 'morphological')  
CONNECT TO sample";
```

3. Введите команду **CREATE INDEX**, чтобы создать индекс текстового поиска на столбце STORY таблицы NGRAMBOOKS. Имя этого индекса текстового поиска - NGRAMINDEX.

```
db2ts " CREATE INDEX db2ts.ngramindex FOR TEXT  
ON ngrambooks (story) LANGUAGE zh_TW  
INDEX CONFIGURATION (CJKSEGMENTATION 'ngram')  
CONNECT TO sample";
```

4. Загрузите данные в две таблицы.

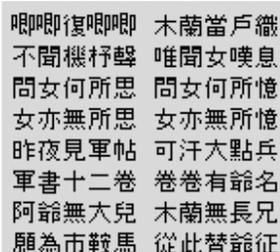
```
db2 "import from ./data/books.del of DEL lobs from ./data/  
replace into morphobooks";
```

```
db2 "import from ./data/books.del of DEL lobs from ./data/  
replace into ngrambooks";
```

Файл books.del содержит следующую запись:

```
"0-13-086755-4", "book1", "Julie", "books_zh_TW1.lob.0.449/", 2004
```

В большой объект Books\_zh\_TW1.lob включается следующее содержимое:



```
唧唧復唧唧 木蘭當戶織  
不聞機杼聲 唯聞女嘆息  
問女何所思 問女何所憶  
女亦無所思 女亦無所憶  
昨夜見軍帖 可汗大點兵  
軍書十二卷 卷卷有爺名  
阿爺無大兒 木蘭無長兄  
願為市鞍馬 從此替爺征
```

Рисунок 14. Содержимое объекта Books\_zh\_TW1.lob

5. Синхронизируйте индексы текстового поиска с данными из соответствующей таблицы, введя следующие команды:

- ```
db2ts "UPDATE INDEX db2ts.morphoindex FOR TEXT CONNECT TO sample";
```
- ```
db2ts "UPDATE INDEX db2ts.ngramindex FOR TEXT CONNECT TO sample";
```
- Здесь успешно выполняется поиск лингвистически значимых слов на китайском языке для морфологической сегментации и сегментации энграмм. Вывод показывает, что результат морфологической сегментации совпадает с

```
db2 "select bookname from morphobooks where contains (story, '軍書') = 1";
```

```
BOOKNAME
```

```
-----
```

```
book1
```

```
1 record(s) selected.
```

```
db2 "select bookname from ngrambooks where contains (story, '軍書') = 1";
```

```
BOOKNAME
```

```
-----
```

```
book1
```

```
1 record(s) selected.
```

Рисунок 15. Результаты запроса для значимых слов на китайском языке

- результатом сегментации энграмм.
- Найдите незначимые слова на китайском языке, чтобы посмотреть различие между морфологической сегментацией и сегментацией энграмм. Название книги возвращается только сегментацией энграмм.

```
db2 "select bookname from morphobooks where contains (story, '書十') = 1";
```

```
BOOKNAME
```

```
-----
```

```
0 record(s) selected.
```

```
db2 "select bookname from ngrambooks where contains (story, '書十') = 1";
```

```
BOOKNAME
```

```
-----
```

```
book1
```

```
1 record(s) selected.
```

Рисунок 16. Результаты запроса для незначимых слов на китайском языке

## Пример: Создание энграмм и морфологических индексов для расширенного текста и собственных форматов

### Об этой задаче

Следующие инструкции позволяют сконфигурировать и синхронизировать индексы текстового поиска DB2 для индексации энграмм и морфологической индексации в базе данных SAMPLE. Выполняется поиск незначимых слов на китайском языке.

### Процедура

- Создайте для морфологической индексации и индексации энграмм две таблицы. Эти таблицы содержат столбцы k и b; столбец k - это первичный ключ, а столбец b будет содержать данные расширенного текстового формата.

```
db2 "create table richtext_morpho(
k varchar(50)not null,
b blob (1G),
primary key(k)
)"
```

```
db2 "create table richtext_ngram(
k varchar(50)not null,
b blob (1G),
primary key(k)
)"
```

2. Введите команду **CREATE INDEX**, чтобы создать индекс текстового поиска на столбце b таблицы RICHTEXT\_MORPHO. Имя этого индекса текстового поиска - MORPHOINDEX.

```
db2ts " CREATE INDEX db2ts.morphoindex FOR TEXT
ON richtext_morpho (b) LANGUAGE zh_CN FORMAT INSO
INDEX CONFIGURATION (CJKSEGMENTATION 'morphological')
CONNECT TO sample";
```

3. Введите команду **CREATE INDEX**, чтобы создать индекс текстового поиска на столбце b таблицы RICHTEXT\_NGRAM. Имя этого индекса текстового поиска - NGRAMINDEX.

```
db2ts " CREATE INDEX db2ts.ngramindex FOR TEXT
ON richtext_ngram (b) LANGUAGE zh_CN FORMAT INSO
INDEX CONFIGURATION (CJKSEGMENTATION 'ngram')
CONNECT TO sample";
```

4. Загрузите данные в две таблицы.

```
db2 "import from ./data/cjk_richtext.del of DEL lobs from ./data/
replace into richtext_morpho ";
```

```
db2 "import from ./data/ cjk_richtext.del of DEL lobs from ./data/
replace into richtext_ngram ";
```

В файле cjk\_richtext.del содержатся следующие записи:

```
"rt_CJK.pdf","rt_CJK.pdf.0.864885/",
"rt_CJK.pdf.doc","rt_CJK.pdf.doc.0.90112/",
"rt_CJK.pdf.txt","rt_CJK.pdf.txt.0.37913/"
```

У файлов rt\_CJK.pdf, rt\_CJK.pdf.doc и rt\_CJK.pdf.txt одинаковое содержимое. Один сегмент содержимого на упрощенном китайском следующий:

```
"aaaaaaaa
IBM Rational License Key Center aaaaaaaaaaaaaaaaaa,aaaaaaaaa Rational
aaa aaaaaaaaaaaa IBM Rational
License Key Center aaaa,aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa,
aaaaaaaaaaaaaaaa License
Key Center aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa:
aa 1 - aaaaaaaaaaaa,aaaa"aaaa"aaaaaaaaaaaaaaaaaaaaaaaaaaaa,
aaaaaaaa License Key Center
aaaa,aaaaaaaaaaaaaaaa
aa 2 - aaa License Key Center aaaaaaaaa License Key Center
aaaaaaaa,aaaaaaaa"
```

#### 如何获得许可证密钥

IBM Rational License Key Center 是一种许可证密钥在线提供服务, 可以很方便地为您生成 Rational

密钥。但是必须成为您公司的 IBM Rational

License Key Center 帐户的成员, 才可以访问许可证密钥。为您下订单的人员被设置为帐户管理员,

并会通过电子邮件向其发送用于访问 License

Key Center 的密码。有两种方法可以使您成为公司帐户的成员:

方法 1 - 与为您下订单的人员联系, 让其使用“帐户成员”功能将您添加为公司帐户成员。一旦成功添加,

您将收到一封来自 License Key Center

的电子邮件, 其中包含了您的密码和登陆说明。

方法 2 - 除了让 License Key Center 管理员将您添加为公司 License Key Center 帐户的成员之外, 也可以自己进行添加”

Рисунок 17. Пример сегмента содержимого на упрощенном китайском

5. Синхронизируйте индексы текстового поиска с данными из соответствующей таблицы, введя следующие команды:

```
db2ts "UPDATE INDEX db2ts.morphoindex FOR TEXT
CONNECT TO sample"
```

```
db2ts "UPDATE INDEX db2ts.ngramindex FOR TEXT
CONNECT TO sample"
```

6. Здесь успешно выполняется поиск лингвистически значимых слов на китайском языке для морфологической сегментации и сегментации энграмм.

Вывод показывает, что результат морфологической сегментации совпадает с

```
db2 "select k from richtext_morpho where contains(b, '密钥')=1"
K
-----
rt_license.pdf
rt_license.pdf.doc
rt_license.pdf.txt

3 record(s) selected.

db2 "select k from richtext_ngram where contains(b, '密钥')=1"
K
-----
rt_license.pdf
rt_license.pdf.doc
rt_license.pdf.txt

3 record(s) selected.
```

Рисунок 18. Результаты запроса для лингвистически значимых слов на китайском языке

результатом сегментации энграмм.

7. Найдите незначимые слова на китайском языке, чтобы посмотреть различие между морфологической сегментацией и сегментацией энграмм.

```

db2 "select k from richtext_morpho where contains(b,'可证')=1"
К
-----

0 record(s) selected.

db2 "select k from richtext_ngram where contains(b,'可证')=1"
К
-----

rt_license.pdf
rt_license.pdf.doc
rt_license.pdf.txt

3 record(s) selected.

```

Рисунок 19. Результаты запроса для незначимых слов на китайском языке

Название книги возвращает только сегментация энграмм.

## Обслуживание индексов текстового поиска

После создания индексов текстового поиска надо выполнить несколько задач по обслуживанию. Есть несколько способов выполнять эти задачи - с помощью различных команд администрирования, хранимых процедур и инструментов администрирования.

Существуют следующие стандартные задачи по обслуживанию индекса текстового поиска:

- Запуск периодических обновлений  
Если не выполняется автоматическое обновление, необходимо обновлять индексы текстового поиска, чтобы они отражали изменения связанных с ними индексируемых текстовых столбцов.
- Мониторинг таблицы событий  
Можно использовать таблицу событий для обнаружения ошибок в документах и определения, не следует ли изменить частоту обновлений.

Менее часто требуется выполнять изменение и отбрасывание индексов поиска.

## Команды управления для DB2 Text Search

Есть несколько команд, позволяющих управлять DB2 Text Search на уровнях экземпляра, базы данных, таблицы и текстового индекса. Все эти команды можно запускать через db2ts.

Команды управления уровня экземпляра служат для запуска и остановки служб экземпляра DB2 Text Search и для очистки более не нужных индексов текстового поиска:

- db2ts START FOR TEXT**  
Запускает службы экземпляра DB2 Text Search
- db2ts STOP FOR TEXT**  
Останавливает службы экземпляра DB2 Text Search
- db2ts CLEANUP FOR TEXT**  
Очищает все ненужные собрания текстового поиска

Команды управления уровня базы данных задают или отключают базы данных для DB2 Text Search и очищают блокировки команд:

**db2ts ENABLE DATABASE FOR TEXT**

Включает для текущей базы данных создание, использование индексов текстового поиска и управление ими.

**db2ts DISABLE DATABASE FOR TEXT**

Отключает DB2 Text Search для базы данных и отбрасывает несколько таблиц и производных таблиц каталога текстового поиска.

**db2ts CLEAR COMMAND LOCKS**

Удаляет блокировки команд для всех индексов в базе данных

Команды уровня таблицы и индекса служат для создания и администрирования индексов текстового поиска по столбцам таблицы:

**db2ts CREATE INDEX**

Создает индекс текстового поиска

**db2ts DROP INDEX**

Отбрасывает индекс текстового поиска, связанный с текстовым столбцом

**db2ts ALTER INDEX**

Изменяет характеристики индекса текстового поиска

**db2ts UPDATE INDEX**

Составляет или обновляет индекс текстового поиска на основе текущего содержимого текстового столбца

**db2ts CLEAR EVENTS FOR TEXT**

Удаляет события из производной таблицы событий SYSIBMTS.TSEVENT, содержащей информацию о состоянии и ошибках индексирования

**db2ts CLEAR COMMAND LOCKS FOR INDEX**

Удаляет все блокировки команд для определенного индекса текстового поиска

**db2ts RESET PENDING FOR TABLE**

Находит все зависимые таблицы, поддерживаемые для текстового поиска, и при необходимости выполняет задание целостности

**db2ts HELP**

Выводит список опций команд **db2ts** и информацию о конкретных сообщениях об ошибках

## Хранимые процедуры DB2 Text Search

DB2 Text Search содержит несколько управляющих подпрограмм SQL для выполнения команд и для возврата сообщения о результатах выполняемых команд и кодов возврата для них.

При помощи управляющих подпрограмм SQL можно запускать следующие команды **db2ts**:

- Включить базу данных - **SYSPROC.SYSTS\_ENABLE**
- Конфигурировать базу данных - **SYSPROC.SYSTS\_CONFIGURE**
- Отключить базу данных - **SYSPROC.SYSTS\_DISABLE**
- Создать текстовый индекс - **SYSPROC.SYSTS\_CREATE**
- Обновить текстовый индекс - **SYSPROC.SYSTS\_UPDATE**
- Изменить текстовый индекс - **SYSPROC.SYSTS\_ALTER**

- Отбросить текстовый индекс - **SYSPROC.SYSTS\_DROP**
- Очистить события для текстового индекса - **SYSPROC.SYSTS\_CLEAR\_EVENTS**
- Очистить блокировки команд - **SYSPROC.SYSTS\_CLEAR\_COMMANDLOCKS**
- Сбросить состояние ожидания - **SYSPROC.SYSTS\_ADMIN\_CMD**
- Очистить неактивные индексы - **SYSPROC.SYSTS\_CLEANUP**

## Обновление индекса текстового поиска

Индекс текстового поиска можно обновлять автоматически или вручную. Автоматические обновления происходят в соответствии с тем, как вы определили частоту обновлений для индекса текстового поиска. Обновить индексы вручную можно, введя команду или вызвав хранимую процедуру.

### Прежде чем начать

Обновление индекса текстового поиска требует роли SYSTS\_MGR, а также привилегии CONTROL или полномочий DATAACCESS в таблице назначения.

### Об этой задаче

После создания и первого обновления (заполнения) индекс текстового поиска нужно поддерживать в актуальном состоянии. Например, при добавлении текстового документа в базу данных или изменении существующего документа в базе данных его нужно проиндексировать для сохранения синхронизации содержимого индекса текстового поиска с содержимым базы данных. Кроме того, при удалении текстового документа из базы данных его термины нужно также удалить и из индекса текстового поиска.

Вы должны тщательно спланировать периодическое индексирование, поскольку индексация текстовых документов - это задача, занимающая много времени и ресурсов. Затрачиваемое время зависит от многих показателей, включая размер документов, число добавленных или измененных документов с момента предыдущего обновления индекса текстового поиска и мощность процессора.

Опция status инструмента администрирования позволяет получить информацию о ходе обновления документов во время выполнения команды **db2ts UPDATE INDEX**. Если новое обновление индекса начинается, когда предыдущее обновление еще не завершено, новое обновление завершается неудачно.

- Автоматические обновления

Для выполнения автоматических обновлений индекса текстового поиска используйте одну из следующих задач, чтобы определить UPDATE FREQUENCY:

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

Минимальное значение параметра **UPDATE FREQUENCY** - пять минут. Параметр **UPDATE MINIMUM** задает минимальное число изменений текста, которые надо помещать в очередь.

Если на заданные дату и время в промежуточной таблице окажется недостаточно изменений, индекс текстового поиска обновлен не будет.

- Обновления вручную
- Иногда может потребоваться обновить индекс текстового поиска немедленно. Например, такое может случиться сразу после создания индекса текстового поиска, когда он еще пуст, или после добавления в базу данных нескольких текстовых документов, для которых нужно произвести поиск.

Для заполнения (обновления) индекса текстового поиска данными таблиц или его синхронизации с данными таблиц используется один из следующих способов:

- Введите команду **UPDATE INDEX**:  
db2ts "UPDATE INDEX *имя-индекса* FOR TEXT"
- Вызовите управляющую подпрограмму SQL SYSPROC.SYSTS\_UPDATE.

## Пример

Предположим, для таблицы PRODUCT созданы два индекса текстового поиска: MYSCHEMA.MYTEXTINDEX по столбцу NAME и MYSCHEMA.MYXMLINDEX по столбцу DESCRIPTION. В таблицу PRODUCT добавляется новая запись:

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

Чтобы сделать информацию в новой записи доступной для поиска, введите следующую команду:

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

Чтобы сделать информацию в новой записи доступной для поиска, используйте следующую хранимую процедуру:

```
db2 "call sysproc.systs_update('MYSCHEMA', 'MYXMLINDEX', '', 'en_US', ?)'
```

## Пример: Выполнение инкрементного обновления индекса текстового поиска DB2 на таблицах с разделением по диапазонам

Инкрементные обновления индексов текстового поиска DB2 на таблицах с разделением по диапазонам требуют расширенной инфраструктуры промежуточного обслуживания текста для применения изменений, обусловленных подсоединением или отсоединением разделов.

### Об этой задаче

При включенной поддержке расширенной промежуточной инфраструктуры для индексов текстового поиска операции обновления документов захватываются через триггер модификации в первичную промежуточную таблицу, а операции вставки и удаления документов захватываются во вспомогательную промежуточную таблицу через обработку целостности.

Если поддержка расширенной промежуточной инфраструктура не включена, инкрементное обновление нельзя использовать ни для обработки изменений, связанных с подсоединением или отсоединением диапазонов, ни для обработки документов, загруженных в добавленный раздел при помощи команды **LOAD** с параметром **INSERT**. Вы должны пересоздать тестовый индекс, чтобы синхронизировать его с базовой таблицей.

По умолчанию расширенная инфраструктура обслуживания текста добавляется для индексов текстового поиска на таблицах с разделением по диапазонам однако для сценариев, где индекс текстового поиска не обновляется при помощи операций инкрементного обновления, можно создать индекс текстового поиска, задав опцию **AUXLOG** со значением OFF, как показано в следующем примере:

```
db2ts create index sampleix for text on sample(comment) administration tables in
mytablespace index configuration(auxlog off) connect to mydb
```

В этом случае добавляется только первичная промежуточная таблица, а изменения документов распознаются при помощи триггеров, которые исключают изменения, связанные, например, с операциями подключения или отсоединения. При создании индексов на таблицах с разделением по диапазонам нужно задать параметр **ADMINISTRATION TABLES IN**; в противном случае генерируется сообщение об ошибке.

## Пример

### Сценарий 1: Подсоединение раздела для таблицы с расширенной промежуточной инфраструктурой текстового поиска

1. Создайте таблицу с разделением по диапазонам.
 

```
db2 "create table uc_007_customer_archive (pk integer not null
primary key, customer varchar(128) not null,
year integer not null, address blob(1M) not null) partition
by range(year)(starting(2000)ending(2001)every 1)"
```
2. Создайте индекс текстового поиска.
 

```
db2ts "create index uc_007_idx for text on
uc_007_customer_archive (address)
administration tables in mytablespace"
```
3. Посмотрите имя индекса и информацию журнала.
 

```
db2 "select indexname, stagingviewname, auxstagingname
from sysibmts.tsindexes"
```
4. Обновите индекс текстового поиска.
 

```
db2ts "update index uc_007_idx for text"
```
5. Создайте еще одну таблицу и импортируйте в нее данные.
 

```
db2 "create table uc_007_customer_2001 (pk integer not null
primary key,
customer varchar(128) not null, year integer not null,
address blob(1M) not null)"
db2 "import from uc_007_2001.del of del lobs
from ./data modified by codepage=1208
insert into uc_007_customer_2001"
```
6. Добавьте данные из новой таблицы в качестве нового раздела.
 

```
db2 "alter table uc_007_customer_archive attach
partition p2001 starting(2001) ending(2002)
exclusive from uc_007_customer_2001"
```
7. Посмотрите содержимое.
 

```
db2 "select * from sysibmts.systsauxlog_ix253720"
```

Вывод выглядит так:

PK	GLOBALTRANSID	GLOBALTRANSTIME	OPERATIONTYPE
-----	-----	-----	-----
Было выбрано 0 записей.			

8. Изменения невидимы, поэтому требуется обработка целостности. Обработка целостности переводит зависимые таблицы в режим отложенной операции.
 

```
db2 "set integrity for uc_007_customer_archive immediate checked"
```
9. Посмотрите содержимое.
 

```
db2 "select * from sysibmts.systsauxlog_ix253720"
```

Будет возвращено следующее сообщение об ошибке:

PK	GLOBALTRANSID	GLOBALTRANSTIME	OPERATIONTYPE
-----	-----	-----	-----
SQL0668N Операция для таблицы недопустима из-за кода причины "1". "SYSIBMTS"."SYSTSAUXLOG_IX253720".SQLSTATE=57016			

10. Выполните обработку целостности для промежуточных таблиц текстового поиска. Указанная команда обрабатывает для таблицы все индексы текстового поиска.

```
db2ts "reset pending for table uc_007_customer_archive for text"
db2 "select * from sysibmts.systsauxlog_ix253720"
```

Вывод выглядит так:

PK	GLOBALTRANSID	GLOBALTRANSTIME	OPERATIONTYPE
1	x'000000000002215B'	x'20081020204612500381000000'	1
2	x'000000000002215B'	x'20081020204612500602000000'	1
3	x'000000000002215B'	x'20081020204612500734000000'	1
5	x'000000000002215B'	x'20081020204612500864000000'	1

11. Используйте инкрементное обновление для обработки данных из только что подсоединенного раздела.

```
db2ts "update index uc_007_idx for text"
```

## Сценарий 2: Отсоединение раздела для таблицы с расширенной промежуточной инфраструктурой текстового поиска

1. Измените таблицу из раздела.

```
db2 alter table uc_007_customer_archive detach partition p2005
into t4p2005
```

Будет возвращено следующее сообщение:

```
SQL3601W Этот оператор вызвал автоматический перевод одной или нескольких
таблиц в состояние отложенной проверки.
SQLSTATE=01586
```

2. Введите команду **RESET PENDING**, чтобы выполнить обработку целостности для промежуточных таблиц текстового поиска.

```
db2ts "reset pending for table uc_007_customer_archive for text"
```

Используйте инкрементное обновление для обработки данных из только что отсоединенного раздела.

```
db2ts "update index uc_007_idx for text"
```

## Очистка событий индекса текстового поиска

Если сообщения в производной таблице событий индекса больше не требуются, их можно очистить (удалить).

### Прежде чем начать

Подробности, в том числе требования авторизации, смотрите в описании команды **CLEAR EVENTS FOR INDEX** или процедуры **SYSTS\_CLEAR\_EVENTS**.

### Об этой задаче

Информация о событиях индексирования, такая как время начала и окончания обновления, число проиндексированных документов или ошибки документов, произошедшие во время обновления, хранится в производной таблице событий индекса текстового поиска. Эта информация может помочь при определении причин ошибок.

### Процедура

Для очистки производной таблицы событий индекса текстового поиска используйте один из следующих способов:

- Введите команду **db2ts CLEAR EVENTS FOR INDEX** следующим образом:  
db2ts "CLEAR EVENTS FOR INDEX *имя-индекса* FOR TEXT"
- Используйте административную подпрограмму SQL SYSPROC.SYSTS\_CLEAR\_EVENTS следующим образом:  
CALL SYSPROC.SYSTS\_CLEAR\_EVENTS('схема-индекса', 'имя-индекса', 'локаль', ?)

## Изменение индекса текстового поиска

Вы можете изменить свойства обновления индекса текстового поиска.

### Прежде чем начать

Подробности, в том числе требования авторизации, смотрите в описании команды **ALTER INDEX** или процедуры **SYSTS\_ALTER**.

### Процедура

Для изменения индекса используйте один из следующих способов:

- Введите следующую команду:  
db2ts "ALTER INDEX *имя-индекса* FOR TEXT *характеристики-изменения*"

Здесь *характеристики-изменения* - характеристика, такая как частота обновления индекса текстового поиска.

- Вызовите управляющую подпрограмму SQL SYSPROC.SYSTS\_ALTER:  
CALL SYSPROC.SYSTS\_ALTER('db2ts', 'myTextIndex', '*alter-option*', 'en\_US', ?)

Здесь *опция-изменения* - характеристика, такая как частота обновления индекса текстового поиска.

### Результаты

Значения свойств текстового индекса изменятся на новые, но только если индекс не заблокирован другой операцией; в этом случае появится сообщение об ошибке, указывающее, что индекс текстового поиска в текущий момент заблокирован и никакие изменения внести невозможно.

### Пример

Вы можете использовать любой из способов, чтобы изменить и частоту обновления индекса текстового поиска, и минимальное количество изменений данных, требуемых для инициирования обновления. (Если никакие параметры не заданы, текущие установки останутся неизменными). Например, чтобы изменить частоту обновления для индекса текстового поиска MYTEXTINDEX таким образом, чтобы он обновлялся с понедельника по пятницу в 12 и 15 часов при условии, что в индексированном столбце произведено по крайней мере 100 изменений, введите следующую команду:

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

Чтобы отменить периодическое обновление MYTEXTINDEX, введите следующую команду:

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

## Просмотр состояния индексов текстового поиска

Для получения информации о текущих индексах текстового поиска в базе данных ее можно запросить из административной производной таблицы или же использовать инструмент администрирования.

### Об этой задаче

Свойства индексов текстового поиска можно просмотреть в управляющей производной таблице SYSIBMTS.TSINDEXES. Например, чтобы получить список всех индексов текстового поиска с их состояниями, введите следующий запрос:

```
db2 "select indschema, indname, indstatus from SYSIBMTS.TSINDEXES"
```

Для проверки при помощи инструмента администрирования всех собраний текстового поиска и их свойств используйте следующую команду:

```
adminTool status -configPath абсолютный-путь-к-папке-config
```

## Изменение положения собрания текстового поиска DB2

Положение собрания текстового поиска может потребоваться изменить, например, в целях администрирования и обслуживания компьютеров и дисков.

### Прежде чем начать

Изменить положение собрания текстового поиска можно только при условии, что положение собрания в таблице SYSIBMTS.TSINDEXES пусто.

### Об этой задаче

Чтобы изменить положение собрания:

### Процедура

1. Убедитесь, что положение собрания пусто.  

```
db2 "select indschema, indname, collectiondirectory, collectionnameprefix from sysibmts.tsindexes"
```
2. Если для собрания назначения отсутствует информация о каталоге, остановите сервер текстового поиска DB2.
3. Отредактируйте файл конфигурации собрания `collection.xml`. По умолчанию положение файла конфигурации собрания - `<ECMTS_HOME>\config\collections\имя_собрания\collection.xml`.
  - a. Задайте положение данных индексов.

```
<indexes>
<index>
<type>Text</type>
<path><каталог></path>
```
  - b. Задайте положение конфигурации синонимов.

```
<indexes>
<index>
<type>Synonym</type>
<path><каталог></path>
```

#### Примечание:

- Задайте эскейп-символы, как требуется в XML. Например, задайте эскейп-символ обратной дробной черты (разделитель путей по умолчанию в Windows) при помощи "\".

- Если конфигурация собрания и данные индексов находятся в каталоге собрания, можно задать путь, относительно положению файла `collection.xml`, например:

```
<indexes>
  <index>
    <type>Synonym</type>
    <path>data/text</path>
```

4. Сохраните внесенные изменения в файле `collection.xml`.
5. Повторно запустите службы DB2 Text Search.

## Резервное копирование и восстановление индексов текстового поиска

### Процедура

- Чтобы создать резервную копию базы данных с индексами текстового поиска DB2:

1. Получите текущий список положений индексов DB2 Text Search.

```
db2 "select indschema, indname, collectiondirectory, collectionnameprefix
from sysibmts.tsindexes"
```

Если не задать значение для `collectiondirectory`, для задания положений будет использоваться параметр **defaultDataDir**.

2. Убедитесь, что никакая административная команда DB2 Text Search не запущена.

3. Остановите службы DB2 Text Search.

```
db2ts stop for text
```

4. Выполните резервное копирование базы данных. Введите команду:

```
db2 backup database
имя_базы_данных
```

5. Создайте резервную копию конфигураций, каталогов и подкаталогов индекса текстового поиска.

6. Повторно запустите службы DB2 Text Search.

- Чтобы восстановить базу данных из резервной копии с индексами текстового поиска DB2:

1. Убедитесь, что не запущена ни одна административная команда текстового поиска DB2.

2. Остановите службы DB2 Text Search.

```
db2ts stop for text
```

3. Восстановите базу данных из резервной копии. Введите команду:

```
db2 restore database имя_базы_данных
```

4. Восстановите резервную копию конфигурации и каталогов индекса текстового поиска в то положение, где они находились ранее.

5. Повторно запустите службы DB2 Text Search.

```
db2ts start for text
```

## Отбрасывание индекса текстового поиска

Если вы больше не собираетесь выполнять текстовые поиски в базе данных, отключите текстовый поиск для нее.

### Прежде чем начать

Подробности, включая требования к авторизации, смотрите в описании команды `DROP INDEX` или процедуры `SYSTS_DROP`.

## Об этой задаче

Когда отбрасывается индекс текстового поиска, отбрасываются также и следующие объекты:

- Промежуточные таблицы индекса и таблицы событий индекса
- Триггеры для пользовательской таблицы

Если с индексом текстового поиска связано расписание, убедитесь, что никакие задачи не запущены. В противном случае запланированную задачу может потребоваться удалить вручную.

Всегда отбрасывайте индексы текстового поиска для таблицы до отбрасывания табличного пространства. Если вы отбросите табличное пространство, содержащее индексы текстового поиска, может образоваться так называемое *бесхозное собрание*. При создании индекса текстового поиска создается собрание (представление индекса в файловой системе) с автоматически сгенерированным именем. Если такое собрание остается в файловой системе после отбрасывания индекса, это может привести к проблемам с будущими запросами в следующих случаях:

- используется то же соединение с базой данных,
- создается таблица с таким же именем,
- для этой таблицы создается текстовый индекс с тем же, что и прежде, именем и
- повторно запускается тот же запрос, что и до этого.

В этом случае может повторно использоваться кэшированный план запроса, что приведет к неправильному результату запроса.

Команда **db2ts CLEANUP FOR TEXT** может отбросить только устаревшие собрания и соответствующие записи каталога текстовых индексов. В этом случае для удаления бесхозных собраний можно использовать инструмент конфигурирования.

Если вы планируете отбросить базу данных, включенную для текстового поиска, убедитесь, что отброшены все индексы текстового поиска, чтобы избежать бесхозных собраний.

## Процедура

Для отбрасывания индекса текстового поиска используйте один из следующих способов:

- Введите команду **DROP INDEX**:  
`db2ts "DROP INDEX имя-индекса FOR TEXT"`
- Вызовите хранимую процедуру SYSPROC.SYSTS\_DROP:  
`CALL SYSPROC.SYSTS_DROP('схема-индекса', 'имя-индекса', 'локаль', ?)`

Здесь *локаль* - это пятисимвольный код локали, например, en\_US, задающий, на каком языке будут записываться сообщения в файл журнала.

## Дальнейшие действия

**Примечание:** Если после отбрасывания индексов текстового поиска остались какие-то бесхозные собрания, их можно удалить при помощи инструмента администрирования.

Если после отбрасывания индекса текстового поиска вы планируете создать новый индекс для того же текстового столбца, необходимо сначала отсоединиться от базы данных, а затем соединиться с ней снова.

## Пример: Планирование обновления индекса текстового поиска DB2

Выполните планирование обновления индексов текстового поиска DB2 и проверьте результат выполнения.

### Прежде чем начать

Перед запуском каких-либо заданий планировщика выполните следующие задачи:

1. Задайте переменную реестра `ATS_ENABLE`.
2. Проверьте существует ли табличное пространство `SYSTOOLSPACE`.
3. Убедитесь, что база данных активирована.

Подробности о предварительных требованиях для планирования обновления индексов текстового поиска DB2 смотрите в теме по конфигурированию планировщика административных задач.

### Об этой задаче

Создайте задачу планировщика при помощи планировщика DB2 и выполняйте ее с заданной частотой.

### Процедура

1. Создайте индекс текстового поиска и задайте частоту обновления.

```
db2ts "create index simix for text on simple(comment)
update frequency (D(*) H(*) M(30))"
```

2. Соединитесь с вашей базой данных.

```
db2 connect to testdb
```

3. Найдите имя задачи планировщика.

```
db2 "select indexidentifier from sysibmts.tsindexes"
```

Для следующих действий давайте предположим, что числовой частью идентификатора индекса является 12345. Тогда имя планировщика будет `TSSCH_12345`.

4. Найдите эту задачу планировщика в управляющей производной таблице `SYSTOOLS.ADMIN_TASK_LIST`.

```
db2 "select * from systools.admin_task_list"
```

5. Проверьте состояние обновления текстового индекса.

```
db2 "select * from sysibmts.tsevent_123456"
```

6. Если никакого сообщения не выводится, но данные доступны для обновления, убедитесь, что задача планировщика запущена.

```
db2 "select * from systools.admin_task_status"
```

В противном случае при помощи имени задачи планировщика ограничьте операцию `SELECT` только данными, принадлежащими к новой задаче планировщика, для приведенного выше примера:

```
db2 "select * from systools.admin_task_status
where name = 'TSSCH_12345'"
```

---

## Глава 8. Поиск с использованием индексов текстового поиска

После заполнения индекса текстового поиска данными можно использовать этот индекс для поиска. DB2 Text Search поддерживает поиск в SQL, XQuery и SQL/XML.

Можно использовать следующие функции поиска:

- Функция SQL CONTAINS и функция XML xmlcolumn-contains - для создания запросов определенных слов или словосочетаний
- Функция SQL SCORE - для получения оценки релевантности найденного текстового документа

Поиск по индексу текстового поиска может быть как простым, например при поиске вхождений одного слова в названии, так и сложным, например при запросах с использованием логических операторов или заданием весов для термов. Помимо операторов, позволяющих уточнять сложность запросов, есть функции, дающие возможность улучшить поиск по индексам текстового поиска, например словари синонимов и лингвистическая поддержка.

---

### Функции поиска для DB2 Text Search

После обновления индекса текстового поиска можно выполнять поиск с помощью скалярной функции поиска CONTAINS или SCORE SQL или с помощью функции xmlcolumn-contains.

Поиск по индексу текстового поиска может быть как простым, например при поиске вхождений одного слова в названии, так и сложным, например при запросах с использованием логических операторов или заданием весов для термов. Помимо операторов, позволяющих уточнять сложность запросов, есть функции, дающие возможность улучшить поиск по индексам текстового поиска, например словари синонимов и лингвистическая поддержка.

Можно использовать следующие функции поиска:

- Функция SQL CONTAINS и функция XML xmlcolumn-contains - для создания запросов определенных слов или словосочетаний
- Функция SQL SCORE - для получения оценки релевантности найденного текстового документа

Скалярные функции текстового поиска CONTAINS и SCORE гладко интегрируются в SQL. Можно использовать эти функции поиска в тех же местах, где бы вы использовали стандартные выражения SQL в запросах SQL. Скалярная функция SQL SCORE возвращает показатель соответствия текстовых документов заданному условию текстового поиска. Выражение SELECT запроса SQL определяет, какая информация возвращается.

Функция CONTAINS выполняет поиск совпадений слова или словосочетания, и ее можно использовать с символами подстановки для поиска совпадений подстроки аналогично предикату SQL LIKE; кроме того, ее можно использовать для поиска точных совпадений строки аналогично оператору SQL =. Однако есть важные различия между использованием функции CONTAINS и предиката SQL LIKE или оператора =. Предикат LIKE и оператор = выполняют поиск шаблонов в документе, тогда как CONTAINS использует лингвистическую обработку, то есть выполняет

поиск различных словоформ термина поиска. Например, даже если символы подстановки не используются, поиск термина work также возвращает документы со словами working и worked. Кроме того, можно добавить словарь синонимов в индекс текстового поиска, расширяя тем самым область поиска. Например, можно сгруппировать слова laptop и ThinkPad, чтобы они возвращались как результаты поиска для notebook computers. Для документов XML синтаксис аргументов поиска XML позволяет искать текст внутри тегов и атрибутов. Кроме того, поиск XQuery учитывает регистр.

Обратите внимание, что оптимизатор DB2 оценивает, сколько текстовых документов предположительно может соответствовать предикату CONTAINS и насколько различны по затратам будут различные планы доступа. Оптимизатор выбирает наименее затратный план доступа.

Функция xmlcolumn-contains - это встроенная функция DB2, возвращающая документы XML из столбца данных XML DB2 на основе текстового поиска, выполняемого механизмом DB2 Text Search. Можно использовать xmlcolumn-contains в выражениях XQuery для извлечения документов на основе поиска определенных элементов документа. Например, если документ XML содержит описание продаваемых игрушек и цены, можно использовать xmlcolumn-contains в выражении XQuery для поиска по элементам описания и цен и возврата только тех документов, которые содержат терм outdoors, не содержат pool и стоят меньше 25,00 долларов.

Есть важные различия между использованием функции xmlcolumn-contains и функции XQuery contains. Функция XQuery contains выполняет поиск подстроки внутри строки; она ищет точное совпадение с термином поиска или словосочетанием. Функция XQuery xmlcolumn-contains похожа на функцию CONTAINS, но она работает только со столбцами XML. Кроме того, она возвращает документы XML, содержащие терм поиска или словосочетание, тогда как contains возвращает только значение (1, 0 или NULL), указывающее, найден ли терм поиска.

---

## Способы полнотекстового поиска

Для поиска в индексах текстового поиска можно использовать оператор SQL или XQuery.

### Процедура

Чтобы найти в индексе текстового поиска конкретное слово или словосочетание, используйте один из следующих способов:

- Поиск при помощи SQL.

Для поиска в индексе текстового поиска конкретного слова или словосочетания при помощи оператора SQL используется функция CONTAINS:

```
db2 "SELECT имя-столбца FROM имя-таблицы
WHERE CONTAINS (...) = 1"
```

Например, следующий запрос выполняет поиск в таблице PRODUCT наименований и цен различных снеговых лопат (snow shovel):

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

- Поиск при помощи XQuery.

Для поиска в индексе текстового поиска конкретного слова или словосочетания при помощи XQuery используется функция db2-fn:xmlcolumn-contains().

Например, следующий запрос выполняет поиск в таблице PRODUCT наименований и цен различных снеговых лопат (snow shovel):

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

**Примечание:** В зависимости от используемой вами оболочки операционной системы перед знаком доллара в информации о переменных может потребоваться другой эскейп-символ. В предыдущем примере использовалась обратная дробная черта (`\`), применяемая в качестве эскейп-символа в операционных системах UNIX.

## Базовый поиск

В поисковых запросах можно использовать логические операции и модификаторы. Чем точнее указан поисковый термин, тем более точными будут результаты.

### Пример

Пример 1: Поиски документов, содержащих термины 'wizard' и 'dragon'. Если никакая логическая операция не указана, используется операция по умолчанию AND.

```
select title from books where contains(story, 'dragon wizard')=1
```

Пример 2: Поиски документов, содержащих словосочетание 'dragon wizard'.

Документы, содержащие, например, термин 'dragons', включены не будут.

```
select title from books where contains(story, "dragon wizard")=1
```

Пример 3: Поиски документов, содержащих термин 'dragon' и необязательно термин 'wizard'. Документы, содержащие оба термина, получают более высокую оценку.

```
select title from books where contains(story, 'dragon ?wizard')=1
```

Пример 4: Поиски документов, содержащих термины 'dragon' или 'wizard', но не содержащие термин 'hobbit'.

```
select title from books where contains(story, '(dragon OR wizard) NOT hobbit')=1
```

## Поиск неполных соответствий

Поиск неполных соответствий используется для нахождения документов, содержащих слова, написание которых сходно с условием поиска.

Для запуска поиска неполных соответствий поставьте в конце фразы символ тильда (`~`) и задайте степень соответствия как дробное число в диапазоне от 0 до 1. Чем меньше это число, тем меньше могут найденные фразы соответствовать заданной. Специальные символы в запросах поиска неполных соответствий не поддерживаются.

### Пример

Шаг 1. Создаем таблицу с именем BOOKS:

```
create table books (
  isbn varchar(18) not null primary key,
  author varchar(30),
  story varchar(100),
  year integer);
```

Шаг 2. Создаем индекс текстового поиска для столбца STORY:

```
db2ts "create index bookidx for text on books(story) connect to test";
```

Шаг 3. Импортируем данные в таблицу:

```
insert into books values ('0-13-086755-1','John','The Blue Can',2001)
insert into books values ('0-13-086755-2','Mike','Cats and Dogs', 2000)
insert into books values ('0-13-086755-3','Peter','Hats on the Rack',1999)
insert into books values ('0-13-086755-4','Agatha','Cat among the Pigeons',1997)
insert into books values ('0-13-086755-5','Edgar','Cars Unlimited',2010)
insert into books values ('0-13-086755-6','Roy','Carson and Lemon',2008)
```

Шаг 4. Обновляем индекс текстового поиска:

```
db2ts "update index bookidx for text connect to test"
```

Шаг 5. Поиск неполных соответствий с функцией CONTAINS:

```
select author, year, story from books where contains(story, 'cat~0.4') = 1
```

Ниже приведен пример вывода:

```
AUTHOR YEAR STORY
```

```
-----
John 2001 The Blue Can
Mike 2000 Cats and Dogs
Agatha 1997 Cat among the Pigeons
```

3 записи выбрано.

Чтобы видеть соответствующие оценки, введите следующий запрос, который отредактирован ради уменьшения степени соответствия:

```
select author, year, story, integer(score(story, 'cat~0.3')*1000) as score
from books where contains(story, 'cat~0.3') = 1 order by score desc
```

Ниже приведен пример вывода:

```
AUTHOR YEAR STORY SCORE
```

```
-----
Agatha 1997 Cat among the Pigeons 32
John 2001 The Blue Can 17
Mike 2000 Cats and Dogs 17
Peter 1999 Hats on the Rack 1
Edgar 2010 Cars Unlimited 1
```

Было выбрано 5 записей.

## Контекстный поиск

Контекстный поиск получает документы, содержащие поисковые слова, если они расположены не далее указанного расстояния друг от друга.

Для запуска поиска с заданием расстояния между словами поставьте в конце фразы символ тильда (~) и задайте расстояние в словах как допустимое целое число. При задании расстояния учитывайте, что граница между предложениями считается за продвижение на 10 позиций. Специальные символы в запросах поиска с заданием расстояния между словами не поддерживаются.

### Пример

Шаг 1. Создаем таблицу с именем BOOKS:

```
create table books (
    isbn varchar(18) not null primary key,
    author varchar(30),
    story varchar(100),
    year integer);
```

Шаг 2. Создаем индекс текстового поиска для столбца STORY:

```
db2ts "create index bookidx for text on books(story) connect to test";
```

Шаг 3. Импортируем данные в таблицу:

```
insert into books values ('0-13-086755-1','John','Understanding Astronomy.',2001)
insert into books values ('0-13-086755-2','Mike','The cat hunts some mice.',2000)
insert into books values ('0-13-086755-3','Peter','Some men were standing beside the table.',1999)
insert into books values ('0-13-086755-4','Astrid','The outstanding adventure of Pippi Longst.',1997)
insert into books values ('0-13-086755-6','Agatha','Cat among the pigeons',2004)
insert into books values ('0-13-086755-7','John','Pigeons land in the square, and a cat plays with a ball',2001)
insert into books values ('0-13-086755-8','Sam','Pigeon on the roof',2007)
```

Шаг 4. Обновим индекс текстового поиска:

```
db2ts "update index bookidx for text connect to test"
```

Задайте поиск с заданием расстояния 4 слова для термов cat и pigeon в документе, как указано; обратите внимание на синтаксис поиска во фразе CONTAINS DB2 Text Search:

```
select author, year, substr(story,1,30) as title from books
where contains(story, "cat pigeon"~4) = 1
```

## Поиск специальных символов

Специальные символы, например, обычные символы пунктуации, индексируются как часть обновления текстового индекса, и их можно искать точно также, как любые другие термины запроса.

Чтобы найти в документе специальный символ, включите его в выражение запроса. В некоторых случаях потребуются эскейп-символы.

### Эскейп-символы для специальных символов

Некоторые специальные символы используются для различных функций в синтаксисе запроса.

Для поиска специального символа, у которого есть специальная функция в синтаксисе запроса, нужно использовать эскейп-символ, а именно добавить перед специальным символом обратную дробную черту, например:

- Для поиска строки "where?" добавьте эскейп-символ перед знаком вопроса: "where\"?"
- Для поиска строки "c:\temp" добавьте эскейп-символ перед двоеточием и перед обратной дробной чертой: "c:\\temp"

Если оставить специальные символы без эскейп-символов, могут возникнуть синтаксические ошибки.

Таблица 3. Специальные символы, для поиска которых необходимы эскейп-символы

Специальный символ	Значение без эскейп-символа
Амперсанд (&)	
Звездочка (*)	Используется как символ подстановки.

Таблица 3. Специальные символы, для поиска которых необходимы эскейп-символы (продолжение)

Специальный символ	Значение без эскейп-символа
Символ @	Если символ @ - первый символ запроса, генерируется синтаксическая ошибка. В выражениях xmlxр символ @ служит для обращения к атрибуту.
Квадратные скобки [ ]	Используются в выражениях xmlxр для поиска в содержимом элементов и атрибутов.
Фигурные скобки { }	Генерируется синтаксическая ошибка.
Обратная дробная черта (\)	
Символ ^	Служит для повышения значимости (веса) термов.
Двоеточие (:)	Служит для поиска содержимого полей.
Знак равенства (=)	Генерируется синтаксическая ошибка.
Восклицательный знак (!)	Если восклицательный знак - первый символ запроса, генерируется синтаксическая ошибка.
Дробная черта (/)	В выражениях xmlxр дробная черта служит разделителем элементов пути.
Знак больше (>), знак меньше (<)	В выражениях xmlxр служат для сравнения величин атрибутов. В других местах эти символы вызывают синтаксические ошибки.
Знак минус (-)	Когда знак минус - первый символ термина, возвращаются только документы, не содержащие данный терм.
Круглые скобки ( )	Служат для группировки.
Знак процента (%)	Указывает, что терм поиска - необязательный.
Знак плюс (+)	
Вопросительный знак (?)	Обрабатывается как символ подстановки.
Точка с запятой (;)	
Одинарная кавычка (')	В одинарные кавычки заключаются выражения xmlxр.
Тильда (~)	Задаёт поиск слов на заданном расстоянии и поиск неполных соответствий.
Вертикальная черта ( )	

Для тех специальных символов, которые не имеют специальной функции в синтаксисе запроса, эскейп-символы не обязательны. В следующей таблице показаны некоторые примеры специальных символов, для которых не требуются эскейп-символы.

Таблица 4. Примеры специальных символов, для которых не требуются эскейп-символы

Специальный символ	Примечания
Запятая (,)	
Знак доллара (\$)	
Точка (.)	В выражениях xmlxр точка служит для поиска в содержимом элементов.

Таблица 4. Примеры специальных символов, для которых не требуются эскейп-символы (продолжение)

Специальный символ	Примечания
Знак решетки (#)	
Символ подчеркивания (_)	

## Специальные символы, примыкающие к термам запроса

Когда специальный символ примыкает к слову в запросе, возвращаются документы, содержащие этот специальный символ и слово в том же порядке.

Например, поиск "30\$" находит только документы, содержащие "30\$", но не находит "\$30". Напротив, поиск "30 \$" (введено через пробел) находит все документы, содержащие "30" и "\$" в любом месте документа, включая варианты "30\$" и "\$30".

Когда специальный символ примыкает к стоп-слову в запросе, стоп-слово из запроса не удаляется. Например, при поиске "at&t" не удаляется стоп-слово "at". Напротив, при поиске "at & t" (введено через пробел) стоп-слово "at" удаляется.

Когда специальный символ разделяет два слова, последовательность маркеров ищется как последовательность. Например, поиск "jack\_jones" находит только документы, содержащие "jack\_jones", и не находит документы, содержащие "jack\_and\_jones".

Для слов, смежных со специальными символами, не выполняется лингвистическая обработка. Например, поиск "cats&dogs" на английском языке находит только документы, содержащие последовательность символов "cat&dog".

Специальные символы можно использовать в выражениях поиска с символами подстановки. Например, поиск "ja\*\_" находит документы, содержащие "jack\_jones". Но символы подстановки нельзя использовать для поиска специальных символов. Например, поиск ca\*s найдет документы, содержащие cats, categories, cast-members или cas, но не найдет документов, содержащих ca\_s.

## Индексация специальных символов

Во время лексического анализа и лингвистической обработки DB2 Text Search обнаруживает специальные символы и индексирует их как пунктуацию.

Специальные символы разделяют маркеры. Например, лексический анализ разделяет "jack\_jones" на три отдельных маркера: "jack", "\_" и "jones". Электронные адреса, URL и пути файлов разбиваются на несколько маркеров. Например:

- Jack\_jones@ibm.com при лексическом анализе трактуется как jack \_ jones @ ibm . com
- http://www.ibm.com is tokenized as http :// www . ibm . com

Специальные символы не занимают позиции как маркер в файле. Например, "jack\_jones" с подчеркиванием индексируется с той же позицией маркера, что и "jack" без подчеркивания. Кроме того, специальные символы не занимают позиции маркера, когда в него включены пробелы. Например, "jack\_jones" индексируется также, как "jack \_ jones".

Позиция маркера служит для поиска точной фразы и для поиска с зазором. Например, если документ содержит выражение jack\_jones, он будет найден при поиске точной фразы ""jack jones"".

Если последовательность специальных символов индексируется отдельно, символы ищутся без учета порядка. Например, поиск "\$#" находит также документы, содержащие последовательность "\$#".

### **Специальные символы в восточных языках**

Чтобы найти последовательность, включающую в себя специальные символы, в документе на восточных языках (китайском, японском или корейском), выражение запроса должно включать в себя специальные символы. В остальных языках специальный символ можно заменить пробелом.

Если документ на разных языках, например, документ на китайском содержит некоторые английские термины, пробел при поиске будет заменять специальные символы для терминов, не принадлежащих восточным языкам (китайскому, японскому и корейскому).

Например, если индексированный документ содержит john\_smith, можно искать john\_smith или "john smith" (точное соответствие, без подчеркивания), и оба запроса вернут документ, содержащий john\_smith.

**Примечание:** Символы '?', '\*' и '\' служат символами подстановки и эскейп-символом, но их нельзя искать как специальные символы.

## **Структурный полнотекстовый поиск в документах XML**

DB2 Text Search поддерживает использование поиска XML для поиска документов XML.

Используя подмножество языка XPath с расширениями для текстового поиска, поиск XML позволяет индексировать документы XML и проводить в них поиск. Вы можете использовать в запросах структурные элементы (имена тегов, имена атрибутов и значения атрибутов) отдельно или в сочетании со свободным текстом.

Основные возможности поиска XML:

### **Структурный поиск XML**

Используя в запросах текстового поиска синтаксис поиска XML, вы можете искать в документах XML структурные элементы (теги, имена атрибутов и значения атрибутов) и текст, находящийся в области действия этих элементов. Обратите внимание на то, что простой поиск не обрабатывает поля атрибутов в документе XML.

### **Лексический анализ запроса XML**

Текст, используемый в выражении предиката поиска XML в качестве термов запроса XML, при лексическом анализе обрабатывается так же, как текст в терминах других запросов, за тем исключением, что не поддерживаются исправления ввода, термы в поле (fielded terms) и вложенные термы поиска XML. Поддерживаются синонимы, символы подстановки, фразы и лемматизация.

### **Игнорирование пространств имен XML**

Префиксы пространства имен не сохраняются при индексации имен тегов и атрибутов XML. Документы XML, в которых объявляются и используются пространства имен, можно индексировать, и в них можно будет проводить поиск, но префиксы пространств имен игнорируются при индексации и удаляются из запросов поиска XML.

### **Числовые значения**

Поддерживаются предикаты, в которых значения атрибутов сравниваются с числами.

### Полное соответствие

Операция = (знак равенства) со строчным аргументом в предикате требует полного соответствия всех маркеров в строке всем маркерам в указанной области текста, при этом порядок существен.

Подмножество XPath, реализованное в поиске XML, отличается от стандартного XPath следующим:

- Оно не поддерживает итераций и диапазонов в выражениях для пути.
- Исключается фильтрация выражений, то есть фильтрация допускается только в выражениях предикатов, но не в выражении для пути.
- Он запрещает имена с абсолютным путем в выражениях предикатов
- Реализуется только один тег axis, распространение разрешено только в направлении вперед.

В следующей таблице приведен список некоторых допустимых запросов поиска XML.

Таблица 5. Допустимые запросы поиска XML

Запрос	Описание
/	Корневой узел; любой документ
/sentences	Все документы с тегом верхнего уровня sentences
//sentences	Все документы с тегом любого уровня sentences
sentences	Все документы с тегом любого уровня sentences
/sentence/paragraph	Все документы с тегом верхнего уровня sentences и тегом - непосредственным потомком paragraph.
/sentence/paragraph/	Все документы с тегом верхнего уровня sentences и тегом - непосредственным потомком paragraph.
/book/@author	Любой документ с тегом верхнего уровня book, у которого есть атрибут author.
/book//@author	Любой документ с тегом верхнего уровня book, у которого есть тег-потомок любого уровня с атрибутом author.
/book[@author contains("barnes") and @title contains("lemon")]	Выводится любой документ с тегом верхнего уровня book, в котором заданы атрибуты author и title со значениями, содержащими нормализованные строки
/book[@author contains("barnes") and (@title contains("lemon") or @title contains("flaubert"))]	Любой документ с тегом верхнего уровня book, в котором задан атрибут author и один из двух указанных атрибутов title.
/program[. contains("hello, world.")]	Любой документ с тегом верхнего уровня program, который содержит по крайней мере маркеры hello и world
/book[paragraph contains("flaubert")]/sentence	Любой документ с тегом верхнего уровня book, у которого, во-первых, есть тег - непосредственный потомок paragraph, содержащий "flaubert", и, во-вторых, тег-потомок sentence на любом уровне со ссылкой на тег book

Таблица 5. Допустимые запросы поиска XML (продолжение)

Запрос	Описание
/auto[@price <30000]	Любой документ с тегом верхнего уровня auto, у которого есть атрибут price с числовым значением меньше 30000.
//microbe[@size < 3.0e-06]	Любой документ, содержащий тег microbe на любом уровне с атрибутом size, значение которого меньше 3.0e-06

**Примечание:** В синтаксисе поиска XML не поддерживается:

- /\*
- //\*
- /@\*
- //@\*

*Простой* поиск не обрабатывает поля атрибутов в документе XML.

## Поиск в индексах тестового поиска при помощи SCORE

Функция SCORE позволяет определить, насколько документ соответствует аргументу поиска.

### Об этой задаче

Функция SCORE возвращает число с плавающей запятой двойной точности от 0 до 1, указывающее степень соответствия документа критериям поиска. Чем точнее документ соответствует запросу, тем больше релевантность оценки и значение результата.

Оценка вычисляется динамически на основе содержимого собрания текстовых индексов на момент запроса и имеет смысл только для однораздельного текстового индекса.

Алгоритмы вычисления оценки могут отличаться для различных форматов текстового индекса или типов запросов. Обратите внимание на то, что удаленные документы оказывают влияние на относительное значение, возвращаемое функцией SCORE, пока они не удалены из индекса текстового поиска. Однако существенные различия в оценках могли бы наблюдаться только в случае удаления из индекса больших чанков данных.

### Пример

Для поиска в базе данных SAMPLE числа сотрудников, которые написали в своих резюме, что владеют языками программирования Java или COBOL, можно ввести следующий запрос:

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

Однако лучше использовать следующий запрос с функцией CONTAINS. Оптимизатор DB2 сначала оценивает предикат CONTAINS в условии WHERE и тем самым избегает оценки функции SCORE в списке SELECT для каждой строки таблицы. Заметим, что это возможно, только если аргументы SCORE и CONTAINS в запросе идентичны.

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

---

## Синтаксис аргументов DB2 Text Search

Аргумент поиска состоит из одного или нескольких терминов и необязательных параметров поиска через пробел, заданных для поиска в текстовых документах.

Когда задан термин, механизм поиска возвращает документы, содержащие этот термин и, по умолчанию, разновидности этого термина. Например, если вы производите поиск с термином king, возвращаются документы, содержащие king и kings. При поиске с несколькими терминами механизм поиска возвращает только документы, содержащие все эти термины.

Если вы хотите искать совпадение по точной фразе, заключите эту фразу в кавычки. Используйте поиск неполных соответствий, чтобы найти документы, содержащие слова, написание которых схоже с термином поиска. Общая причина для выполнения поиска с неполным совпадением - включить в результаты документы, содержащие в результатах поиска опечатки.

Выполните контекстный поиск для получения документов, содержащих поисковые слова, если они расположены не далее указанного расстояния друг от друга.

### Напоминание:

- Поиск выполняется без учета регистра, поэтому при поиске в русском тексте точного термина "ГНИ" может возвратиться ГНИ или гни.
- Запросы текстового поиска не должны превышать пределы запросов SQL DB2.

Чем точнее указан поисковый термин, тем более точными будут результаты. Однако уточнить поиск можно и с помощью следующих опций:

### Логические операторы

Используйте оператор AND для поиска документов, содержащих все заданные термины. AND - это оператор конъюнкции по умолчанию. Если между двумя терминами не указана логическая операция, используется AND.

Используйте оператор OR для поиска документов, содержащих все заданные термины. Оператор OR связывает два или больше терминов для нахождения документов, в которых встречается любой из этих терминов.

### Модификаторы встречаемости

Используйте знак плюс (+) для указания, что термины обязательные. Модификатор знака плюс (+) отличается от оператора AND, так как этот модификатор означает, что для второго термина требуется точное совпадение. Синонимы не используются.

Используйте знак минус (-) или модификатор NOT для указания, что термины запрещены.

### Модификатор значимости

Используйте знак ^, чтобы повысить важность встречаемости конкретного термина. Знак ^ увеличивает значимость термина или предшествующего ему словосочетания, когда заданное число больше 1. Если вы хотите уменьшить ранг термина или словосочетания в возвращаемом списке, задайте число в диапазоне от 0 до 1.

Используйте модификатор значимости с функцией SCORE или с условием ORDER BY.

#### **Символы подстановки**

Используйте знак вопроса (?) для указания, что что в термине поиска может быть добавлен один символ. Используйте звездочку (\*) для указания, что в ваш термин поиска можно добавить любое количество символов.

Используйте эти символы подстановки для поиска терминов и данных с вариантами написания, а также для расширения области поиска.

**Важное замечание:** Использование символа подстановки звездочка (\*) в начале термина поиска отрицательно сказывается на производительность запросов поиска.

При поиске документов символ подстановки звездочка (\*) применяет расширение терминов. Если количество совпадающих терминов в собрании текстовых индексов превышает предел расширения, возвращается только подмножество документов, соответствующих критериям поиска.

Дополнительные подробности смотрите в теме Аргументы текстового поиска. Кроме этого, поиск с символом подстановки находит только обыкновенные символы и не находит специальные символы. Например, поиск с термином US-\* -abc находит такие строки, как US-xxx-abc, US-x-abc и US-x#-abc, но не находит US-#-abc.

#### **Знак процента (%)**

Используйте знак процента (%) для указания, что термин или словосочетание необязательны.

#### **Escape-символ - обратная дробная черта (\)**

Используйте символ обратной дробной черты (\) для включения в поиск специальных символов. Все следующие символы считаются специальными символами в запросах текстового поиска:

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

### Двойные кавычки ("")

Чтобы возвращались только точные соответствия, заключите термин или словосочетание поиска в двойные кавычки ("").

### Скобки

Используйте кавычки, чтобы термины поиска и взаимосвязь между ними трактовались как единый элемент.

Для запросов поиска XML, направляемых в анализатор XML, применяйте подмножество языка XPath с использованием неанализируемых терминов. Анализатор запросов распознает неанализируемый термин по синтаксису, который используется в запросе.

Если при поиске нужна обработка с учетом языка, для параметра аргумента поиска подразумевается локаль. Язык этого запроса соответствует локали индекса текстового поиска, используемого при работе функции поиска.

Для аргументов поиска используется следующий синтаксис:

*Аргумент поиска*

Модифицированное\_условие ((Операция) (Модифицированное\_условие))

*Операция*

AND | OR

*Модифицированное\_условие*

(Модификатор Условие (^число))

*Модификатор*

+ | - | NOT

*Условие*

Неспецифицированный термин | неанализируемый термин

- Неспецифицированный термин - это термин или словосочетание. Термин может быть словом, например, king; точным словом, например, "king"; или же словом с символом подстановки, например, king\* или king?. Аналогично, словосочетание может быть группой слов, например, cabbages and kings, точным словосочетанием, например, "The King and I", или словосочетанием, содержащим символ подстановки, например, all the king's h\* или all the kin?'s horses.
- Неанализируемые термины поиска не анализируются лингвистическим анализатором запросов; неанализируемые запросы идентифицируются по их синтаксису. Неанализируемый термин, используемый для запросов текстового поиска, - @xpath, например, @xpath: '/TagA/TagB[. contains("king")]'

## Примеры

Таблица 6. Логические операторы

Оператор	Пример	Результаты запросов
AND King AND Lear King Lear	Возвращает все документы, содержащие термины King и Lear. Если включен словарь синонимов, могут быть также возвращены такие слова, как monarch.	
OR	Hamlet OR Othello	Возвращает документы, содержащие Hamlet или Othello.

Таблица 7. Модификаторы встречаемости

Модификатор	Пример	Результат запроса
NOT -	Hamlet NOT Othello Hamlet -Othello	Возвращает документы, которые содержат Hamlet, но не содержат Othello.
+	Lear + King	Возвращает все документы, содержащие термины Lear и King. Документы, содержащие Lear и monarch, не возвращаются.

Таблица 8. Другие модификаторы

Модификатор	Пример	Результат запроса
<i>термин1</i> или <i>словосочетание1</i> <sup>1</sup> <i>число</i> <i>термин2</i> или <i>словосочетание2</i> <sup>2</sup>	Hamlet <sup>2</sup> Othello Hamlet Othello <sup>5</sup>	Возвращает документы, содержащие Hamlet и Othello, но более важным считается термин Hamlet. В обоих примерах запросов каждому найденному соответствию термина Hamlet назначается в два раза большая значимость, чем каждому найденному соответствию термина Othello.
*	king* k*ng *ing	Возвращает документы, содержащие возможные варианты поискового термина, содержащего символ подстановки. Этот пример запроса может вернуть такие результаты, как king и kingdom в первом примере, king и kissing во втором примере и king и skiing в третьем примере.
*	www.*.com	Поиск с использованием символов подстановки не возвращает термины, содержащие специальные символы. Пример запроса мог бы вернуть www.ibm.com, но www.#.com не возвращается.
?	mea? be?n ?ean	Возвращает документы, содержащие возможные варианты поискового термина, содержащего символ подстановки. В первом примере будут возвращены такие результаты, как meal и mean, во втором - bean и been, а в третьем - mean и bean.
%	King James %Edition	Возвращает документы, содержащие и king, и james, но edition - необязательный термин.

Таблица 8. Другие модификаторы (продолжение)

Модификатор	Пример	Результат запроса
"словосочетание" "точный термин" "словосочетание с символом подстановки"	"King Lear" "king" "John * Kennedy" "John ? Kennedy"	Возвращает документы, содержащие точные совпадения со словом или словосочетанием. В первом примере будет возвращено King Lear. Во втором примере возвращается слово king, но никакие другие его формы, такие как kings или kingly.  С символами подстановки можно использовать кавычки. В третьем примере будут возвращаться все комбинации слов John Kennedy с различным обозначением среднего имени или инициала. В четвертом примере возвращается John <i>инициал</i> Kennedy.
()	(Hamlet OR Othello) AND plays	Возвращает документы, содержащие следующие термины: <ul style="list-style-type: none"> <li>• Термин Hamlet или Othello</li> <li>• Термин plays</li> </ul>
\	\\(1+1)\\:2	Возвращает документы, содержащие (1+1):2. Используйте символ обратной дробной черты (\), чтобы специальные символы считались частью синтаксиса запроса.

## Синтаксис поиска для документов XML

Используя выражения поиска XML, можно при помощи механизма DB2 Text Search выполнять поиск в заданных секциях документа XML в столбце XML в DB2.

### Синтаксис

►► @xml:xp: ' | Запрос поиска XML | ' ◀◀

### Запрос поиска XML:

|—путь\_положения—| [—предикат\_поиска—]

#### @xml:xp:

Ключевое слово, с которого начинается запрос текстового поиска в документе XML.

**Примечание:** Ключевое слово @xpath объявлено устаревшим.

#### Запрос поиска XML

Запрос текстового поиска, используемый DB2 Text Search для поиска в документах XML. Этот запрос указывается в одинарных кавычках. Поисковый запрос XML - это выражение поиска XML, состоящее из пути размещения, указывающего участок документа XML, где надо выполнять поиск, и необязательный предикат, указывающий критерии поиска.

#### *путь\_положения*

Выражение поиска XML, использующее подмножество сокращенного синтаксиса XPath для указания узла или атрибута в документе XML. Дополнительную информацию смотрите в разделе "Путь положения".

#### *предикат\_поиска*

Необязательные критерии поиска, используемые DB2 Text Search при поиске в документе XML. Дополнительную информацию смотрите в разделе "Предикат поиска".

Механизм DB2 Text Search возвращает документ XML, если находит текст, указанный в параметре *предикат\_поиска*, в указанных узлах или атрибутах документа XML.

## Путь положения

При выполнении текстового поиска в документе XML DB2 Text Search использует локальные имена узлов и атрибутов, а также подмножество синтаксиса XPath для указания узлов и атрибутов в документе XML. DB2 Text Search поддерживает следующие элементы поиска XML:

- Локальные имена узлов или атрибутов
- . (точка) как узел текущего контекста
- / или // как символ-разделитель
- @ как сокращенный символ для атрибута
- Нормализация имени

Имена узлов и атрибутов XML не нормализуются при индексации для использования механизмом DB2 Text Search: они не переводятся в нижний регистр, не подвергаются лексическому анализу и вообще никак не изменяются. В именах узлов и атрибутов XML регистр учитывается, так что для успешного поиска строки, используемые в качестве этих имен в запросах, должны в точности совпадать с именами в документах.

- Обработка пространства имен

При создании индекса текстового поиска можно использовать документы XML, содержащие спецификаторы пространств имен XML, но спецификаторы пространств имен не сохраняются в индексе. Например, тег `<nsdoc:heading>` индексируется как просто `heading`, а термин запроса `@xmlxp:'/nsdoc:heading'` при анализе заменяется на `@xmlxp:'/heading'`. При анализе запросов префиксы пространств имен XML отбрасываются.

## Примеры

Ниже приведен пример допустимого запроса текстового поиска, использующего поиск XML для нахождения термина `snow shovel` (снеговая лопата) в узле описания информации о продукте:

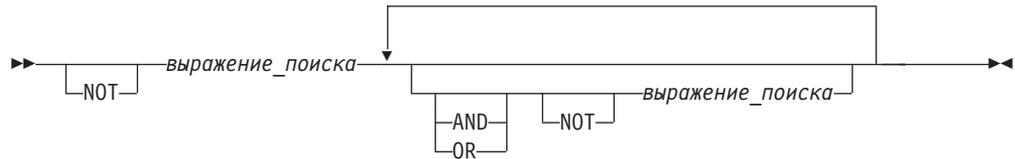
```
@xmlxp:'/info/product/description[. contains("snow shovel")]'
```

Следующий пример не является допустимым запросом текстового поиска, использующего поиск XML, так как в нем используется "...", сокращение поиска XML для `parent::node()`:

```
@xmlxp:'/info/product/description/..@ID[. contains("A2")]'
```

## Предикат поиска

### Синтаксис



#### выражение\_поиска

Запрос поиска XML DB2 Text Search. DB2 Text Search использует выражение поиска для поиска значений узлов или атрибутов в документе XML.

Для создания выражений поиска можно использовать следующие операции:

- Логические операции: AND, OR и NOT
- Операции содержания: contains и excludes
- Операции сравнения: =, >, <, >=, <= и !=

#### Примечание:

Операции сравнения применимы только к значениям атрибутов; они неприменимы к значениям узлов.

Таким образом, для `<root><aaa id="10">100</aaa><aaa id="11">101</aaa></root>` следующий запрос недопустим:

```
select id from testtable where contains(item,'@xmlxp: '/root/aaa[. > 20]''')>0
```

Пример допустимого запроса:

```
select id from testtable where contains(item,'@xmlxp: '/root/aaa/@id[. > 20]''')>0
```

Можно создавать сложные выражения поиска, сочетая операции сравнения и содержания с помощью логических операторов AND, OR и NOT. Кроме того, для группировки выражений можно использовать скобки.

Строки задаются в одинарных или двойных кавычках. Строку, содержащую кавычки, нельзя заключать в кавычки того же типа. Например, строка, заключенная в одинарные кавычки, не может содержать символ одинарной кавычки.

В предикатах поиска XML у операций сравнения есть приоритет над логическими операциями, а у всех логических операций приоритет одинаковый. Чтобы задать нужный порядок вычисления операций, используйте скобки.

Свободный текст в документах XML (расположенный не внутри тега, а между тегами) и значения атрибутов нормализуются перед индексированием. Свободный текст в запросах XML (в операторах содержания) нормализуется таким же образом, как и в других запросах (не XML).

### Пример

Ниже приведен пример использования запроса поиска XML для поиска продуктов, в описании которых содержится термин `snow shovel` (снеговая лопата) и цена которых ниже 29,99 долларов.

```
@xmlxp:'/info/product [(description contains("snow shovel"))  
and (@price < 29.99)]'
```

## Выражения сравнения

Выражения сравнения сравнивают значение атрибута с указанным значением.

## Синтаксис

►—*выражение\_пути*—*операция*—*литерал*—◄

### *выражение\_пути*

Выражение пути, использующее подмножество сокращенного синтаксиса поиска XML для указания узла или атрибута.

### *операция*

Тип выполняемого сравнения. Возможные типы операций:

- = *выражение\_пути* равно *литерал*.
- > *выражение\_пути* больше, чем *литерал*.
- < *выражение\_пути* меньше, чем *литерал*.
- >= *выражение\_пути* больше либо равно *литерал*.
- <= *выражение\_пути* меньше либо равно *литерал*.
- != *выражение\_пути* не равно *литерал*.

### *литерал*

Строка или число, сравниваемое со значением узла или атрибута *выражение\_пути*.

Строки нужно заключать в одинарные или двойные кавычки. Строку, содержащую кавычки, нельзя заключать в кавычки того же типа. Например, строка, заключенная в одинарные кавычки, не может содержать символ одинарной кавычки. Используйте символ обратной дробной черты (\) как эскейп-символ для двойных кавычек (").

Если строка содержит двойные кавычки, ее можно заключить в одинарные кавычки. В следующем примере показана строка в одинарных кавычках, содержащая двойные кавычки:

```
'he said "Hello, World"'
```

Если строка содержит одинарные кавычки, ее можно заключить в двойные кавычки, заданные при помощи эскейп-символов. В следующем примере показана строка в двойных кавычках, содержащая одинарную кавычку:

```
"the cat's toy"
```

Такие возможности DB2 Text Search, как словосочетания, символы подстановки и синонимы, не поддерживаются в запросах поиска XML.

## Пример

В следующем примере используется операция = для поиска ID продуктов, совпадающих со строкой 100-200-101:

```
@xmlxp:'/info/product/@pid[. = "100-200-101" ]'
```

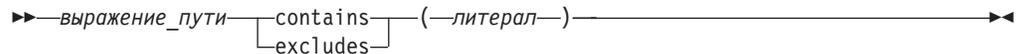
### Примечание:

Строковые аргументы поддерживаются только операциями сравнения = и !=, так что использовать <, <=, >, >= нельзя. С числовыми аргументами работают все шесть операций. Числовые аргументы можно сравнивать со значениями атрибутов, но нельзя сравнивать с содержимым тегов (узлов)

### Выражения содержания

Выражения содержания определяют, содержится ли указанное значение в значении узла или атрибута.

### Синтаксис



#### *выражение\_пути*

Выражение поиска XML, задающее узел или атрибут XML.

#### **contains**

Выражение, указывающее, что значение *выражение\_пути* содержит *литерал*.

#### **excludes**

Выражение, указывающее, что значение *выражение\_пути* не содержит *литерал*.

#### *литерал*

Строка, сравниваемая со значением узла или атрибута *выражение\_пути*.

Строки задаются в одинарных или двойных кавычках. Строка не может содержать тех кавычек, в которые она заключается: например, строка, заключенная в одинарные кавычки, не может содержать символ одинарной кавычки. Используйте символ обратной дробной черты (\) как эскейп-символ для двойных кавычек (").

Если строка содержит двойные кавычки, ее можно заключить в одинарные кавычки.

В следующем примере показана строка в одинарных кавычках, содержащая двойные кавычки:

```
'he said "Hello, World"'
```

Если строка содержит одинарные кавычки, ее можно заключить в двойные кавычки, заданные при помощи эскейп-символов. В следующем примере показана строка в двойных кавычках, содержащая одинарную кавычку:

```
"the cat's toy"
```

### Пример

В следующем примере используется сокращенный синтаксис XQuery для выражений пути, чтобы указать, что узел `description` не содержит термина `ice scraper`:

```
@xml:product/info/product/description[. excludes('ice scraper')]
```

---

## Повышение производительности для полнотекстовых запросов

Для повышения производительности при поиске рекомендуются следующие подходы:

## Процедура

- Используйте оператор EXPLAIN для проверки плана доступа оптимизатора DB2 при поиске с SQL.
- Избегайте использования функции SCORE без функции CONTAINS. Кроме того, во избежание дублирования обработки убедитесь, что строка (аргумент и все опции поиска), задаваемая вами для функции CONTAINS, в точности совпадает (включая пробелы) со строкой, используемой вами для функции SCORE.
- Убедитесь, что у компилятора DB2 есть верная статистика таблиц. Для обновления статистики используйте команду **RUNSTATS**.
- Проверьте параметр конфигурации базы данных **STMT\_CONC**. Если для этого параметра задано использование опции LITERAL, может произойти снижение производительности запросов текстового поиска.

---

## Глава 9. Встроенные функции поиска SQL и XML

Можно использовать следующие встроенные функции поиска DB2 в DB2 Text Search. Схема этих функций - SYSIBM.

### CONTAINS

Возвращает NULL или значение INTEGER, равное 0 или 1, в зависимости от того, соответствует ли входной текстовый документ условию текстового поиска

### SCORE

Возвращает NULL или значение DOUBLE от 0 до 1, указывающее, насколько текстовый документ отвечает критериям поиска.

### xmlcolumn-contains

Возвращает NULL или значение INTEGER, равное 0 или 1, в зависимости от того, соответствует ли входной текстовый документ типа данных XML условию текстового поиска

---

## функция CONTAINS

Функция CONTAINS выполняет поиск по индексу текстового поиска при помощи заданных вами в аргументе поиска критериев и возвращает значение, которое показывает, найдено ли совпадение.

### Синтаксис функции

►► CONTAINS (—имя-столбца—, —аргумент-поиска—)  
          (1)  
          ,—строчная-константа—

### Примечания:

- 1 *строчная-константа* должна соответствовать правилам для опций-аргумента-поиска.

### опции-аргументов-поиска:

(1)  
—QUERYLANGUAGE— — — локаль  
—RESULTLIMIT— — — значение  
—SYNONYM— — — OFF  
                  — — — ON

### Примечания:

- 1 Нельзя задавать несколько раз одно и то же условие.

Схема - SYSIBM.

## Параметры функций

### *имя-столбца*

Специфицированное или неспецифицированное имя столбца индекса текстового поиска, в котором будет выполняться поиск. Этот столбец должен существовать в таблице или производной таблице, указанной в операторе в условии FROM, а со столбцом таблицы или базовой таблицы этой производной таблицы должен быть связан индекс текстового поиска (SQLSTATE 38N12). Базовое выражение столбца должно представлять собой простую ссылку на столбец базовой таблицы: либо непосредственную, либо через вложенную производную таблицу.

### *аргумент-поиска*

Выражение, возвращающее строчное значение (кроме типа большого объекта), которое содержит поисковые термины; не может быть строкой из пробелов или пустой строкой (SQLSTATE 42815). Строчное значение - результат выражения не может содержать больше 4096 байт (SQLSTATE 42815). Это значение должно быть преобразовано в Unicode перед его использованием для поиска по индексу текстового поиска. Максимальное число терминов в одном запросе не может превышать 1024 (SQLSTATE 38N10).

### *строчная-константа*

Строчная константа, задающая опции аргумента поиска, которые действуют для этой функции.

В составе *опций-аргумента-поиска* можно задать следующие опции:

#### **QUERYLANGUAGE=локаль**

Задает локаль, которую механизм DB2 Text Search использует при выполнении текстового поиска в текстовом столбце DB2. Значение - любая поддерживаемая локаль. Если не задать **QUERYLANGUAGE**, значением по умолчанию будет локаль индекса текстового поиска. Если в индексе текстового поиска значение параметра **LANGUAGE** - AUTO, для **QUERYLANGUAGE** используется значение en\_US.

#### **RESULTLIMIT=значение**

Если оптимизатор выбирает план, при котором, чтобы получить оценку, механизм поиска вызывается для каждой строки набора результатов, то опция **RESULTLIMIT** на производительность не влияет. Если же механизм поиска вызывается один раз для всего набора результатов, то **RESULTLIMIT** действует как условие FETCH FIRST.

При использовании нескольких текстовых поисков, в которых **RESULTLIMIT** задается в одном и том же запросе, использовать один и тот же *аргумент-поиска*. Если вы используете разные значения *аргумента-поиска*, можно не получить ожидаемых результатов.

Для многораздельных текстовых индексов это предельное число результатов применяется к каждому разделу по отдельности.

#### **SYNONYM = OFF | ON**

Задает, следует ли использовать словарь синонимов, связанный с индексом текстового поиска. Значение по умолчанию - OFF. Чтобы использовать синонимы, добавьте словарь синонимов в индекс текстового поиска при помощи Инструментов синонимов.

**OFF** Не использовать словарь синонимов

**ON**      Использовать словарь синонимов, связанный с индексом текстового поиска.

Результат выполнения этой функции - длинное целое (large integer). Если второй аргумент допускает пустые значения, результат также допускает пустые значения; если второй аргумент пуст, результатом будет пустое значение. Если третий аргумент пуст, результат будет тем же, как и в случае, когда третий аргумент не задан. CONTAINS возвращает целое число 1, если в документе есть совпадение с критериями, заданными в аргументе поиска. Иначе возвращается 0.

Функция CONTAINS - недетерминированная.

**Примечание:** Используя маркеры параметров как аргумент поиска в функциях текстового поиска, надо выполнить дополнительные действия. У маркеров параметров нет типа во время прекомпиляции в программах JDBC и ODBC, но аргумент поиска в функциях текстового поиска должен при анализе разрешаться в строчное значение. Поскольку неизвестный тип маркера параметра не может разрешаться в строчное значение (SQLCODE -418), надо явно отобразить маркер параметра в тип данных VARCHAR.

## Примеры

- Следующий запрос служит для поиска всех сотрудников, в резюме которых упоминается COBOL. В аргументе текстового поиска регистр символов не учитывается.

```
SELECT EMPNO
FROM EMP_RESUME
WHERE RESUME_FORMAT = 'ascii'
AND CONTAINS(RESUME, 'COBOL') = 1
```

- В следующей программе С ищется точный термин ate в столбце COMMENT:

```
char search_arg[100]; /* входная переменная хоста */
...
EXEC SQL DECLARE C3 CURSOR FOR
SELECT CUSTKEY
FROM CUSTOMERS
WHERE CONTAINS(COMMENT, :search_arg)= 1
ORDER BY CUSTKEY;
strcpy(search_arg, "ate");
EXEC SQL OPEN C3;
...
```

- Следующий запрос служит для поиска любых 10 студентов, написавших интерактивные рефераты, которые содержат словосочетание горючие полезные ископаемые на испанском языке, то есть combustible fósil. Для соответствующего индекса текстового поиска был создан словарь синонимов. Поскольку требуется только 10 студентов, запрос оптимизируется при помощи опции **RESULTLIMIT**, чтобы ограничить число результатов с базового сервера текстового поиска.

```
SELECT FIRSTNAME, LASTNAME
FROM STUDENT ESSAYS
WHERE CONTAINS(TERM_PAPER, 'combustible fósil',
'QUERYLANGUAGE= es_ES RESULTLIMIT = 10 SYNONYM=ON') = 1
```

---

## функция SCORE

Функция SCORE выполняет поиск по индексу текстового поиска при помощи критериев, заданных вами в аргументе поиска, и возвращает оценку релевантности, показывающую степень соответствия документа запросу по сравнению с другими документами в столбце.

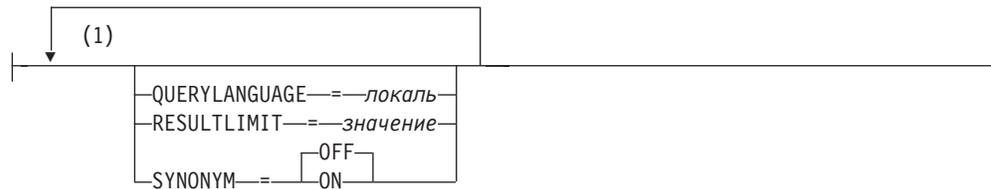
## Синтаксис функции



### Примечания:

- 1 *строочная-константа* должна соответствовать правилам для опций-аргумента-поиска.

### опции-аргументов-поиска:



### Примечания:

- 1 Нельзя задавать несколько раз одно и то же условие.

Схема - SYSIBM.

## Параметры функций

### *имя-столбца*

Специфицированное или неспецифицированное имя столбца индекса текстового поиска, в котором будет выполняться поиск. Этот столбец должен существовать в таблице или производной таблице, указанной в операторе в условии FROM, а со столбцом таблицы или базовой таблицы этой производной таблицы должен быть связан индекс текстового поиска (SQLSTATE 38N12). Базовое выражение столбца должно представлять собой простую ссылку на столбец базовой таблицы: либо непосредственную, либо через вложенную производную таблицу.

### *аргумент-поиска*

Выражение, возвращающее строочное значение (кроме типа большого объекта), которое содержит поисковые термины; не может быть строкой из пробелов или пустой строкой (SQLSTATE 42815). Строочное значение - результат выражения не может содержать больше 4096 байт (SQLSTATE 42815). Это значение должно быть преобразовано в Unicode перед его использованием для поиска по индексу текстового поиска. Максимальное число терминов в одном запросе не может превышать 1024 (SQLSTATE 38N10).

### *строочная-константа*

Строочная константа, задающая опции аргумента поиска, которые действуют для этой функции.

В составе *опций-аргумента-поиска* можно задать следующие опции:

**QUERYLANGUAGE=***локаль*

Задает локаль, которую механизм DB2 Text Search использует при

выполнении текстового поиска в текстовом столбце DB2. Значение - любая поддерживаемая локаль. Если не задать **QUERYLANGUAGE**, значением по умолчанию будет локаль индекса текстового поиска. Если в индексе текстового поиска значение параметра **LANGUAGE** - **AUTO**, для **QUERYLANGUAGE** используется значение **en\_US**.

#### **RESULTLIMIT=значение**

Если оптимизатор выбирает план, при котором, чтобы получить оценку, механизм поиска вызывается для каждой строки набора результатов, то опция **RESULTLIMIT** на производительность не влияет. Если же механизм поиска вызывается один раз для всего набора результатов, то **RESULTLIMIT** действует как условие **FETCH FIRST**.

При использовании нескольких текстовых поисков, в которых **RESULTLIMIT** задается в одном и том же запросе, использовать один и тот же *аргумент-поиска*. Если вы используете разные значения *аргумента-поиска*, можно не получить ожидаемых результатов.

Для многораздельных текстовых индексов это предельное число результатов применяется к каждому разделу по отдельности.

**Примечание:** Если важно снизить число результатов, для этого лучше уточнить термины поиска, а не использовать **RESULTLIMIT**. **RESULTLIMIT** ограничивает число результатов независимо от их оценок, так что документы высокого ранга могут не попасть в набор результатов.

#### **SYNONYM = OFF | ON**

Задаёт, следует ли использовать словарь синонимов, связанный с индексом текстового поиска. Значение по умолчанию - **OFF**. Чтобы использовать синонимы, добавьте словарь синонимов в индекс текстового поиска при помощи Инструментов синонимов.

**OFF** Не использовать словарь синонимов

**ON** Использовать словарь синонимов, связанный с индексом текстового поиска.

Результат выполнения этой функции - число с плавающей точкой двойной точности. Если второй аргумент допускает пустые значения, результат также допускает пустые значения; если второй аргумент пуст, результатом будет пустое значение. Если третий аргумент пуст, результат будет тем же, как и в случае, когда третий аргумент не задан.

Если столбец содержит совпадение с критериями, задаваемыми аргументом поиска, результатом будет значение от 0 до 1. Чем чаще встречалось соответствие, тем больше значение результата. Если столбец не содержит соответствий, результат - 0.

Функция **SCORE** - недетерминированная.

**Примечание:** Используя маркеры параметров как аргумент поиска в функциях текстового поиска, надо выполнить дополнительные действия. У маркеров параметров нет типа во время прекомпиляции в программах **JDBC** и **ODBC**, но аргумент поиска в функциях текстового поиска должен при анализе разрешаться в строчное значение. Поскольку неизвестный тип маркера параметра не может разрешаться в строчное значение (**SQLCODE -418**), надо явно отобразить маркер параметра в тип данных **VARCHAR**.

## Пример

- Следующий запрос служит для генерирования списка сотрудников в порядке соответствия их резюме запросу "программист AND (java OR cobol)" и нормализованного значения релевантности от 0 до 100:

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

## Замечания по использованию

- Значение SCORE отражает относительную релевантность документа при сравнении со значением SCORE всех документов из того же собрания текстового индекса. Для многораздельной базы данных текстовый индекс может состоять из нескольких собраний, однако оценки документов не нормализуются по всем собраниям. Сравнение или сортировка значений SCORE для различных собраний текстовых индексов, таким образом, не является осмысленной и не дает правильной меры релевантности для документов в многораздельном текстовом индексе.

---

## Функция xmlcolumn-contains

Функция db2-fn:xmlcolumn-contains возвращает последовательность документов XML из столбца данных XML на основе текстового поиска, выполненного механизмом DB2 Text Search для заданных терминов поиска.

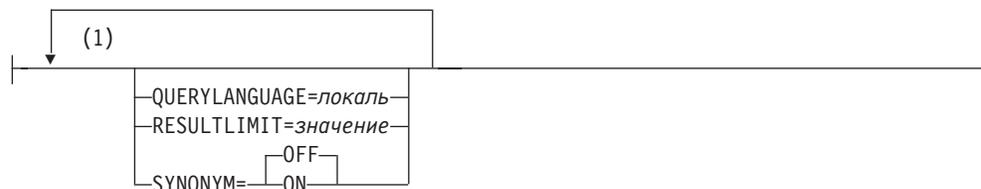
### Синтаксис

►► db2-fn:xmlcolumn-contains(строковый-литерал, аргумент-поиска (1) ,строковый-литерал-опций)►►

### Примечания:

- 1 строковый-литерал-опций должен соответствовать правилам для опций-аргумента-поиска.

### опции-аргументов-поиска:



### Примечания:

- 1 Вы можете задать каждую опцию только однажды.

### строковый-литерал

Задает имя столбца типа данных XML для поиска с помощью функции db2-fn:xmlcolumn-contains. Значение строкового литерала строковый-литерал регистрозависимо; регистр должен совпадать с регистром имени таблицы и столбца. Вы должны специфицировать имя столбца с помощью имени таблицы

или имени производной таблицы. Имя схемы SQL не обязательно. Если вы не задаете имя схемы SQL, используется значение CURRENT SCHEMA.

У столбца должен быть индекс текстового поиска.

#### *аргумент-поиска*

Выражение, возвращающее атомарное строковое значение или пустую последовательность. Строка не может полностью состоять из пробелов и не может быть пустой строкой. Строка должна быть преобразуема к типу данных VARCHAR максимальной длины 4096 байт в соответствии с правилами XMLCAST.

#### *строковый-литерал-опций*

Задаёт опции аргументов поиска, действительные для функции.

В составе *опций-аргумента-поиска* можно задать следующие опции:

#### **QUERYLANGUAGE=***локаль*

Задаёт локаль, которую механизм DB2 Text Search использует при выполнении текстового поиска в текстовом столбце DB2. Значение - любая поддерживаемая локаль. Если не задать **QUERYLANGUAGE**, значением по умолчанию будет локаль индекса текстового поиска. Если в индексе текстового поиска значение параметра **LANGUAGE** - AUTO, для **QUERYLANGUAGE** используется значение en\_US.

#### **RESULTLIMIT=***значение*

Если оптимизатор выбирает план, при котором, чтобы получить оценку, механизм поиска вызывается для каждой строки набора результатов, то опция **RESULTLIMIT** на производительность не влияет. Если же механизм поиска вызывается один раз для всего набора результатов, то **RESULTLIMIT** действует как условие FETCH FIRST.

При использовании нескольких текстовых поисков, в которых **RESULTLIMIT** задается в одном и том же запросе, использовать один и тот же *аргумент-поиска*. Если вы используете разные значения *аргумента-поиска*, можно не получить ожидаемых результатов.

Для многораздельных текстовых индексов это предельное число результатов применяется к каждому разделу по отдельности. Пример того, что может произойти при использовании нескольких текстовых поисков, а также возможное решение смотрите в последнем примере в теме “Примеры” на стр. 134.

#### **SYNONYM = OFF | ON**

Задаёт, следует ли использовать словарь синонимов, связанный с индексом текстового поиска. Значение по умолчанию - OFF. Чтобы использовать синонимы, добавьте словарь синонимов в индекс текстового поиска при помощи Инструментов синонимов.

**OFF** Не использовать словарь синонимов

**ON** Использовать словарь синонимов, связанный с индексом текстового поиска.

## **Возвращаемые значения**

Возвращаемое значение - это конкатенация непустых значений XML из столбца, который задан строковым литералом *строковой-литерал*. Непустые значения XML возвращаются в недетерминированном порядке. Такие значения XML - это документы XML, где функция SQL CONTAINS, использующая *аргумент-поиска* для

заданного строковым литералом *строковый-литерал* столбца, может вернуть 1. Если таких значений XML нет, возвращается пустая последовательность.

Если *аргумент-поиска* - это пустая последовательность, возвращается также пустая последовательность. Если *аргумент-поиска* - это пустая строка или строка, полностью состоящая из пробелов, возвращается ошибка. Если третий аргумент пуст, результат будет тем же, как и в случае, когда третий аргумент не задан.

Если у столбца, для определения которого вы использовали *строковый-литерал*, нет индекса текстового поиска, возвращается ошибка.

Функция `db2-fn:xmlcolumn-contains` связана с функцией `db2-fn:sqlquery`, и обе функции могут дать один и тот же результат. Однако аргументы двух функций отличаются по зависимости от регистра символов. Первый аргумент, *строковый-литерал*, в функции `db2-fn:xmlcolumn-contains` обрабатывается XQuery; он регистрозависим. Поскольку имена столбцов и таблиц в базе данных DB2 по умолчанию состоят из заглавных букв, обычно первый аргумент функции `db2-fn:xmlcolumn-contains` также состоит из заглавных букв. Первый аргумент функции `db2-fn:sqlquery` обрабатывается SQL, который автоматически преобразует идентификаторы в символы верхнего регистра.

Следующие вызовы функций эквивалентны и возвращают одинаковые результаты, в предположении, что таблица `PRODUCT` принадлежит схеме, в настоящее время назначенной `CURRENT SCHEMA`:

```
db2-fn:xmlcolumn-contains("PRODUCT.DESCRPTION", "snow shovel")
```

```
db2-fn:sqlquery("select description from product  
where contains(description, 'snow shovel') = 1")
```

## Примеры

В следующих примерах для выполнения поиска использован механизм DB2 Text Search. Столбцы, по которым проводится поиск - это столбцы XML, у которых есть индекс текстового поиска.

Первая функция ищет документы XML, хранимые в столбце `PRODUCT.DESCRPTION` и содержащие слова `snow` и `shovel` (снеговая лопата). Эта функция задает для максимального количества возвращаемых документов значение 2. Если текстовый поиск возвращает больше документов, можно оптимизировать поиск при помощи опции **RESULTLIMIT**, ограничивающей максимальное число возвращаемых документов.

```
db2-fn:xmlcolumn-contains('PRODUCT.DESCRPTION', 'snow shovel', 'RESULTLIMIT=2')
```

Функция возвращает документы XML, соответствующие критериям поиска. Документы могут содержать не только описание одного продукта. Например, следующий фрагмент XML состоит из описаний двух продуктов из столбца XML. Каждый документ содержит описание продукта и такую информацию, как имя продукта, цена, вес и ID продукта.

```
<product xmlns="http://posample.org" pid="100-100-01">  
  <description>  
    <name>Snow Shovel, Basic 22 inch</name>  
    <details>Basic Snow Shovel, 22 inches wide, straight handle with  
      D-Grip</details>  
    <price>9.99</price>  
    <weight>1 kg</weight>  
  </description>  
</product>  
<product xmlns="http://posample.org" pid="100-101-01">  
  <description>
```

```

<name>Snow Shovel, Deluxe 24 inch</name>
<details>A Deluxe Snow Shovel, 24 inches wide, ergonomic curved handle
  with D-Grip</details>
<price>19.99</price>
<weight>2 kg</weight>
</description>
</product>

```

Следующая функция ищет в столбце XML STUDENT\_ESSAYS.ABSTRACTS десять студенческих рефератов, содержащих фразу горячие полезные ископаемые на испанском, то есть combustible fósil. В качестве языка текстового поиска функция задает es\_ES (испанский в Испании) и использует словарь синонимов, который был создан для связанного индекса текстового поиска. Эта функция оптимизирует поиск, используя параметр **RESULTLIMIT**, который ограничивает количество результатов.

```

db2-fn:xmlcolumn-contains('STUDENT_ESSAYS.ABSTRACTS', 'combustible fosil',
'QUERYLANGUAGE=es_ES RESULTLIMIT=10 SYNONYM=ON')

```

В следующем примере функция db2-fn:xmlcolumn-contains используется для поиска документов XML, хранящихся в столбце PRODUCT.DESCRPTION и содержащих слово ergonomic. Выражение возвращает имя продукта, цена которого меньше 20.

```

xquery
declare default element namespace "http://posample.org";
db2-fn:xmlcolumn-contains(
'PRODUCT.DESCRPTION', 'ergonomic')/product/description[price < 20]/name

```

Предыдущее выражение возвращает только элементы имен из возвращаемых документов XML. Например, если термин ergonomic содержится в описании продукта Snow Shovel, Deluxe 24 inch, это выражение возвращает элемент имени аналогично следующему:

```

<name xmlns="http://posample.org" >Snow Shovel, Deluxe 24 inch<name>

```

В следующем выражении функция db2-fn:xmlcolumn-contains используется для поиска документов XML из столбца PRODUCT.DESCRPTION, содержащих слова ice и scraper. Это выражение использует ID продуктов из описаний продуктов, чтобы найти заказы на покупку в таблице PURCHASEORDER, содержащей ID продуктов. Это выражение возвращает ID покупателей из заказов на покупку, содержащих ID продуктов из соответствующих документов описаний XML.

```

xquery
declare default element namespace "http://posample.org";
for $po in db2-fn:sqlquery('
  select XMLElement(Name "po", XMLElement(Name "custid", purchaseorder.custid),
    XMLElement(Name "porder", purchaseorder.porder))
  from purchaseorder')
let $product := db2-fn:xmlcolumn-contains('PRODUCT.DESCRPTION',
'ice scraper')/product
where $product/@pid = $po/porder/PurchaseOrder/item/partid
order by $po/custid
return $po/custid

```

Это выражение возвращает элементы custid, содержащие ID покупателей. Эти элементы расположены в возрастающем порядке. Например, если запросу соответствуют три заказа на покупку, а ID покупателей равны 1001, 1002 и 1003, выражение возвращает следующие элементы:

```

<custid xmlns="http://posample.org">1001</custid>
<custid xmlns="http://posample.org">1002</custid>
<custid xmlns="http://posample.org">1003</custid>

```

Если в одном запросе выполняется несколько текстовых поисков, механизм DB2 Text Search объединяет несколько результатов текстового поиска и возвращает их.

Например, следующий оператор SELECT ищет резюме сотрудников, содержащие точные фразы `ruby on rails` и `ajax web`. Условие WHERE содержит два текстовых поиска. Каждый текстовый поиск возвращает не более десяти результатов, и каждый текстовый поиск использует различные аргументы поиска для искомым резюме сотрудников. Оператор может вернуть меньше десяти ID, даже если существует более десяти резюме, содержащих обе фразы.

```
SELECT EMPNO FROM EMP_RESUME
WHERE XMLEXISTS('db2-fn:xmlcolumn-contains(''EMP_RESUME.XML_FORMAT'',
      ''"ruby on rails"'', ''RESULTLIMIT=10''))
AND XMLEXISTS('db2-fn:xmlcolumn-contains(''EMP_RESUME.XML_FORMAT'',
      ''"ajax web"'', ''RESULTLIMIT=10''))
```

Для предыдущего оператора DB2 Text Search возвращает не более десяти строк для каждого текстового поиска. Однако если резюме в возвращаемых строках содержат только одну фразу, а не обе, никакие ID работников не возвращаются.

Один из способов изменить оператор SELECT - скомбинировать эти два текстовых поиска в условии WHERE в единый текстовый поиск. Следующий оператор использует единый текстовый поиск и возвращает ID работников, в чьих резюме содержатся обе фразы, `ruby on rails` и `ajax web`:

```
SELECT EMPNO FROM EMP_RESUME
WHERE XMLEXISTS('db2-fn:xmlcolumn-contains(''EMP_RESUME.XML_FORMAT'',
      ''"ruby on rails" AND "ajax web"'', ''RESULTLIMIT=10''))
```

Используйте одиночную обратную дробную черту как эскейп-символ для двоеточия в атрибуте XQuery:

```
xquery for $i in db2-fn:xmlcolumn-contains('DBCP1208.T_AUTO.T_XML',
'@xpath:'//en//en[. contains("purpose") and @a1 = "value for en\:attribute1"
and @slope = "9"] '' ') return $i/*//fn:string
```

---

## Глава 10. Команды управления для DB2 Text Search

Есть несколько команд, позволяющих управлять DB2 Text Search на уровнях экземпляра, базы данных, таблицы и текстового индекса. Все эти команды можно запускать через db2ts.

Команды управления уровня экземпляра служат для запуска и остановки служб экземпляра DB2 Text Search и для очистки более не нужных индексов текстового поиска:

**db2ts START FOR TEXT**

Запускает службы экземпляра DB2 Text Search

**db2ts STOP FOR TEXT**

Останавливает службы экземпляра DB2 Text Search

**db2ts CLEANUP FOR TEXT**

Очищает все ненужные собрания текстового поиска

Команды управления уровня базы данных задают или отключают базы данных для DB2 Text Search и очищают блокировки команд:

**db2ts ENABLE DATABASE FOR TEXT**

Включает для текущей базы данных создание, использование индексов текстового поиска и управление ими.

**db2ts DISABLE DATABASE FOR TEXT**

Отключает DB2 Text Search для базы данных и отбрасывает несколько таблиц и производных таблиц каталога текстового поиска.

**db2ts CLEAR COMMAND LOCKS**

Удаляет блокировки команд для всех индексов в базе данных

Команды уровня таблицы и индекса служат для создания и администрирования индексов текстового поиска по столбцам таблицы:

**db2ts CREATE INDEX**

Создает индекс текстового поиска

**db2ts DROP INDEX**

Отбрасывает индекс текстового поиска, связанный с текстовым столбцом

**db2ts ALTER INDEX**

Изменяет характеристики индекса текстового поиска

**db2ts UPDATE INDEX**

Составляет или обновляет индекс текстового поиска на основе текущего содержимого текстового столбца

**db2ts CLEAR EVENTS FOR TEXT**

Удаляет события из производной таблицы событий SYSIBMTS.TSEVENT, содержащей информацию о состоянии и ошибках индексирования

**db2ts CLEAR COMMAND LOCKS FOR INDEX**

Удаляет все блокировки команд для определенного индекса текстового поиска

**db2ts RESET PENDING FOR TABLE**

Находит все зависимые таблицы, поддерживаемые для текстового поиска, и при необходимости выполняет задание целостности



### опции:



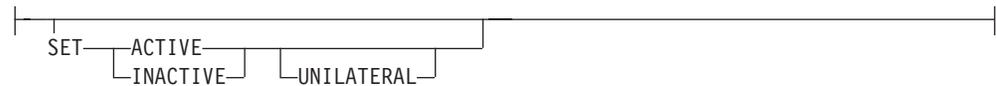
### опции конфигурации индекса:



### значение\_опции:



### опции активации:



### Возможности соединения:



## Параметры команды

### ALTER INDEX *имя\_индекса*

Схема и имя индекса, как они были заданы в команде **CREATE INDEX**. Они уникальным образом задают индекс текстового поиска в базе данных.

### UPDATE FREQUENCY

Задает частоту выполнения обновлений индекса. Индекс будет обновлен, если число изменений не меньше значения, заданного для **UPDATE MINIMUM**. Частота обновления **NONE** указывает, что индекс не будет обновляться. Это может пригодиться для текстового столбца в таблице, данные в которой никогда не изменяются. Это полезно также, если пользователь хочет обновлять индекс вручную (с помощью команды **UPDATE INDEX**). Автоматические обновления могут выполняться, только если выполнена команда **START FOR TEXT** и запущены службы экземпляра DB2 Text Search.

Значение частоты по умолчанию берется из производной таблицы SYSIBMTS.TSDEFAULTS, из строки с DEFAULTNAME='UPDATEFREQUENCY'.

### NONE

К этому текстовому индексу не будет применяться автоматическое обновление. Все последующие обновления индекса должны запускаться вручную.

**D** День недели для обновления индекса.

\* Каждый день недели.

*целое1*

Определенные дни недели, от воскресенья (0) до субботы (6)

**H** Час указанного дня для обновления индекса.

\* Каждый час дня.

*целое2*

Определенные часы дня, от полуночи до 11 вечера: от 0 до 23

**M** Минута указанного часа для обновления индекса.

*целое3*

Указывается как начало часа (0) или как кратное 5 число минут с начала часа: 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 или 55

Если вы не задали опцию **UPDATE FREQUENCY**, параметры частоты обновления не изменяются.

**UPDATE MINIMUM** *минимальное\_число\_изменений*

Задаёт минимальное число изменений в текстовых документах перед инкрементным обновлением индекса. Несколько изменений одного и того же текстового документа рассматриваются как отдельные изменения. Если вы не задали опцию **UPDATE MINIMUM**, этот параметр не изменяется.

**INDEX CONFIGURATION** (*значение\_опции*)

Начиная с версии 9.7 с пакетом Fix Pack 3 и более новыми пакетами Fix Pack, этот необязательный входной аргумент типа VARCHAR(32K) позволяет изменять параметры конфигурации текстового индекса. Поддерживаются следующие опции:

Таблица 9. Спецификации для опции - значения

Опция	Значение	Тип данных	Описание
SERIALUPDATE	<i>updatemode</i>	Целое	Задаёт, запускать ли обработку обновления для многораздельного индекса текстового поиска в параллельном или последовательном режиме. В параллельном режиме выполнение распределено по разделам базы данных и выполняется независимо на каждом узле. В последовательном режиме выполнение происходит без распределения и останавливается при возникновении ошибки. Выполнение в последовательном режиме обычно занимает больше времени, но требует значительно меньшего количества ресурсов. <ul style="list-style-type: none"><li>• 0 = параллельный режим</li><li>• 1 = последовательный режим</li></ul>

Таблица 9. Спецификации для опции - значения (продолжение)

Опция	Значение	Тип данных	Описание
UPDATEAUTOCOMMIT	<i>счетчик_принятия</i>	Целое	<p>Задаёт число изменений индекса, после которого автоматически выполняется принятие для сохранения сделанной работы, для первого обновления индекса или для инкрементных обновлений.</p> <ul style="list-style-type: none"> <li>Первоначальное обновление индекса обрабатывает пакеты документов из базовой таблицы после того, как будет активирован триггер для захвата изменений данных. Когда число измененных документов достигает COMMITCOUNT, сервер выполняет промежуточное принятие. Записи журнала, сгенерированные необработанными документами, будут удалены из таблицы кэша. Использование опции UPDATEAUTOCOMMIT для первоначального обновления текстового индекса ведет к значительному увеличению времени выполнения.</li> <li>Для инкрементных обновлений обработанные записи журнала удаляются из таблицы кэша с каждым промежуточным принятием. COMMITCOUNT подсчитывает число измененных документов, а не число записей таблицы кэша.</li> </ul>

#### ОПЦИИ АКТИВАЦИИ

Начиная с версии 9.7 с пакетом Fix Pack 3 и более новыми пакетами Fix Pack, этот входной аргумент типа integer задает состояние текстового индекса.

#### ACTIVE

Задаёт активное состояние текстового индекса

#### INACTIVE

Задаёт неактивное состояние текстового индекса

#### UNILATERAL

Задаёт одностороннее изменение, которое действует только на состояние индексов текстового поиска DB2. Если указан этот аргумент, меняется на активное или неактивное состояние только индекса текстового поиска DB2. Без аргумента UNILATERAL состояния активации индексов DB2 Text Search и DB2 Net Search Extender меняются синхронно, так что активным будет только один из этих текстовых индексов.

#### CONNECT TO *имя\_базы\_данных*

Это условие задает базу данных, с которой надо установить соединение. База данных должна находиться в локальной системе. Если условие задано, оно имеет приоритет перед переменной среды DB2DBDFT. Это условие можно опустить, если выполнены следующие условия:

- Для переменной среды DB2DBDFT задано допустимое имя базы данных.
- Пользователь, запускающий эту команду, обладает необходимой авторизацией для соединения с сервером баз данных.

#### USER *имя-пользователя* USING *пароль*

Это условие задает имя пользователя и пароль, используемые для установки соединения.

## Замечания по использованию

Все ограничения и соглашения об именовании, применимые к объектам баз данных и запросам DB2, применяются также к возможностям и запросам текстового поиска DB2. Идентификаторы, относящиеся к текстовому поиску DB2, должны отвечать соглашениям именовании DB2. Есть также некоторые дополнительные ограничения. Например, идентификаторы должны иметь вид:

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

или

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

Заметим, что нельзя одновременно вызывать для индекса текстового поиска несколько команд, если они могут конфликтовать. Если эта команда введена во время работы конфликтующей команды, произойдет ошибка и команда завершится неудачно; после этого можно попытаться запустить команду еще раз. Некоторые из конфликтующих команд:

- **ALTER INDEX**
- **CLEAR EVENTS FOR INDEX**
- **DROP INDEX**
- **UPDATE INDEX**
- **DISABLE DATABASE FOR TEXT**

Изменения в базе данных: Обновляется информация в каталоге DB2 Text Search.

Результат активации индексов зависит от исходного состояния индекса. Эти результаты описаны в следующей таблице.

Таблица 10. Изменения состояния при отсутствии недопустимых индексов:

Начальное состояние DB2 Text Search или Net Search Extender	опция активации Active	опция активации Active Unilateral	опция активации Inactive	опция активации Inactive Unilateral
Активен / Неактивен	Без изменений	Без изменений	Неактивен / Активен	Неактивен / Неактивен
Неактивен / Активен	Активен / Неактивен	Ошибка	Без изменений	Без изменений
Неактивен / Неактивен	Активен / Неактивен	Активен / Неактивен	Неактивен / Активен	Без изменений

При конфликтах активных индексов возвращаются сообщения об ошибках SQL20427N и CIE0379E.

---

## db2ts CLEANUP FOR TEXT,

Очищает собрания текстового поиска DB2 в экземпляре или в базе данных.

Если операция очистки выполнена для базы данных, недопустимые текстовые индексы и связанные с ними собрания будут отброшены. Если операция очистки выполнена для экземпляра, устаревшие собрания будут удалены. Собрание может стать устаревшим, если база данных с индексами текстового поиска отброшена раньше, чем для этой базы данных отключен текстовый поиск DB2.

**Примечание:** С индексами текстового поиска работают команды, тогда как с собраниями текстового поиска работают инструменты сервера текстового поиска. Собрание текстового поиска - это базовое представление индекса текстового поиска. Соотношение между индексом текстового поиска и связанными с ним собраниями - 1:1 при односторонней установке и 1:n при многосторонней установке, где n - число разделов данных. Запросите таблицу каталога SYSIBMTS.TSCOLLECTIONNAMES, чтобы определить собрания текстового поиска для индекса текстового поиска. Дополнительную информацию смотрите в теме об инструментах управления для текстового поиска DB2.

Для выполнения необходимо перед командой ввести **db2ts** в командной строке.

## Авторизация

Чтобы ввести команду на уровне экземпляра, надо быть владельцем процесса сервера текстового поиска. Для интегрированного сервера текстового поиска это владелец экземпляра.

Чтобы ввести команду на уровне базы данных, привилегии ID авторизации этого оператора должны включать роль SYSTS\_ADM и полномочия DBADM.

## Обязательное соединение

Эту команду надо запускать с сервера баз данных DB2.

## Синтаксис команды

Уровень экземпляра

►►—CLEANUP FOR TEXT—

Уровень базы данных

►►—CLEANUP FOR TEXT—опции-соединения—

## Параметры команды

None (отсутствует)

---

## db2ts CLEAR COMMAND LOCKS

Удаляет все блокировки команд для определенного индекса текстового поиска или для всех индексов текстового поиска в базе данных. Блокировка команды создается в начале выполнения команды текстового поиска и уничтожается при завершении этой команды. Она предотвращает конфликты между различными командами.

Использование этой команды требуется в том редком случае, когда блокировка остается неудаленной в результате неожиданного поведения системы, и ее требуется очистить явно.

Для выполнения необходимо перед командой ввести **db2ts** в командной строке.

## Авторизация

Привилегии ID авторизации этого оператора, используемого для очистки блокировок индекса, должны содержать оба указанных ниже полномочия:

- Роль SYSTS\_MGR
- Полномочия DBADM или привилегия CONTROL для базовой таблицы, на которой определен индекс

Привилегии ID авторизации этого оператора, используемого для очистки блокировок соединения с базой данных, должны включать роль SYSTS\_ADM:

## Обязательное соединение

База данных

### Синтаксис команды



### Возможности соединения:



## Параметры команды

### FOR INDEX *имя\_индекса*

Имя индекса, как оно было задано в команде **CREATE INDEX**.

### CONNECT TO *имя\_базы\_данных*

Это условие задает базу данных, с которой надо установить соединение. База данных должна находиться в локальной системе. Если условие задано, то оно имеет приоритет перед переменной среды **DB2DBDFT**. Это условие можно опустить, если выполнены следующие условия:

- Для переменной среды **DB2DBDFT** задано допустимое имя базы данных.
- Пользователь, запускающий эту команду, обладает необходимой авторизацией для соединения с сервером баз данных.

### USER *имя-пользователя* USING *пароль*

Это условие задает имя авторизации и пароль, используемые для установки соединения.

## Замечания по использованию

Вы вызываете эту процедуру, потому что процесс, которому принадлежит блокировка команды, не работает. В таком случае эта команда (владелец блокировки) не может завершиться, и индекс не может стать доступным. Вам надо предпринять действия по исправлению ситуации. Предположим, процесс, выполняющий команду **DROP INDEX**, неожиданно завис. Он удалил некоторые данные индексов, но не всю информацию каталога и собраний. Блокировка команды остается нетронутой. После очистки блокировки команды **DROP INDEX** вы можете

захотеть повторить команду **DROP INDEX**. В другом примере процесс, выполняющий команду **UPDATE INDEX**, был прерван. Он обработал не все документы, и блокировка команды все еще действует. После просмотра состояния индекса текстового поиска и освобождения блокировки команды **UPDATE INDEX** можно повторно выполнить эту команду.

Когда вводится эта команда, изменяется содержимое производной таблицы текстового поиска DB2 SYSIBMTS.TSLOCKS.

---

## db2ts CLEAR EVENTS FOR TEXT,

Эта команда удаляет события индексации из таблицы событий индекса, используемой для управления. Имя этой таблицы можно найти в столбце EVENTVIEWNAME производной таблицы SYSIBMTS.TSINDEXES.

Каждая операция обновления индекса, обработавшая хотя бы один документ, заносит информационные записи и, в некоторых случаях, записи об ошибках в таблицу событий. Для автоматических обновлений эту таблицу следует регулярно проверять. Ошибки конкретных документов надо исправлять (корректируя содержимое документов). После исправления этих ошибок события могут быть (и должны быть ради экономии места) очищены.

Для выполнения необходимо перед командой ввести **db2ts** в командной строке.

### Авторизация

Привилегии ID авторизации этого оператора должны содержать оба указанных ниже полномочия:

- Роль SYSTS\_MGR
- Полномочия DBADM с DATAACCESS или привилегия CONTROL для таблицы, на которой определен индекс

### Обязательное соединение

База данных

### Синтаксис команды

►►—CLEAR EVENTS FOR INDEX—*имя\_индекса*—FOR TEXT— | Возможности соединения | ◀◀

### Возможности соединения:

|—CONNECT TO—*имя\_базы\_данных*— |—USER— *имя\_пользователя*—USING— *пароль*— |

### Параметры команды

*имя\_индекса*

Имя индекса, как оно было задано в команде **CREATE INDEX**. Это имя должно подчиняться ограничениям именованности для индексов DB2.

**CONNECT TO** *имя\_базы\_данных*

Это условие задает базу данных, с которой надо установить соединение. База

данных должна находиться в локальной системе. Если условие задано, оно имеет приоритет перед переменной среды DB2DBDFT. Это условие можно опустить, если выполнены следующие условия:

- Для переменной среды DB2DBDFT задано допустимое имя базы данных.
- Пользователь, запускающий эту команду, обладает необходимой авторизацией для соединения с сервером баз данных.

**USER** *имя-пользователя* **USING** *пароль*

Это условие задает имя авторизации и пароль, используемые для установки соединения.

## Замечания по использованию

Все ограничения и соглашения об именовании, применимые к объектам баз данных и запросам DB2, применяются также к возможностям и запросам текстового поиска DB2. Идентификаторы, относящиеся к текстовому поиску DB2, должны отвечать соглашениям именованя DB2. Есть также некоторые дополнительные ограничения. Например, идентификаторы должны иметь вид:

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

или

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

Если планируются регулярные обновления (смотрите опции **UPDATE FREQUENCY** в командах **CREATE INDEX** или **ALTER INDEX**), таблицу событий надо проверять регулярно. Чтобы очистить таблицу событий текстового поиска DB2 для индекса текстового поиска, после проверки причины события и устранения источника ошибки используйте команду **CLEAR EVENTS FOR INDEX**.

Убедитесь, что во все строки, упомянутые в таблице событий, внесены изменения. Изменяя строки в пользовательской таблице, вы гарантируете, что следующая попытка **UPDATE INDEX** успешно переиндексирует ранее содержавшие ошибки документы.

Заметим, что нельзя одновременно вызывать для индекса текстового поиска несколько команд, если они могут конфликтовать. Если эта команда введена во время работы конфликтующей команды, произойдет ошибка и команда завершится неудачно; после этого можно попытаться запустить команду еще раз. Некоторые из конфликтующих команд:

- **CLEAR EVENTS FOR INDEX**
- **UPDATE INDEX**
- **ALTER INDEX**
- **DROP INDEX**
- **DISABLE DATABASE FOR TEXT**

Изменения в базе данных: Очищается таблица событий.

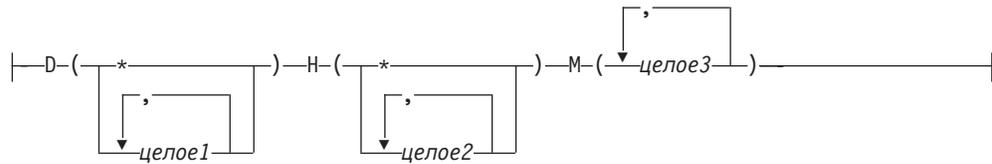
---

## db2ts CREATE INDEX,

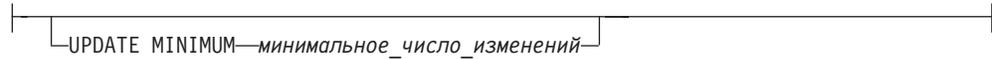
Команда **db2ts CREATE INDEX** создает индекс текстового поиска для текстового столбца. Вы можете выполнять поиск данных в этом столбце с помощью функций текстового поиска.



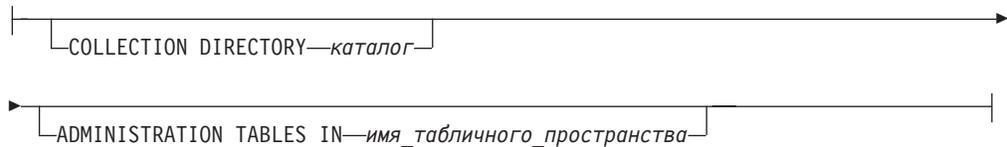
### частота обновления:



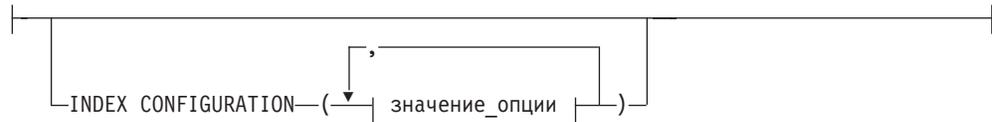
### характеристики инкрементного обновления:



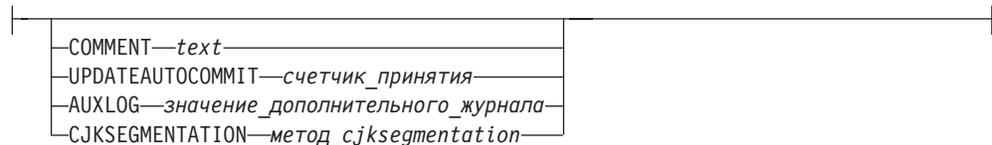
### опции хранения:



### опции конфигурации индекса:



### значение\_опции:



### Возможности соединения:



## Параметры команды

### CREATE INDEX имя\_индекса

Задаёт имя индекса (необязательно - с указанием схемы), который создается и однозначно идентифицирует индекс текстового поиска в базе данных. Это имя должно подчиняться ограничениям именования для индексов DB2.

### ON имя\_таблицы

Имя таблицы, содержащей текстовый столбец. Индексы текстового поиска нельзя создать для следующих таблиц:

- таблиц объединения

- материализованных таблиц запросов
- производных таблиц

#### *имя\_текстового\_столбца*

Имя столбца для индексирования. Столбец должен быть одного из следующих типов данных: CHAR, VARCHAR, CLOB, DBCLOB, BLOB, GRAPHIC, VARGRAPHIC или XML. Если у столбца другой тип данных, воспользуйтесь функцией преобразования (указывается как *схема\_функции.имя\_функции*), чтобы преобразовать тип данных этого столбца в один из допустимых типов. Другой вариант - указать внешнюю пользовательскую функцию, обращающуюся к текстовым документам для индексации. Для одного столбца можно создать только один индекс текстового поиска,

#### *имя\_функции(имя\_текстового\_столбца)*

Задаёт имя с указанием схемы, соответствующее соглашениям DB2 об именовании и указывающее внешнюю скалярную функцию, обращающуюся к текстовым документам в столбце, тип данных которого не поддерживается текстовым поиском. Функция выполняет преобразование типа данных значения и возвращает это значение преобразованным в один из типов данных, поддерживаемых текстовым поиском. Её задача - выполнение преобразования типа столбца. Эта функция должна принимать ровно один параметр и возвращать ровно одно значение.

#### **CODEPAGE** *кодовая\_страница*

Задаёт кодовую страницу DB2 (CODEPAGE), которую нужно использовать при индексировании текстовых документов. Значение по умолчанию задаётся значением в производной таблице SYSIBMTS.TSDEFAULTS в строке с DEFAULTNAME='CODEPAGE'; (то есть используется кодовая страница по умолчанию для базы данных). Этот аргумент применяется к двоичным типам данных, таким как тип столбца или тип возврата из функции преобразования должен быть типом BLOB или символьным типом FOR BIT DATA.

#### **LANGUAGE** *локаль*

Задаёт язык, используемый DB2 для зависящей от языка обработки документа при индексировании. Если вы не указали локаль, значение по умолчанию для параметра **LANGUAGE** определяется территорией базы данных. Чтобы выполнять автоматический просмотр документов для определения локали, задайте для переменной *locale* значение AUTO.

#### **FORMAT** *формат*

Задаёт формат текстовых документов в столбце. Поддерживаемые форматы включают в себя TEXT, XML, HTML и INSO. Эта информация нужна DB2 Text Search при индексировании документов. Если формат не указан, используется значение по умолчанию. Значение по умолчанию находится в производной таблице SYSIBMTS.TSDEFAULTS, в строке с DEFAULTNAME='FORMAT';. Для столбцов типа данных XML используется формат по умолчанию 'XML'; независимо от значения DEFAULTNAME. Для использования формата INSO должна быть установлена поддержка расширенного текста.

#### **UPDATE FREQUENCY**

Задаёт частоту выполнения обновлений индекса. Индекс обновляется, если число изменений не меньше значения, заданного для **UPDATE MINIMUM**. Частота обновления **NONE** указывает, что индекс не обновляется. Частота обновления **NONE** может пригодиться для текстового столбца в таблице, данные в котором никогда не изменяются. Это полезно также, если пользователь хочет обновлять индекс вручную (с помощью команды **UPDATE INDEX**). Автоматические обновления могут выполняться, только если выполнена команда **START FOR TEXT** и запущены службы экземпляра DB2 Text Search.

Значение частоты по умолчанию берется из производной таблицы SYSIBMTS.TSDEFAULTS, из строки с DEFAULTNAME='&apos;UPDATEFREQUENCY&apos;.

#### **NONE**

Индекс не будет обновляться. Обновление должно запускаться вручную.

**D** Дни недели для обновления индекса.

\* Каждый день недели.

*целое1*

Определенные дни недели, от воскресенья (0) до субботы (6)

**H** Часы указанных дней для обновления индекса.

\* Каждый час дня.

*целое2*

Определенные часы дня, от полуночи до 11 вечера: от 0 до 23

**M** Минуты указанных часов для обновления индекса.

*целое3*

Указывается как начало часа (0) или как кратное 5 число минут с начала часа: 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 или 55

#### **UPDATE MINIMUM** *минимальное\_число\_изменений*

Задаёт минимальное число изменений в текстовых документах перед инкрементным обновлением индекса в момент, заданный в **UPDATE FREQUENCY**. Допускаются только положительные целые значения. Значение по умолчанию берется из производной таблицы SYSIBMTS.TSDEFAULTS, из строки с DEFAULTNAME='UPDATEMINIMUM'.

**Примечание:** Это значение игнорируется командой **UPDATE INDEX** (если в ней не указана опция **USING UPDATE MINIMUM**). Меньшее значение улучшает соответствие между столбцом таблицы и индексом текстового поиска. Однако оно также приводит к дополнительной нагрузке на систему.

#### **COLLECTION DIRECTORY** *каталог*

Задаёт каталог, в котором хранится собрание индексов текстового поиска. Надо задать абсолютный путь; максимальная длина абсолютного пути - 215 символов. У владельца процесса службы экземпляра сервера текстового поиска должен быть доступ для чтения и записи в этот каталог.

Параметр **COLLECTION DIRECTORY** поддерживается только для установки интегрированного сервера текстового поиска. Дополнительную информацию о положениях собраний смотрите в замечаниях об использовании.

#### **ADMINISTRATION TABLES IN** *имя\_табличного\_пространства*

Задаёт имя существующего не временного табличного пространства для управляющих таблиц, создаваемых для индекса.

Для однораздельных баз данных, если табличное пространство не задано, будет использовано табличное пространство базовой таблицы, для которого создается индекс.

Для многораздельных баз данных необходимо при создании текстовых индексов необходимо использовать параметр **ADMINISTRATION TABLES** в явном виде. Чтобы промежуточные таблицы для индекса текстового поиска были распределены аналогично соответствующей базовой таблице, табличное пространство должно находиться в той же группе разделов, что и табличное пространство базовой таблицы.

### INDEX CONFIGURATION (значение\_опции)

Задаёт дополнительные значения, связанные с индексом, в виде пар строк опция/значение. Поддерживаются следующие значения:

Таблица 11. Спецификации для опции - значения

Опция	Значение	Тип данных	Описание
<b>COMMENT</b>	<i>текст</i>	Строка длиной до 512 байтов	Добавляет строковый комментарий в столбец REMARKS в производной таблице каталога DB2 Text Search TSINDEXES. Добавляет также значение строкового комментария в качестве описания собрания.
<b>UPDATEAUTOCOMMIT</b>	<i>счетчик_принятия</i>	Целое	<p>Эта опция задает число изменений индекса, после которого автоматически выполняется принятие для сохранения сделанной работы, для первого обновления индекса или для инкрементных обновлений.</p> <ul style="list-style-type: none"><li>Первоначальное обновление индекса обрабатывает пакеты документов из базовой таблицы после того, как будет активирован триггер для захвата изменений данных. Когда число измененных документов достигает COMMITCOUNT, сервер выполняет промежуточное принятие. Записи журнала, сгенерированные необработанными документами, будут удалены из таблицы кэша. Использование опции <b>UPDATEAUTOCOMMIT</b> для первоначального обновления текстового индекса ведет к значительному увеличению времени выполнения.</li><li>Для инкрементных обновлений обработанные записи журнала удаляются из таблицы кэша с каждым промежуточным принятием. COMMITCOUNT подсчитывает число измененных документов, а не число записей таблицы кэша.</li></ul>
<b>AUXLOG</b>	<i>значение_доп_журнала</i>	Строка	Управляет созданием дополнительной инфраструктуры журнала для захвата изменений, не распознанных триггером. Значение по умолчанию для таблиц с разделением по диапазонам - ON, в противном случае - OFF. Это значение по умолчанию можно изменить в таблице по умолчанию на значение AuxLogNorm для таблиц без разделения по диапазонам и на AuxLogPart для таблиц с разделением по диапазонам.

Таблица 11. Спецификации для опции - значения (продолжение)

Опция	Значение	Тип данных	Описание
<b>CJKSEGMENTATION</b>	<i>CJKSEGMENTATION_method</i>	Строка длиной до 512 байтов	<p>Задаёт метод сегментации для документов на китайском, японском и корейском языках (наборы локалей zh_CN, zh_TW, ja_JP, ko_KR), в том числе документов при включенном автоопределении языка (LANGUAGE AUTO). Если никакая опция не задана, будет применено значение для <b>CJKSEGMENTATION</b> в таблице значений по умолчанию. Поддерживаемые значения:</p> <ul style="list-style-type: none"> <li>• MORPHOLOGICAL</li> <li>• NGRAM</li> </ul> <p>Указанный метод сегментации добавлен в SYSIBMTS.TSCONFIGURATION - управляющая производная таблица, ее нельзя изменить после создания текстового индекса.</p>

**Напоминание:** Нечисловые значения, например, комментарии, должны указываться в одинарных кавычках. Знак одинарной кавычки внутри строкового значения указывается как две одинарных кавычки подряд.

**Пример:**

```
INDEX CONFIGURATION (COMMENT 'Index on User''s Guide column')
```

#### опции разделов

Зарезервировано для внутреннего использования IBM.

#### **CONNECT TO** *имя\_базы\_данных*

Это условие задает базу данных, с которой будет установлено соединение. База данных должна находиться в локальной системе. Если условие задано, то оно имеет приоритет перед переменной среды **DB2DBDFT**. Это условие можно опустить, если выполнены следующие условия:

- Для переменной среды **DB2DBDFT** задано допустимое имя базы данных.
- Пользователь, запускающий эту команду, обладает необходимой авторизацией для соединения с сервером баз данных.

#### **USER** *имя-пользователя* **USING** *пароль*

Это условие задает имя авторизации и пароль, используемые для установки соединения.

### Замечания по использованию

Все ограничения и соглашения об именовании, применимые к объектам баз данных и запросам DB2, применяются также к возможностям и запросам текстового поиска DB2. Идентификаторы, относящиеся к текстовому поиску DB2, должны отвечать соглашениям именованию DB2. Есть также некоторые дополнительные ограничения. Например, идентификаторы должны иметь вид:

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

или

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

В случае успешного завершения команды **CREATE INDEX**:

- Обновляются данные сервера DB2 Text Search. На каждый раздел базы данных создается собрание с именем *экземпляр\_имя\_базы\_данных\_идентификатор\_индекса\_номер*, как в следующем примере:  
tigertail\_MYTSDB\_TS250517\_0000

Имя собрания можно получить из столбца COLLECTIONNAME производной таблицы SYSIBMTS.TSCOLLECTIONNAMES.

- Обновляется информация в каталоге DB2 Text Search. В указанном табличном пространстве создается таблица кэша индексов с соответствующими индексами DB2. Кроме того, в указанном табличном пространстве создается таблица событий индексирования. Если задана опция `auxlog on`, будет создана вторая промежуточная таблица для захвата изменений посредством обработки целостности.
- Если DB2 Text Search используется одновременно с DB2 Net Search Extender и уже существует активный индекс Net Search Extender для данного столбца таблицы, новый текстовый индекс деактивируется.
- Вновь созданный индекс текстового поиска не заполняется автоматически. Для заполнения индекса текстового поиска нужно запустить команду **UPDATE INDEX** вручную или автоматически (в результате определения расписания обновлений для этого индекса с помощью опции **UPDATE FREQUENCY**).
- Если задана частота, для планировщика администратора DB2 будет создана задача расписания.

Ограничения использования:

- Для таблицы должен быть задан первичный ключ. В DB2 Text Search можно без каких либо ограничений по типу данных использовать многостолбцовые первичные ключи DB2. Число столбцов в первичном ключе должно быть не более числа столбцов в первичном ключе, разрешенного DB2, минус 2.
- Общая длина всех столбцов первичного ключа для таблицы с индексами DB2 Text Search должна быть не больше максимальной общей длины первичного ключа, разрешенной DB2, минус 15 байтов. Посмотрите описание ограничений оператора DB2 CREATE INDEX.

Нельзя одновременно вызывать для индекса текстового поиска несколько команд, если они могут конфликтовать. Если эта команда введена во время работы конфликтующей команды, произойдет ошибка и команда завершится неудачно; после этого можно попытаться запустить команду еще раз. Конфликтующая команда:

- **DISABLE DATABASE FOR TEXT**

Свойство вспомогательного журнала для текстового индекса нельзя изменить после создания индекса.

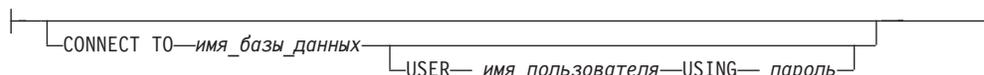
В среде многораздельных баз данных должны выполняться следующие условия:

- Управляющие таблицы, специфичные для индекса поиска, например, промежуточные таблицы, а также индексы текстового поиска, распределены аналогично соответствующей базовой таблице. При создании индексов текстового поиска используйте условие **ADMINISTRATION TABLES IN**, чтобы заданное табличное пространство было в той же группе разделов, что и табличное пространство базовой таблицы.

Опция **CJKSEGMENTATION** применяется к наборам локалей zh\_CN, zh\_TW, ja\_JP and ko\_KR для китайского, японского и корейского языков. Опции MORPHOLOGICAL и



## Возможности соединения:



## Параметры команды

### FORCE

Задаёт, что все индексы текстового поиска будут отброшены из базы данных принудительно.

Если эта опция не задана, а для этой базы данных определены индексы текстового поиска, команда завершится неудачно.

Если задана эта опция и служба текстового поиска DB2 ещё не запущена (не введена команда db2ts **START FOR TEXT**), индексы (собрания) текстового поиска не отбрасываются, и их нужно очистить вручную с помощью команды **db2ts CLEANUP**.

### CONNECT TO имя\_базы\_данных

Это условие задаёт базу данных, с которой надо установить соединение. База данных должна находиться в локальной системе. Если условие задано, оно имеет приоритет перед переменной среды DB2DBDFT. Это условие можно опустить, если выполнены следующие условия:

- Для переменной среды DB2DBDFT задано допустимое имя базы данных.
- Пользователь, запускающий эту команду, обладает необходимой авторизацией для соединения с сервером баз данных.

### USER имя-пользователя USING пароль

Это условие задаёт имя авторизации и пароль, используемые для установки соединения.

## Замечания по использованию

Эта команда не влияет на состояние включения DB2 Net Search Extender для базы данных. Она удаляет таблицы и производные таблицы каталога DB2 Text Search, созданные командой **ENABLE FOR TEXT**.

Перед отбрасыванием базы данных DB2 с определениями индексов текстового поиска запустите эту команду и убедитесь, что все текстовые индексы и собрания удалены успешно.

Если какие-либо индексы не удаётся удалить при помощи опции **FORCE**, имена таких собраний записываются в файлы журнала **db2diag**.

**Примечание:** Не рекомендуется выполнять действия, которые приводят к возникновению бесхозных собраний, например, собраний, остающихся определёнными на сервере текстового поиска, но не используемых DB2. Вот некоторые ситуации, которые приводят к появлению бесхозных собраний:

- Когда команда процессора командной строки **DROP DATABASE** выполняется без предварительного запуска команды **DISABLE DATABASE FOR TEXT**
- Когда команда **DISABLE DATABASE FOR TEXT** выполняется с использованием опции **FORCE**.
- Некоторые другие ситуации ошибок.



### **CONNECT TO** *имя\_базы\_данных*

Это условие задает базу данных, с которой будет установлено соединение. База данных должна находиться в локальной системе. Если условие задано, оно имеет приоритет перед переменной среды DB2DBDFT. Это условие можно опустить, если выполнены следующие условия:

- Для переменной среды DB2DBDFT задано допустимое имя базы данных.
- Пользователь, запускающий эту команду, обладает необходимой авторизацией для соединения с сервером баз данных.

### **USER** *имя-пользователя* **USING** *пароль*

Это условие задает имя авторизации и пароль, используемые для установки соединения.

## **Замечания по использованию**

Нельзя одновременно вызывать для индекса текстового поиска несколько команд, если эти команды могут конфликтовать. Если эта команда введена во время работы конфликтующей команды, происходит ошибка и команда завершается неудачно; после этого можно попытаться запустить команду еще раз. Ниже приведены часто встречающиеся конфликтующие команды:

- **DROP INDEX**
- **UPDATE INDEX**
- **CLEAR EVENTS FOR INDEX**
- **ALTER INDEX**
- **DISABLE DATABASE FOR TEXT**

Команда **STOP FOR TEXT**, выполняемая параллельно с операцией **DROP**, не вызывает сообщения о конфликтующей команде, вместо этого, если сервер текстового поиска завершил работу до удаления собрания командой **DROP**, будет возвращено сообщение об ошибке, информирующее, что сервер текстового поиска недоступен.

После отбрасывания индекса текстового поиска текстовый поиск по соответствующему столбцу становится невозможным. Если вы планируете создать новый индекс для того же текстового столбца, перед созданием нового индекса текстового поиска надо отсоединиться от базы данных, а затем соединиться с ней снова.

Команда **db2ts DROP INDEX FOR TEXT** выполняет в базе данных следующие изменения:

- Обновляет информацию в каталоге DB2 Text Search.
- Отбрасывает таблицы кэша индекса и таблицы событий.
- Удаляет триггеры для пользовательской текстовой таблицы.
- Уничтожает собрание, связанное с этим определением индекса текстового поиска DB2.

---

## **db2ts ENABLE DATABASE FOR TEXT**

Команда **db2ts ENABLE DATABASE FOR TEXT** включает текстовый поиск DB2 для текущей базы данных. Она создает административные таблицы и производные таблицы, задает значения по умолчанию для параметров и должна быть успешно выполнена прежде, чем можно будет создавать индексы текстового поиска по столбцам таблиц в базе данных. Перед командой нужно ввести в командной строке префикс db2ts.

После включения базы данных необходимо задать информацию о соединении для сервера текстового поиска в производной таблице SYSIBMTS.TSSERVERS. В операцию включения входит попытка заполнения сервера данными, и если нет доступа к конфигурации сервера, будет выведено предупреждение. В любом случае рекомендуется проверить информацию о соединении в производной таблице. Подробности смотрите в теме об изменении информации сервера текстового поиска DB2.

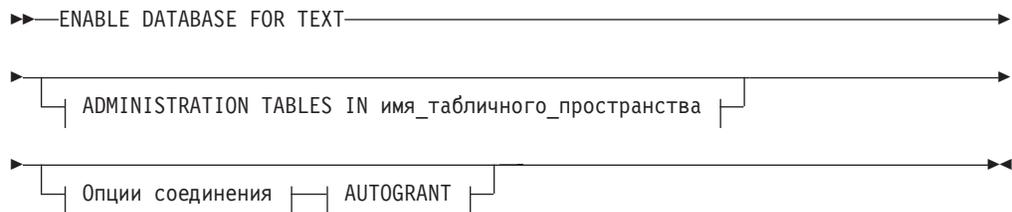
## Авторизация

- Привилегии ID авторизации этого оператора должны включать в себя роль SYSTS\_ADM и полномочия DBADM.

## Обязательное соединение

База данных

## Синтаксис команды



## Возможности соединения:



## Параметры команды

### ADMINISTRATION TABLES IN *имя\_табличного\_пространства*

Задаёт имя существующего постоянного табличного пространства для управляющих таблиц, созданных при включении в базе данных текстового поиска DB2. Рекомендуется, чтобы это табличное пространство находилось в группе разделов базы данных IBMCATGROUP. В многораздельной базе данных для пула буферов и табличного пространства должны быть задан размер страницы 32 Кбайта.

Если это условие не задано, в качестве табличного пространства по умолчанию используется SYSTOOLSPACE. В этом случае проверьте, что SYSTOOLSPACE уже существует. Если оно не существует, воспользуйтесь для его создания процедурой SYSPROC.SYSINSTALLOBJECTS.

**Примечание:** Используйте кавычки для задания регистрозависимого имени табличного пространства.

### AUTOGRANT

Эта опция устарела и больше не предоставляет полномочий владельцу экземпляра. Его использование более не рекомендуется, и сам оператор может быть удален в будущем выпуске.

### CONNECT TO *имя\_базы\_данных*

Это условие задает базу данных, с которой будет установлено соединение. База

данных должна находиться в локальной системе. Если условие задано, оно имеет приоритет перед переменной среды DB2DBDFT. Это условие можно опустить, если выполнены следующие условия:

- Для переменной среды DB2DBDFT задано допустимое имя базы данных.
- Пользователь, запускающий эту команду, обладает необходимой авторизацией для соединения с сервером баз данных.

**USER** *имя-пользователя* **USING** *пароль*

Это условие задает имя авторизации и пароль, используемые для установки соединения.

## Пример

Пример 1: Включить для базы данных создание управляющих таблиц текстового поиска DB2 в табличном пространстве tsspace и возврат сообщений об ошибках на английском языке.

```
CALL SYSPROC.SYSTS_ENABLE('ADMINISTRATION TABLES IN tsspace', 'en_US', ?)
```

Ниже приведен пример вывода этого запроса.

Значение выходных параметров

-----

Имя параметра : MESSAGE

Значение параметра: Операция выполнена успешно.

Состояние возврата = 0

## Замечания по использованию

При успешном выполнении эта команда выполняет следующие действия:

- Включает возможность текстового поиска DB2 для базы данных.
- Устанавливает значения по умолчанию конфигурации базы данных DB2 Text Search в производной таблице SYSIBMTS.TSDEFAULTS.
- Создает следующие управляющие производные таблицы текстового поиска DB2 в схеме SYSIBMTS:
  - SYSIBMTS.TSDEFAULTS
  - SYSIBMTS.TSLOCKS
  - SYSIBMTS.TSINDEXES
  - SYSIBMTS.TSCONFIGURATION
  - SYSIBMTS.TSCOLLECTIONNAMES
  - SYSIBMTS.TSSERVERS

---

## db2ts HELP

Команда **db2ts HELP** выводит список доступных команд текстового поиска DB2 или синтаксис отдельной команды.

Команда **db2ts HELP** служит также для получения справки по конкретным сообщениям об ошибках.

Для выполнения необходимо перед командой ввести **db2ts** в командной строке.

## Авторизация

Нет.

## Синтаксис команды



## Параметры команды

### HELP | ?

Выводит информацию справки для команды или кода ошибки.

#### *команда*

Первые ключевые слова указывают команду текстового поиска DB2 Text Search:

- ALTER
- CLEANUP
- CLEAR (и для CLEAR COMMAND LOCKS, и для CLEAR EVENTS FOR INDEX)
- CREATE
- DISABLE
- DROP
- ENABLE
- RESET PENDING
- START
- STOP
- UPDATE

#### *sqlcode*

SQLCODE для сообщения, возвращаемого командой db2ts (в хранимой процедуре управления или вне ее) или запросом текстового поиска.

#### *sqlstate*

SQLSTATE, возвращаемое командой, хранимой процедурой управления или запросом текстового поиска.

#### *идентификатор\_ошибки*

Идентификатор составляет часть *сообщения\_об\_ошибке\_текстового\_поиска*, встроенную в сообщения об ошибках. Этот идентификатор начинается с 'CIE' и имеет вид CIE#####, где ##### - число. Этот идентификатор представляет конкретную ошибку, возвращаемую при текстовом поиске. Он может также быть возвращен в информационном сообщении при завершении команды текстового поиска или в сообщении по завершении процедуры управления текстовым поиском. Если идентификатор не начинается с 'CIE', **db2ts help** не сможет вывести информацию о таком *идентификаторе\_ошибки*. Например, db2ts не может дать справку о сообщении с *идентификатором\_ошибки* IQQR0012E.

## Замечания по использованию

При использовании оболочки UNIX может оказаться необходимым задать аргументы для **db2ts** в двойных кавычках, как в следующем примере:

```
db2ts "? CIE00323"
```

Без этих кавычек оболочка попытается заменить символ подстановки содержимым рабочего каталога, что может привести к непредсказуемым результатам.

Если в качестве первого ключевого слова задана какая-либо из команд db2ts, выводится синтаксис указанной команды. Если две команды db2ts начинаются с одного и того же ключевого слова (**CLEAR COMMAND LOCKS** и **CLEAR EVENTS FOR INDEX**), в результате ввода db2ts help clear выводится синтаксис обеих команд; можно вывести справку по каждой команде конкретно, добавив второе ключевое слово, чтобы различить их, например, db2ts help clear events. Если после ? или **HELP** не задан параметр, db2ts выводит список всех доступных команд db2ts.

При задании *sqlcode*, *sqlstate* или CIEидентификатор\_ошибки выводится информация о SQLCODE, SQLSTATE или идентификаторе ошибки. Например,  
db2ts help SQL20423

или

```
db2ts ? 38H10
```

или

```
db2ts ? CIE00323
```

---

## Команда db2ts RESET PENDING

Запускает оператор SET INTEGRITY для всех текстовых промежуточных таблиц, связанных с конкретной таблицей.

Некоторые команды переводят промежуточные таблицы текстового поиска DB2 в состояние ожидания, блокируя другие операции базы данных и текстового поиска. При использовании команды **db2ts RESET PENDING** нет необходимости искать все текстовые индексы и связанные с ними промежуточные таблицы и затем запускать для каждой таблицы оператор SET INTEGRITY.

После отключения раздела данных необходимо ввести команду **RESET PENDING**, чтобы обновить содержимое промежуточной таблицы.

### Авторизация

Для этой команды требуется роль SYSTS\_MGR и одни из следующих полномочий и привилегий:

- Полномочия DATAACCESS
- Привилегия CONTROL для базовой таблицы, на которой определен текстовый индекс

**Примечание:** Для роли SYSTS\_MGR предоставляются сейчас привилегии ALL, что позволяет создавать и отбрасывать новые таблицы индексов. Однако если для таблицы индекса неявно создается независимый объект, авторизация не распространяется дальше. Чтобы удалить зависимый объект, пользователю нужно предоставить привилегию CONTROL.

### Обязательное соединение

Эту команду надо запускать с сервера баз данных DB2.

## Синтаксис команды

►►—RESET PENDING FOR TABLE— *схема-таблицы.имя-таблицы*—FOR TEXT—►►

└─|опции-соединения|─┘

### Опции-соединения:

└─CONNECT TO—*имя\_базы\_данных*—┘

└─USER—*ID-пользователя*—USING—*пароль*—┘

## Параметры команды

### *имя-таблицы*

Имя таблицы, для которой была добавлена текстовая промежуточная инфраструктура и требуется обработка целостности.

### *схема-таблицы*

Схема таблицы, для которой была запущена команда, завершившаяся переходом в режим ожидания.

## Замечания по использованию

После ввода команды, переводящей соответствующие таблицы в режим ожидания, например, команды **LOAD** с параметром **INSERT** или после ввода команды, требующей обновления зависимых таблиц оператором **SET INTEGRITY**, например, выполнения оператора **ALTER TABLE ... DETACH**, воспользуйтесь командой **RESET PENDING**.

---

## db2ts START FOR TEXT,

Команда **db2ts START FOR TEXT** запускает службы экземпляра DB2 Text Search, поддерживающие другие команды управления для DB2 Text Search и возможность указывать индексы текстового поиска в запросах SQL.

Если сервер сконфигурирован для поддержки расширенного текста, команда **db2ts START FOR TEXT** также запускает процессы для поддержки расширенного текста на компьютере хоста, где запущен сервер текстового поиска DB2.

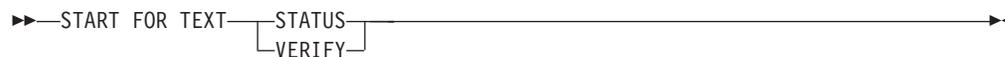
Эту команду надо запускать с сервера баз данных DB2.

Чтобы запустить службы экземпляра в среде многораздельных баз данных с помощью интегрированной установки текстового поиска, запустите эту команду с компьютера хоста интегрированного сервера текстового поиска. По умолчанию компьютер хоста интегрированного сервера текстового поиска - это хост сервера раздела базы данных с наименьшим номером.

## Авторизация

Владелец экземпляра. Никаких привилегий базы данных не требуется

## Синтаксис команды



## Параметры команды

### STATUS

Проверяет состояние сервера текстового поиска DB2. Будет возвращено подробное информационное сообщение с указанием состояния сервера STARTED (запущен) или STOPPED (остановлен).

### VERIFY

Проверяет состояние запуска сервера текстового поиска DB2 и завершается стандартным сообщением с кодом возврата 0, указывающим, что операция выполнена успешно. При любом другом состоянии сервера текстового поиска или же невозможности проверить состояние возвращается ненулевой код.

## Примеры

- Проверка, запущен ли сервер текстового поиска.

```
Linux/UNIX:
$ db2ts START FOR TEXT VERIFY
CIE00001 Операция завершена успешно.
```

```
$ echo $?
0
```

```
Windows:
C:\> db2ts START FOR TEXT VERIFY
CIE00001 Операция завершена успешно.
```

```
C:\> echo %ERRORLEVEL%
0
```

## Замечания по использованию

- В среде многораздельных баз данных команду **db2ts START FOR TEXT** с параметрами **STATUS** и **VERIFY** можно запускать на любом хосте сервера разделов. Чтобы запустить службы экземпляра, необходимо ввести команду **db2ts START FOR TEXT** на компьютере хоста интегрированного сервера текстового поиска. Компьютер хоста интегрированного сервера текстового поиска - это хост сервера раздела базы данных с наименьшим номером. Если используются пользовательские каталоги собраний, проверьте, не были ли позже созданы разделы с меньшими номерами. Это ограничение особенно важно для платформ Linux и UNIX. Если при создании экземпляра вы конфигурируете текстовый поиск DB2, нужно вначале задать хост интегрированного сервера текстового поиска. Этот хост всегда должен оставаться в данной конфигурации хостом сервера раздела базы данных с наименьшим номером.
- На платформах Windows есть служба Windows, связанная с каждым экземпляром DB2 для текстового поиска DB2. Имя этой службы можно узнать, введя команду:
 

```
DB2TS - <имя_экземпляра>[-<номер_раздела>]
```

. Кроме команды **db2ts START FOR TEXT**, для запуска службы можно воспользоваться также панелью управления или командой **NET START**.

---

## db2ts STOP FOR TEXT

Команда **db2ts STOP FOR TEXT** останавливает службы экземпляра текстового поиска DB2. Если запущенные службы включают процессы для поддержки расширенного текста, эти службы также будут остановлены.

Эту команду надо запускать с сервера баз данных DB2.

При выполнении этой команды из командной строки снабдите команду префиксом **db2ts** в командной строке DB2.

Эта команда обеспечивает удобную остановку автономного сервера текстового поиска, что возможно также в его собственной среде установки с помощью предлагаемого сценария. Если службы экземпляра уже остановлены, команда только сообщает его состояние и сообщает его пользователю.

### Авторизация

Владелец экземпляра. Никаких привилегий базы данных не требуется

### Синтаксис команды

►—STOP FOR TEXT—  
└──STATUS──┐  
└──VERIFY──┘

### Параметры команды

#### STATUS

Проверяет состояние серверов текстового поиска DB2. Будет возвращено подробное информационное сообщение с указанием состояния серверов STARTED (запущен) или STOPPED (остановлен).

#### VERIFY

Проверяет состояние остановки сервера текстового поиска DB2. Завершается стандартным сообщением с кодом возврата 0, указывающим, что команда выполнена успешно. В противном случае сервер текстового поиска возвращает ненулевой код, указывающий на неудачное завершение.

### Замечания по использованию

- Чтобы избежать прерывания выполнения одновременно запущенных команд перед вводом команды **db2ts STOP FOR TEXT** убедитесь, что не активны никакие другие команды управления, например, **db2ts UPDATE INDEX FOR TEXT**.
- В среде многораздельных баз данных команду **db2ts START FOR TEXT** с параметрами **STATUS** и **VERIFY** можно запускать на любом хосте сервера разделов.
- В среде многораздельных баз данных на платформах Windows, использующих интегрированный сервер текстового поиска, остановите службы экземпляра с помощью команды **db2ts STOP FOR TEXT**, запустив ее с компьютера хоста интегрированного сервера текстового поиска. По умолчанию компьютер хоста интегрированного сервера текстового поиска - это хост сервера раздела базы данных с наименьшим номером. Запуск команды на компьютере хоста интегрированного сервера текстового поиска обеспечивает остановку все служб и процессов. Если команда запущена на хосте другого сервера раздела, службу DB2TS надо будет остановить отдельно с помощью команды NET STOP.

## db2ts UPDATE INDEX,

Команда **db2ts UPDATE INDEX** обновляет индекс текстового поиска (собрание в текстовом поиске DB2 Text Search), чтобы отразить текущее содержимое текстового столбца, с которым связан индекс. Во время выполнения изменения возможен поиск. До завершения изменения поиск работает с частично обновленным индексом.

Для выполнения необходимо перед командой ввести **db2ts** в командной строке.

### Авторизация

Привилегии ID авторизации этого оператора должны включать в себя роль SYSTS\_MGR и хотя бы одно из следующих полномочий:

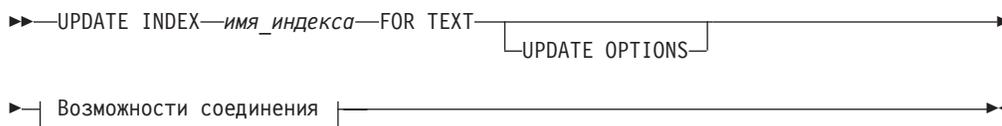
- Полномочия DATAACCESS
- Привилегия CONTROL для таблицы, на которой определен текстовый индекс
- Привилегия INDEX с условием SELECT для базовой таблицы, на которой определен текстовый индекс

Кроме того, требования авторизации для начального обновления применяются, как указано в операторе **CREATE TRIGGER**.

### Обязательное соединение

База данных

### Синтаксис команды



#### Возможности соединения:



### Параметры команды

#### UPDATE INDEX *имя\_индекса*

Задаёт имя индекса текстового поиска для обновления. Это имя должно подчиняться ограничениям именования для индексов DB2.

#### UPDATE OPTIONS

Входной аргумент типа VARCHAR(32K), задающий опции обновления. Если не задано никаких опций, обновление запускается безусловно. Возможные значения:

значение UPDATE OPTIONS	Описание
USING UPDATE MINIMUM	Эта опция требует использовать значение UPDATE MINIMUM, заданное для индекса текстового поиска, и обрабатывает обновления после того, как выполнено заданное минимальное количество изменений.

значение UPDATE OPTIONS	Описание
FOR DATA REDISTRIBUTION	Эта опция задает, что индекс текстового поиска в многораздельной базе данных требует обновления после того, как разделы данных были добавлены или удалены и выполнена последующая операция перераспределения данных. Результаты поиска могут быть несогласованными, пока индекс текстового поиска не будет обновлен с опцией FOR DATA REDISTRIBUTION.
ALLROWS	Эта опция задает, что начальное обновление должно предприниматься безусловно.
UPDATEAUTOCOMMIT	<p>Эта опция задает число изменений индекса, после которого автоматически выполняется принятие для сохранения сделанной работы, для первого обновления индекса или для инкрементных обновлений.</p> <ul style="list-style-type: none"> <li>Первоначальное обновление индекса обрабатывает пакеты документов из базовой таблицы после того, как будет активирован триггер для захвата изменений данных. Когда число измененных документов достигает COMMITCOUNT, сервер выполняет промежуточное принятие. Записи журнала, сгенерированные необработанными документами, удаляются из таблицы кэша. Использование опции UPDATEAUTOCOMMIT для первоначального обновления текстового индекса ведет к значительному увеличению времени выполнения.</li> <li>Для инкрементных обновлений обработанные записи журнала удаляются из таблицы кэша с каждым промежуточным принятием. COMMITCOUNT подсчитывает число измененных документов, а не число записей таблицы кэша.</li> </ul>

#### **CONNECT TO** *имя\_базы\_данных*

Это условие задает базу данных, с которой будет установлено соединение. База данных должна находиться в локальной системе. Если условие задано, оно имеет приоритет перед переменной среды DB2DBDFT. Это условие можно опустить, если выполнены следующие условия:

- Для переменной среды DB2DBDFT задано допустимое имя базы данных.
- Пользователь, запускающий эту команду, обладает необходимой авторизацией для соединения с сервером баз данных.

#### **USER** *имя-пользователя* **USING** *пароль*

Это условие задает имя авторизации и пароль, используемые для установки соединения.

## Замечания по использованию

Все ограничения и соглашения об именовании, применимые к объектам баз данных и запросам DB2, применяются также к возможностям и запросам текстового поиска DB2. Идентификаторы, относящиеся к текстовому поиску DB2, должны отвечать соглашениям именовании DB2. Есть также некоторые дополнительные ограничения. Например, идентификаторы должны иметь вид:

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

или

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

Если для текстового индекса созданы словари синонимов, при выполнении опций изменения ALLROWS и FOR DATA REDISTRIBUTION словари для существующих собраний будут удалены. После добавления разделов базы данных могут появиться новые собрания, связанные с текстовым индексом. Нужно будет снова добавить словари синонимов для всех связанных собраний.

Возврат из этой команды не происходит, пока не будет полностью завершена обработка обновления индекса. Продолжительность зависит от числа индексируемых документов и от числа уже проиндексированных документов. Имя собрания для индекса можно получить из столбца COLLECTIONNAME производной таблицы SYSIBMTS.TSCOLLECTIONNAMES.

Нельзя одновременно вызывать для индекса текстового поиска несколько команд, если они могут конфликтовать. Если эта команда запущена во время работы конфликтующей команды, произойдет ошибка и команда завершится неудачно; после этого можно попытаться запустить команду еще раз. Ниже приведены часто встречающиеся конфликтующие команды:

- **UPDATE INDEX**
- **CLEAR EVENTS FOR INDEX**
- **ALTER INDEX**
- **DROP INDEX**
- **DISABLE DATABASE FOR TEXT**

**Примечание:** Если в отдельных документах есть ошибки, эти документы надо исправить. Первичные ключи документов с ошибками можно найти в таблице событий для индекса. Следующая команда **UPDATE INDEX** переиндексирует эти документы, если соответствующие строки в пользовательской таблице изменены.

Изменения в базе данных:

- В таблицу событий вставляются строки (с информацией об ошибках анализа текстового поиска DB2).
- В случае инкрементного обновления удаляется информация из таблицы кэша индекса.
- Перед первым обновлением создаются триггеры для пользовательской текстовой таблицы.
- Обновляется собрание.
- Новые или измененные документы анализируются и индексируются.
- Удаленные документы удаляются из индекса.



---

## Глава 11. Хранимые процедуры DB2 Text Search

DB2 Text Search содержит несколько управляющих подпрограмм SQL для выполнения команд и для возврата сообщения о результатах выполняемых команд и кодов возврата для них.

При помощи управляющих подпрограмм SQL можно запускать следующие команды **db2ts**:

- Включить базу данных - **SYSPROC.SYSTS\_ENABLE**
- Конфигурировать базу данных - **SYSPROC.SYSTS\_CONFIGURE**
- Отключить базу данных - **SYSPROC.SYSTS\_DISABLE**
- Создать текстовый индекс - **SYSPROC.SYSTS\_CREATE**
- Обновить текстовый индекс - **SYSPROC.SYSTS\_UPDATE**
- Изменить текстовый индекс - **SYSPROC.SYSTS\_ALTER**
- Отбросить текстовый индекс - **SYSPROC.SYSTS\_DROP**
- Очистить события для текстового индекса - **SYSPROC.SYSTS\_CLEAR\_EVENTS**
- Очистить блокировки команд - **SYSPROC.SYSTS\_CLEAR\_COMMANDLOCKS**
- Сбросить состояние ожидания - **SYSPROC.SYSTS\_ADMIN\_CMD**
- Очистить неактивные индексы - **SYSPROC.SYSTS\_CLEANUP**

---

### Процедура **SYSTS\_ADMIN\_CMD** - Запуск команд управления текстовым поиском

Процедура **SYSTS\_ADMIN\_CMD** используется программами для запуска команд управления текстовым при помощи оператора SQL CALL.

#### Авторизация

Привилегия EXECUTE для процедуры SYSTS\_ADMIN\_CMD и требуемая авторизация, как перечислено для затребованной операции.

#### Привилегия PUBLIC по умолчанию

В базе данных без ограничений при автоматическом создании процедуры привилегии EXECUTE предоставляется значение PUBLIC.

#### Синтаксис

►—SYSTS\_ADMIN\_CMD—(—строка\_команды—, —локаль\_сообщений—, —сообщение—)————►

Схема - SYSPROC.

#### Параметры процедуры

*строка-команды*

Входной аргумент типа VARCHAR (32 K), задающий для выполнения одну команду управления индексом DB2 Text Search. Синтаксис команды совпадает с синтаксисом в DB2 Text Search, за исключением опций соединения, которые не

поддерживаются в этой процедуре. Команды, которые вы вводите через эту процедуру, используют текущее соединение.

Эта процедура поддерживает следующие команды текстового поиска DB2:

- ALTER INDEX
- CLEAR COMMAND LOCKS
- CLEAR EVENTS
- CREATE INDEX
- DISABLE DATABASE
- DROP INDEX
- ENABLE DATABASE
- RESET PENDING
- UPDATE INDEX

#### *локаль\_сообщений*

Входной аргумент типа VARCHAR(33), задающий локаль для всех возвращаемых текстовых сообщений об ошибках. Если аргумент - пустое значение или пустая строка, или же на сервере недоступны файлы сообщений для указанной локали, используется 'en\_US'.

#### *сообщение*

Выходной аргумент типа VARCHAR(32K), задающий предупреждение или информационное сообщение для успешно завершенной операции.

## Пример

В следующем примере индекс текстового поиска MYTEXTINDEX обновляется в схеме DB2TS, а сообщения об ошибках возвращаются по-русски:

```
CALL SYSPROC.SYSTS_ADMIN_CMD  
( 'UPDATE INDEX DB2TS.MYTEXTINDEX FOR TEXT', 'ru_RU', ? );
```

Пример вывода выглядит так:

Значение выходных параметров

-----

Имя параметра : MESSAGE

Значение параметра: CIE00001 Операция завершена успешно.

Состояние возврата = 0

## Замечания по использованию

Если команда выполняется неудачно, возвращаются коды ошибок SQLCODE -20427 и SQLSTATE 38H14 с конкретным сообщением, относящимся к текстовому поиску. Например, если индекс MYTEXTINDEX существует, и введен следующий оператор:

```
CALL SYSPROC.SYSTS_ADMIN_CMD ('CREATE INDEX MYTEXTINDEX FOR TEXT  
ON DB2TS.TEXTBOOKS (STORY)', 'ru_RU', ?)
```

Создание индекса завершилось неудачно со следующим сообщением об ошибке:

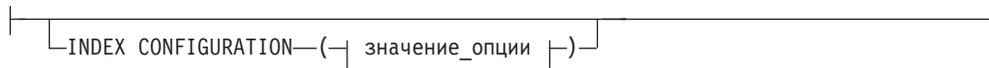
```
SQL20427N Ошибка при выполнении процедуры или команды  
управления текстовым поиском. Сообщение об ошибке "CIE00201 Индекс текстового поиска  
"DB2TS"."MYTEXTINDEX" уже существует. ". SQLSTATE=38H14
```

Если процедура возвращает SQLCODE, сообщение может быть усечено. Полная информация сообщения содержится в файлах журнала **db2diag**.

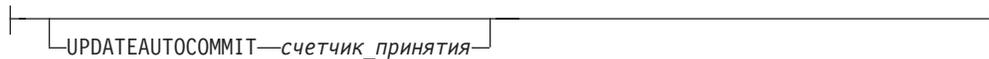




### опции конфигурации индекса:



### значение\_опции:



### опции активации:

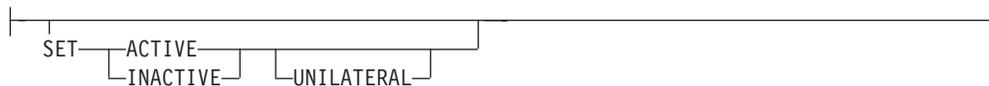


Схема - SYSPROC.

## Параметры процедуры

### *схема\_индекса*

Входной аргумент типа VARCHAR(128), задающий схему индекса текстового поиска. Значение *схема\_индекса* должно соответствовать ограничениям именования для имен схем DB2. Если аргумент является пустым значением или пустой строкой, используется значение CURRENT SCHEMA Аргумент *схема\_индекса* регистрозависим.

### *имя\_индекса*

Входной аргумент типа VARCHAR(128), задающий имя индекса. Вместе с аргументом *схема\_индекса* уникальным образом задает индекс текстового поиска в базе данных. Аргумент *имя\_индекса* регистрозависим.

### Изменить характеристики

Входной аргумент типа VARCHAR(32K), задающий опции изменения. Возможны следующие опции изменения:

#### UPDATE FREQUENCY

Задает частоту выполнения обновлений индекса. Индекс будет обновлен, если число изменений не меньше значения, заданного для **UPDATE MINIMUM**.

Частота обновления **NONE** указывает, что индекс не будет обновляться. Это может пригодиться для текстового столбца в таблице, данные в которой никогда не изменяются. Это полезно также, если пользователь хочет обновлять индекс вручную (с помощью команды **UPDATE INDEX**).

Автоматические обновления могут быть выполнены только в том случае, если задана переменная реестра DB2\_ATS\_ENABLE и введена команда **START FOR TEXT**.

Значение частоты по умолчанию берется из производной таблицы SYSIBM.TSDEFAULTS, из строки с DEFAULTNAME='UPDATEFREQUENCY'.

**NONE**

К этому текстовому индексу не будет применяться автоматическое обновление. Все последующие обновления индекса должны запускаться вручную.

**D** День недели для обновления индекса.

\* Каждый день недели.

*целое1*

Определенные дни недели, от воскресенья (0) до субботы (6)

**H** Час указанного дня для обновления индекса.

\* Каждый час дня.

*целое2*

Определенные часы дня, от полуночи до 11 вечера: от 0 до 23

**M** Минута указанного часа для обновления индекса.

*целое3*

Если вы не задали опцию **UPDATE FREQUENCY**, параметры частоты обновления не изменяются.

**UPDATE MINIMUM** *минимальное\_число\_изменений*

Задаёт минимальное число изменений в текстовых документах перед инкрементным обновлением индекса. Несколько изменений одного и того же текстового документа рассматриваются как отдельные изменения. Если вы не задали опцию **UPDATE MINIMUM**, этот параметр не изменяется.

**INDEX CONFIGURATION** (*значение\_опции*)

Начиная с версии 9.7 с пакетом Fix Pack 3 и более новыми пакетами Fix Pack, этот необязательный входной аргумент типа VARCHAR(32K) позволяет изменять параметры конфигурации текстового индекса. Поддерживаются следующие опции:

Таблица 12. Спецификации для опции - значения

Опция	Значение	Тип данных	Описание
SERIALUPDATE	<i>updatemode</i>	Целое	<p>Задаёт, в каком режиме (в параллельном или в последовательном) надо запускать обработку многораздельного индекса текстового поиска. В параллельном режиме выполнение распределено по разделам базы данных и выполняется независимо для каждого узла. В последовательном режиме выполнение запускается без распределения и останавливается в случае возникновения ошибки. Выполнение в последовательном режиме обычно занимает больше времени, но требует значительно меньше ресурсов.</p> <ul style="list-style-type: none"> <li>• 0 = параллельный режим</li> <li>• 1 = последовательный режим</li> </ul>

Таблица 12. Спецификации для опции - значения (продолжение)

Опция	Значение	Тип данных	Описание
UPDATEAUTO COMMIT	<i>commitcount</i> <i>_number</i>	Целое	<p>Задаёт число изменений индекса, после которого автоматически выполняется принятие для сохранения сделанной работы, для первого обновления индекса или для инкрементных обновлений.</p> <ul style="list-style-type: none"> <li>Первоначальное обновление индекса обрабатывает пакеты документов из базовой таблицы после того, как будет активирован триггер для захвата изменений данных. Когда число изменённых документов достигает COMMITCOUNT, сервер выполняет промежуточное принятие. Записи журнала, сгенерированные необработанными документами, будут удалены из таблицы кэша. Использование опции UPDATEAUTOCOMMIT для первоначального обновления текстового индекса ведёт к значительному увеличению времени выполнения.</li> <li>Для инкрементных обновлений обработанные записи журнала удаляются из таблицы кэша с каждым промежуточным принятием. COMMITCOUNT подсчитывает число изменённых документов, а не число записей таблицы кэша.</li> </ul>

*опции активации*

Начиная с версии 9.7 с пакетом Fix Pack 3 и более новыми пакетами Fix Pack, этот входной аргумент типа integer задаёт состояние текстового индекса.

## ACTIVE

Задаёт активное состояние текстового индекса

## INACTIVE

Задаёт неактивное состояние текстового индекса

## UNILATERAL

Задаёт одностороннее изменение, которое действует только на состояние индексов текстового поиска DB2. Если указан этот аргумент, меняется на активное или неактивное состояние только индекса текстового поиска DB2. Без аргумента UNILATERAL состояния активации индексов DB2 Text Search и DB2 Net Search Extender меняются синхронно, так что активным будет только один из этих текстовых индексов.

### локаль\_сообщений

Входной аргумент типа VARCHAR(33), задающий локаль, используемую для всех возвращаемых сообщений об ошибках. Если аргумент - пустое значение или пустая строка, или же на сервере недоступны файлы сообщений для указанной локали, используется 'en\_US'.

### сообщение

Выходной аргумент типа VARCHAR(32K), задающий предупреждение или информационное сообщение для успешно завершённой операции.

## Примеры

*Пример 1:* В следующем примере изменяются характеристики обновления индекса текстового поиска. Этот индекс был создан изначально со *схемой\_индекса* 'db2ts' и *именем\_индекса* 'myTextIndex'. Задание 'UPDATE FREQUENCY NONE' указывает более не обновлять индекс текстового поиска, так как, вероятно, изменения в связанном столбце таблицы не ожидаются. Указано, что сообщения об ошибках должны возвращаться по-английски. При успешном выполнении процедуры вызывающему возвращается сообщение выходного параметра.

```
CALL SYSPROC.SYSTS_ALTER('db2ts', 'myTextIndex',
'UPDATE FREQUENCY NONE', 'ru_RU', ?)
```

Ниже приведен пример вывода этого запроса.

```
Значение выходных параметров
-----
Имя параметра   : MESSAGE
Значение параметра: Операция выполнена успешно.

Return Status = 0
```

*Пример 2:* В следующем примере хранимая процедура SYSTS\_ALTER вызывается для индекса текстового поиска со значениями *схема\_индекса* 'db2ts' и *имя\_индекса* 'myTextIndex2'. Сделана попытка задать обновление этого индекса в начале каждого часа. Однако такой индекс не существует, и происходит ошибка.

```
CALL SYSPROC.SYSTS_ALTER('db2ts', 'myTextIndex2',
'update frequency D(*) H(*) M(0)', 'ru_RU', ?)
```

Ниже приведен пример вывода этого запроса.

```
SQL20427N Ошибка при выполнении процедуры или команды
управления текстовым поиском. Сообщение об ошибке "C1E00316 Индекс текстового поиска
"db2ts"."myTextIndex2" не существует. ". SQLSTATE 38H14
```

## Замечания по использованию

- Процедуры управления текстовыми индексами используют уже существующее соединение с базой данных. Рекомендуется перед выполнением этой процедуры

управления текстовым поиском выполнить принятие всех изменений, внесенных транзакциями, чтобы избежать нежелательного влияния принятия или отката этой процедуры. Один из способов достигнуть этого - включить AUTOCOMMIT.

- Нельзя одновременно запускать для индекса текстового поиска несколько процедур или команд, если они могут конфликтовать. Некоторые из конфликтующих процедур и команд:
  - Процедура SYSTS\_ALTER или команда db2ts ALTER INDEX
  - Процедура SYSTS\_CLEAR\_EVENTS или команда db2ts CLEAR EVENTS FOR INDEX
  - Процедура SYSTS\_DISABLE или команда db2ts DISABLE DATABASE FOR TEXT
  - Процедура SYSTS\_DROP или команда db2ts DROP INDEX
  - Процедура SYSTS\_UPDATE или команда db2ts UPDATE INDEX
  - процедура SYSTS\_CONFIGURE

В случае конфликта процедура возвращает SQLCODE -20426 и SQLSTATE 38N13.

- При запуске этой процедуры для изменения частоты создается, изменяется или отбрасывается задача планирования для текстового индекса.
- Результат активации индексов зависит от исходного состояния индекса. Эти результаты описаны в следующей таблице.

Таблица 13. Изменения состояния при отсутствии недопустимых индексов:

Начальное состояние DB2 Text Search или Net Search Extender	опция активации Active	опция активации Active Unilateral	опция активации Inactive	опция активации Inactive Unilateral
Активен / Неактивен	Без изменений	Без изменений	Неактивен / Активен	Неактивен / Неактивен
Неактивен / Активен	Активен / Неактивен	Ошибка	Без изменений	Без изменений
Неактивен / Неактивен	Активен / Неактивен	Активен / Неактивен	Неактивен / Активен	Без изменений

При конфликтах активных индексов возвращаются сообщения об ошибках SQL20427N и CIE0379E.

## Процедура SYSTS\_CLEANUP - Удаление недопустимых индексов текстового поиска

Эта процедура удаляет недопустимые индексы текстового поиска и связанные с ними собрания из базы данных. Индекс может стать недопустимым, когда производятся операции базы данных, влияющие на все содержимое таблиц, например, при усечении.

Эта процедура - альтернатива команде **db2ts CLEANUP FOR TEXT** в области действия базы данных. Эту команду нужно использовать для удаления устаревших собраний на уровне экземпляра.

### Авторизация

Привилегии ID авторизации этого оператора должны содержать роль SYSTS\_ADM и полномочия DBADM и DATAACCESS.

## Привилегия PUBLIC по умолчанию

Не требуется

### Синтаксис

►—SYSTS\_CLEANUP—(—опции—, —локаль\_сообщений—, —сообщение—)►

Схема - SYSPROC.

### Параметры процедуры

#### опции

Входной аргумент типа VARCHAR(32K). Принимается только NULL или пустой параметр. Зарезервирован для внутреннего использования IBM.

#### локаль\_сообщений

Входной аргумент типа VARCHAR(33), задающий локаль, используемую для всех возвращаемых сообщений об ошибках. Если аргумент - пустое значение или пустая строка, или же на сервере недоступны файлы сообщений для указанной локали, используется 'en\_US'.

#### сообщение

Выходной аргумент типа VARCHAR(32K), задающий предупреждение или информационное сообщение для успешно завершенной операции.

### Пример

*Пример 1:* В следующем примере отбрасываются все недопустимые индексы текстового поиска в текущей базе данных. Для сообщений задан русский язык, а когда процедура завершается успешно, вызывающему возвращается сообщение с обозначающим это параметром вывода.

```
CALL SYSPROC.SYSTS_CLEANUP('', 'ru_RU', ?)
```

Значение выходных параметров

-----

Имя параметра : MESSAGE

Значение параметра: Операция выполнена успешно.

Return Status = 0

*Пример 2:* Вывод в случае возникновения ошибки:

SQL0462W Команда или подпрограмма "SYSTS\_CLEANUP" (с уникальным именем "\*N") вернула код предупреждения в SQLSTATE и диагностическое сообщение "CIE00212W 2 из 20 собраний не могут быть удалены. Подробности смотрите в файле db2diag.log.". SQLSTATE=01H14

---

## Процедура SYSTS\_CLEAR\_COMMANDLOCKS - Удаление блокировок команд для индексов текстового поиска

Эта процедура удаляет все блокировки команд для определенного индекса текстового поиска или для всех индексов текстового поиска в базе данных.

### Авторизация

Привилегии ID авторизации оператора, используемого для очистки блокировок индекса, должны включать в себя следующие полномочия:

- Роль SYSTS\_MGR

- Полномочия DBADM или привилегия CONTROL для базовой таблицы, в которой определен индекс

Привилегии ID авторизации оператора, используемого для очистки блокировок соединения базы данных, должны включать в себя роль SYSTS\_ADM.

## Привилегия PUBLIC по умолчанию

None (отсутствует)

Блокировка команды создается в начале выполнения команды текстового поиска и уничтожается при завершении этой команды. Она предотвращает конфликты между различными командами. Использование этой процедуры требуется в том редком случае, когда блокировка остается неудаленной в результате неожиданного поведения системы, и ее требуется очистить явно.

Эта процедура передает на сервер базы данных команду управления текстовым поиском **CLEAR COMMAND LOCKS**.

## Синтаксис

```
►►—SYSTS_CLEAR_COMMANDLOCKS—(—схема_индекса—, —имя_индекса—, —————►
►—локаль_сообщений—, —сообщение—)—————►◄
```

Схема - SYSPROC.

## Параметры процедуры

### *схема\_индекса*

Входной аргумент типа VARCHAR(128), задающий схему текстового индекса. Значение *схема\_индекса* должно соответствовать ограничениям именования для имен схем DB2. Если аргумент является пустым значением или пустой строкой, используется значение CURRENT SCHEMA. Аргумент *схема\_индекса* регистрозависим.

### *имя\_индекса*

Входной аргумент типа VARCHAR(128), задающий имя индекса. Вместе с аргументом *схема\_индекса* уникальным образом задает индекс текстового поиска в базе данных. Если аргумент является пустым значением или пустой строкой, процедура удаляет блокировки команд для всех индексов текстового поиска в базе данных. Аргумент *имя\_индекса* регистрозависим.

### *локаль\_сообщений*

Входной аргумент типа VARCHAR(33), задающий локаль, используемую для всех возвращаемых сообщений об ошибках. Если аргумент - пустое значение или пустая строка, или же на сервере недоступны файлы сообщений для указанной локали, используется 'en\_US'.

### *сообщение*

Выходной аргумент типа VARCHAR(32K), задающий предупреждение или информационное сообщение для успешно завершенной операции.

## Примеры

*Пример 1:* В этом примере SYSTS\_CLEAR\_COMMANDLOCKS вызывается для индекса текстового поиска с именем\_схемы 'db2ts' и именем\_индекса

'myTextIndex'. Сообщения об ошибках должны возвращаться по-русски. При успешном выполнении процедуры вызывающему возвращается сообщение выходного параметра.

```
CALL SYSPROC.SYSTS_CLEAR_COMMANDLOCKS('db2ts', 'myTextIndex', 'ru_RU', ?)
```

Ниже приведен пример вывода этого запроса.

Значение выходных параметров

-----

Имя параметра : MESSAGE

Значение параметра: Операция выполнена успешно.

Return Status = 0

*Пример 2:* В следующем примере SYSTS\_CLEAR\_COMMANDLOCKS вызывается для очистки блокировок команд для индекса текстового поиска со *схемой\_индекса* 'db2ts' и *именем\_индекса* 'myTextIndex2'. Такой индекс не существует, и процедура возвращает ошибку.

```
CALL SYSPROC.SYSTS_CLEAR_COMMANDLOCKS('db2ts', 'myTextIndex2', 'ru_RU', ?)
```

Ниже приведен пример вывода этого запроса.

```
SQL20427N Ошибка при выполнении процедуры или команды  
управления текстовым поиском. Сообщение об ошибке "CIE00316 Индекс текстового поиска  
"db2ts"."myTextIndex2" не существует. ". SQLSTATE 38H14
```

## Замечания по использованию

- Процедуры управления текстовыми индексами используют уже существующее соединение с базой данных. Рекомендуется перед выполнением этой процедуры управления текстовым поиском выполнить принятие всех изменений, внесенных транзакциями, чтобы избежать нежелательного влияния принятия или отката этой процедуры. Один из способов достигнуть этого - включить AUTOCOMMIT.
- Вы вызываете эту процедуру потому, что процесс, которому принадлежит блокировка команды, не работает. В таком случае эта команда (владелец блокировки) не может завершиться, и индекс не может стать доступным. Вам надо предпринять действия по исправлению ситуации. Предположим, процесс, выполняющий команду DROP INDEX, неожиданно завис. Он удалил некоторые данные индексов, но не всю информацию каталога и собраний. Блокировка команды остается нетронутой. После очистки блокировки команды DROP INDEX вы можете захотеть повторить процедуру SYSTS\_DROP. В другом примере процесс, выполняющий команду UPDATE INDEX, прерывается. Этот процесс обработал несколько документов, но не все, и блокировка команды все еще существует. После проверки состояния индекса текстового поиска и очистки блокировки команды UPDATE INDEX можно повторно запустить команду UPDATE INDEX.
- При выполнении этой процедуры изменяется содержимое производной таблицы текстового поиска DB2 SYSIBMTS.TSLOCKS.

---

## Процедура SYSTS\_CLEAR\_EVENTS - Удаление событий индексации из таблицы событий индекса

Эта процедура удаляет события индексации из таблицы событий индекса, используемой для управления.

Имя таблицы событий можно найти в столбце EVENTVIEWNAME производной таблицы SYSIBMTS.TSINDEXES. Каждая операция обновления индекса, обработавшая хотя бы один документ, заносит информационные записи и, в

некоторых случаях, записи об ошибках в таблицу событий. Для автоматических обновлений таблицу событий следует регулярно проверять. Ошибки, связанные с конкретными документами, надо исправлять, изменяя содержимое документа. После исправления этих ошибок события могут быть (и должны быть ради экономии места) очищены.

Эта процедура передает на сервер базы данных команду управления текстовым поиском **CLEAR EVENTS FOR INDEX**.

## Авторизация

Привилегии ID авторизации этого оператора должны содержать оба следующих полномочия:

- Роль SYSTS\_MGR
- DBADM с полномочиями DATAACCESS или привилегией CONTROL для таблицы, на которой определен этот индекс

## Привилегия PUBLIC по умолчанию

None (отсутствует)

## Синтаксис

```
►►—SYSTS_CLEAR_EVENTS—(—схема_индекса—,—имя_индекса—,——————►  
►—локаль_сообщений—,—сообщение—)—————►
```

Схема - SYSPROC.

## Параметры процедуры

### *схема\_индекса*

Входной аргумент типа VARCHAR(128), задающий схему индекса текстового поиска. Значение *схема\_индекса* должно соответствовать ограничениям именования для имен схем DB2. Если аргумент является пустым значением или пустой строкой, используется значение CURRENT SCHEMA. Аргумент *схема\_индекса* регистрозависим.

### *имя\_индекса*

Входной аргумент типа VARCHAR(128), задающий имя индекса. Вместе с аргументом *схема\_индекса* уникальным образом задает индекс текстового поиска в базе данных. Аргумент *имя\_индекса* регистрозависим.

### *локаль\_сообщений*

Входной аргумент типа VARCHAR(33), задающий локаль, используемую для всех возвращаемых сообщений об ошибках. Если аргумент - пустое значение или пустая строка, или же на сервере недоступны файлы сообщений для указанной локали, используется 'en\_US'.

### *сообщение*

Выходной аргумент типа VARCHAR(32K), задающий предупреждение или информационное сообщение для успешно завершенной операции.

## Примеры

*Пример 1:* В этом примере команда SYSTS\_CLEAR\_EVENTS вызывается для индекса текстового поиска, созданного с именем\_схемы 'db2ts' и именем\_индекса

'myTextIndex'. Указано, что сообщения об ошибках должны возвращаться по-русски. При успешном выполнении процедуры вызывающему возвращается сообщение выходного параметра.

```
CALL SYSPROC.SYSTS_CLEAR_EVENTS('db2ts', 'myTextIndex', 'ru_RU', ?)
```

Ниже приведен пример вывода этого запроса.

Значение выходных параметров

-----

Имя параметра : MESSAGE

Значение параметра: Операция выполнена успешно.

Return Status = 0

*Пример 2:* В следующем примере SYSTS\_CLEAR\_EVENTS вызывается для очистки таблицы событий для индекса текстового поиска со *схемой\_индекса* 'db2ts' и *именем\_индекса* 'myTextIndex2'. Такой индекс не существует, и происходит ошибка.

```
CALL SYSPROC.SYSTS_CLEAR_EVENTS('db2ts', 'myTextIndex2', 'ru_RU', ?)
```

Ниже приведен пример вывода этого запроса.

```
SQL20427N Ошибка при выполнении процедуры или команды
управления текстовым поиском. Сообщение об ошибке "CIE00316 Индекс текстового поиска
"db2ts"."myTextIndex2" не существует. ". SQLSTATE 38H14
```

## Замечания по использованию

- В процедурах администрирования текстового поиска используется существующее соединение с базой данных. В зависимости от завершения этих процедур для текущей транзакции выполняется принятие или откат. Поэтому вам может быть полезно выполнять принятие всех изменений транзакции во избежание неожиданных последствий такого принятия или отката. Один из способов достигнуть этого - включить AUTOCOMMIT.
- Нельзя одновременно запускать для индекса текстового поиска несколько процедур или команд, если они могут конфликтовать. Некоторые из конфликтующих процедур и команд:
  - Процедура SYSTS\_ALTER или команда db2ts ALTER INDEX
  - Процедура SYSTS\_DISABLE или команда db2ts DISABLE DATABASE FOR TEXT
  - Процедура SYSTS\_DROP или команда db2ts DROP INDEX
  - процедура SYSTS\_CONFIGURE
  - Процедура SYSTS\_UPDATE или команда db2ts UPDATE INDEX

В случае конфликта процедура возвращает SQLCODE -20426 и SQLSTATE 38H13.

- Если планируются регулярные обновления (смотрите опции UPDATE FREQUENCY в процедурах SYSTS\_CREATE и SYSTS\_ALTER), таблицу событий надо проверять регулярно.
- Чтобы очистить таблицу событий текстового поиска DB2 для индекса текстового поиска, после проверки причины события и устранения источника ошибки используйте процедуру SYSTS\_CLEAR\_EVENTS или команду db2ts **CLEAR EVENTS FOR INDEX**.
- Убедитесь, что во все строки, упомянутые в таблице событий, внесены изменения. Изменяя строки в пользовательской таблице, вы гарантируете, что при новом запуске процедуры SYSTS\_UPDATE или команды db2ts **UPDATE INDEX** будет выполнена новая попытка проиндексировать ошибочные документы.
- При выполнении этой команды таблица событий очищается.

---

## Процедура SYSTS\_CONFIGURE - Конфигурирование текущей базы данных для текстового поиска

Процедура **SYSTS\_CONFIGURE** применяет информацию о соединении сервера текстового поиска к каталогу текстового поиска.

Некоторые свойства сервера текстового поиска должны быть отображены в управляющих таблицах текстового поиска (в каталоге текстового поиска). Эти свойства связываются с текстовым индексом при создании этого индекса. Когда параметры, например, маркер, периодически изменяются, это изменение должно быть отображено в базе данных, и должны быть соответственно изменены свойства для всех индексов.

Эта процедура первоначально требуется в следующих случаях:

- Неполное выполнение
- Конфигурирование автономного сервера текстового поиска
- Многораздельные базы данных
- И так далее, после любого изменения информации о соединении сервера текстового поиска

При последующих обновлениях убедитесь, что нет активных управляющих операций текстового поиска, и остановите текущий сконфигурированный сервер текстового поиска.

Во время включения базы данных управляющая производная таблица **SYSIBMTS.TSSERVER** изменяется с учетом информации о соединении для интегрированного сервера текстового поиска. Изучите информацию сервера текстового поиска и измените ее в производной таблице **SYSIBMTS.TSSERVER** на релевантные данные сервера текстового поиска, а затем запустите процедуру **SYSTS\_CONFIGURE** для применения измененной информации. При наличии в экземпляре нескольких баз данных сконфигурируйте каждую базу с учетом информации одного сервера текстового поиска.

В общем случае последовательность операций следующая:

1. Сконфигурируйте сервер текстового поиска. Интегрированные серверы текстового поиска можно сконфигурировать при установке или при создании экземпляра DB2. Автономный сервер текстового поиска конфигурируется отдельно.
2. Включите базу данных для текстового поиска, используя команду **db2ts ENABLE** или одну из процедур, **SYSTS\_ENABLE** или **SYSTS\_ADMIN\_CMD**, с опцией **ENABLE**.
3. Определите параметры для сервера текстового поиска, нужные для производной таблицы **SYSIBMTS.TSSERVERS**.
4. Измените управляющую производную таблицу **SYSIBMTS.TSSERVERS** с учетом параметров сервера текстового поиска.
  - Если производная таблица пустая, используйте оператор **INSERT**. Например:

```
INSERT INTO SYSIBMTS.TSSERVERS (HOST, PORT, TOKEN, SERVERSTATUS)
VALUES ('localhost', 55000, '9kfsjg48=', 0);
```
  - Если производная таблица уже содержит строку, используйте оператор **SQL UPDATE**. Например:

```
UPDATE SYSIBMTS.TSSERVERS SET (HOST, PORT, TOKEN) =
('tsmach1.ibm.com', 55002, 'k3j4fjk9u=')
```
5. Выполните процедуру **SYSTS\_CONFIGURE**.

## Авторизация

Привилегии ID авторизации этой процедуры должны содержать роль SYSTS\_ADM.

## Привилегия PUBLIC по умолчанию

None (отсутствует)

## Синтаксис

►—SYSTS\_CONFIGURE—(—options—, —локаль-сообщения—, —сообщение—)————►

Схема - SYSPROC.

## Параметр процедуры

### опции

Входной аргумент типа VARCHAR(32K), задающий используемые опции. Если никаких опций не требуется, в качестве аргумента можно указать пустое значение или пустую строку.

### локаль\_сообщений

Входной аргумент типа VARCHAR(33), задающий локаль, используемую для всех возвращаемых сообщений об ошибках. Если аргумент - пустое значение, пустая строка, или же на сервере недоступны файлы сообщений для указанной локали, используется en\_US.

### сообщение

Выходной аргумент типа VARCHAR(32K), задающий предупреждение или информационное сообщение для успешно завершенной операции.

## Пример

Пример 1: Подготовка базы данных для служб текстового поиска и возвращение любых сообщений по-русски.

```
CALL SYSTS_ENABLE('', 'ru_RU', ?)"
```

```
INSERT INTO SYSIBMTS.TSSERVERS  
(HOST,PORT,TOKEN,KEY,LOCALE,SERVERTYPE,SERVERSTATUS)  
VALUES ('tsmach1.ibm.com', 55000, '9kfsjg48=', 'ru_RU', 0);
```

```
CALL SYSPROC.SYSTS_CONFIGURE('', 'ru_RU', ?)
```

Пример вывода этого запроса:

Значение выходных параметров

-----

Имя параметра : MESSAGE

Значение параметра: Операция выполнена успешно.

Return Status = 0

## Замечания по использованию

- Производная таблица SYSIBMTS.TSSERVERS создается, когда база данных включается для текстового поиска. Эта производная таблица изменяется с учетом информации об интегрированном сервере текстового поиска во время включения базы данных. Если не удастся получить конфигурацию сервера текстового поиска, операция включения завершится с предупреждением 'неполное включение'.

- Процедуру SYSTS\_CONFIGURE нужно запускать всякий раз, когда строка вставляется в SYSIBMTS.TSSERVERS или изменяется. Убедитесь, что нет активных управляющих операций текстового поиска и остановите сервер текстового поиска до изменения SYSIBMTS.TSSERVERS.
- При изменении в базе данных SYSIBMTS.TSSERVERS те же параметры должны быть изменены во всех базах данных с включенным текстовым поиском. С данным экземпляром DB2 поддерживается только один сервер текстового поиска.
- Использование сконфигурированного сервера текстового поиска регистрируется для экземпляра при запуске процедуры SYSTS\_CONFIGURE. Если эта процедура не запускается, серьезная ошибка не возникает, но ответ на некоторые команды может быть неожиданным.
- Нельзя одновременно вызывать для индекса текстового поиска несколько процедур или команд, если они могут конфликтовать. Некоторые из конфликтующих процедур и команд:
  - процедура SYSTS\_ALTER
  - процедура SYSTS\_DISABLE
  - процедура SYSTS\_CONFIGURE
  - процедура SYSTS\_UPDATE

В случае конфликта процедура возвращает SQLCODE -20426 и SQLSTATE 38N13.
- Нужно изменить некоторые аспекты, относящиеся к установке текстового поиска и к конфигурации экземпляра DB2 для текстового поиска. К ним относятся:
  - Интегрируется указатель, используется ли текстовый поиск экземпляром DB2 (DB2 конфигурируется как часть экземпляра DB2), или это отдельная автономная установка сервера ECMTS.
  - Используется указатель, включена ли при конфигурировании текстового поиска поддержка расширенного текста.

---

## Процедура SYSTS\_CREATE - Создание индекса текстового поиска для столбца

Процедура SYSTS\_CREATE создает индекс текстового поиска для столбца после ввода команды DB2 Text Search **CREATE INDEX** на сервере баз данных. После создания и обновления индекса текстового поиска можно проводить поиск для данных в столбце с использованием функций текстового поиска.

Индекс не содержит никаких данных, пока не проведена операция обновления индекса. Операцию обновления можно запустить или с помощью интерфейса хранимой процедуры, или в интерфейсе командной строки. В дальнейшем вы можете непосредственно вводить команду DB2 Text Search **UPDATE INDEX**, или же эта команда может неявно вводиться планировщиком административных задач DB2 в соответствии с частотой обновлений, определенной для индекса

### Авторизация

У ID авторизации этого оператора должна быть роль SYSTS\_MGR и полномочия CREATETAB в базе данных, а также одна из следующих привилегий или один из типов полномочий:

- Привилегия CONTROL для таблицы, на которой будет определен индекс
- Привилегия INDEX для таблицы, на которой будет определен индекс, и одни из следующих полномочий:
  - Полномочия IMPLICIT\_SCHEMA для базы данных, если нет явного или неявного имени схемы индекса

- Привилегия CREATEIN для схемы, если имя схемы этого индекса существует
- Полномочия DBADM

## Привилегия PUBLIC по умолчанию

None (отсутствует)

### Синтаксис

```

▶▶ SYSTS_CREATE (—схема_индекса—, —имя_индекса—, — источник текста |—, —▶
▶ | варианты |—, —локаль_сообщений—, —сообщение—) —▶▶

```

#### источник текста:

```

|— [—схема-таблицы—] имя_таблицы— (— имя текстового столбца |—) —, —|—

```

#### имя текстового столбца:

```

|— имя_столбца —|—
|— имя_функции (имя_столбца) —|—

```

#### опции:

```

|—
|— информация о тексте по умолчанию |—
|— характеристики обновления |—
|— опции хранения |—
|— опции конфигурации индекса |—
|— partition options |—
|—

```

#### информация о тексте по умолчанию:

```

|— [CODEPAGE—кодовая_страница—] [LANGUAGE—язык—] [FORMAT—формат—] —|—

```

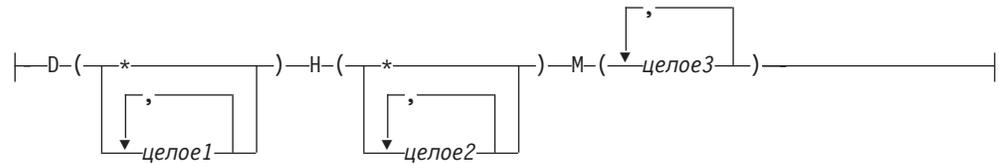
#### Изменить характеристики:

```

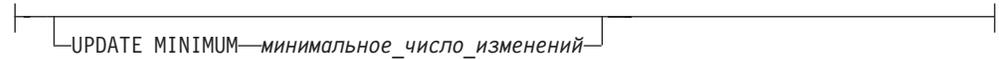
|—
|— UPDATE FREQUENCY—NONE—|—
|— частота обновления |—
▶ |— характеристики инкрементного обновления |—

```

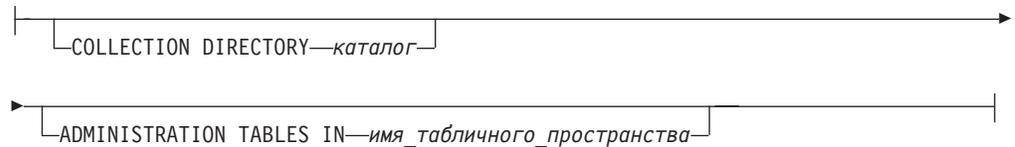
### частота обновления:



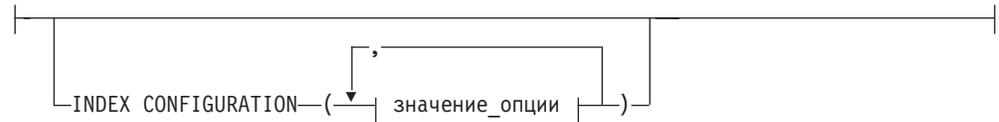
### характеристики инкрементного обновления:



### опции хранения:



### опции конфигурации индекса:



### значение\_опции:

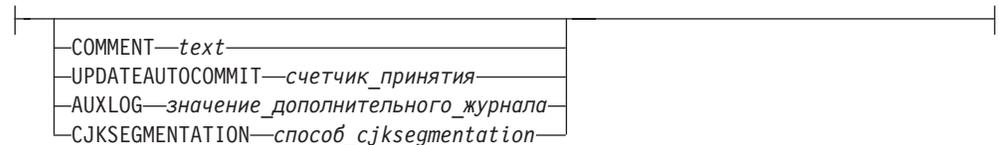


Схема - SYSPROC.

## Параметры процедуры

### схема\_индекса

Входной аргумент типа VARCHAR(128), задающий схему индекса текстового поиска. *схема-индекса* должна удовлетворять ограничениям именованию для имен схем DB2. Если аргумент - это пустое значение или пустая строка, используется значение схемы по умолчанию. Аргумент *схема\_индекса* регистрозависим.

### имя\_индекса

Входной аргумент типа VARCHAR(128), задающий имя индекса. Значение *схема\_индекса* вместе со значением *имя\_индекса* однозначно идентифицируют индекс текстового поиска в базе данных. Аргумент *имя\_индекса* регистрозависим.

### **источник текста**

Входной аргумент типа VARCHAR(1024), задающий имя столбца для индексирования. Используются следующие опции:

#### *схема-таблицы*

Схема таблицы, для которой будет создан индекс текстового поиска.

#### *имя-таблицы*

Задает имя таблицы, содержащей текстовый столбец. Нельзя создавать индексы текстового поиска для таблиц объединения, материализованных таблиц запросов и производных таблиц. Аргумент *имя\_таблицы* регистрозависим.

### **имя текстового столбца**

Задает имя столбца для индексирования.

#### *имя-столбца*

Задает имя столбца для индексирования. Столбец должен быть одного из следующих типов данных: CHAR, VARCHAR, CLOB, DBCLOB, BLOB, GRAPHIC, VARGRAPHIC или XML. Если тип данных столбца не совпадает ни с одним из перечисленных типов, можно использовать функцию преобразования с названием *схема-функции.имя-функции*, чтобы преобразовать тип данных столбца в один из допустимых типов. (Подробности смотрите в параметре *имя-функции (имя-столбца)*). Другой вариант - указать внешнюю пользовательскую функцию, обращающуюся к текстовым документам для индексации. Для столбца можно создать только один индекс текстового поиска. Аргумент *имя\_столбца* регистрозависим.

#### *имя\_функции (имя-столбца)*

Задает имя с указанием схемы внешней скалярной функции, преобразующей тип данных столбца, не поддерживаемый для поиска, в тип поддерживаемых для текстового поиска данных. Эта функция должна принимать ровно один параметр и возвращать ровно одно значение. Имя этой функции должно соответствовать соглашениям об именовании для DB2. Параметр *имя-функции(имя-столбца)* регистрозависим.

### **опции**

Входной аргумент типа VARCHAR(32), задающий используемые опции. Если никакие опции не нужны, этот параметр может быть или пустым значением, или пустой строкой. Используются следующие опции:

#### **CODEPAGE** *кодовая\_страница*

Задает кодовую страницу DB2 для использования при индексации текстовых документов. Значение кодовой страницы по умолчанию берется из производной таблицы SYSIBM.TSDEFAULTS, из строки со значением столбца DEFAULTNAME, равным CODEPAGE. Начальное значение по умолчанию кодовой страницы для новых индексов - это кодовая страница базы данных. Этот параметр применяется только к двоичным типам данных, то есть тип столбца или тип возврата из функции преобразования должен быть типом BLOB или символьным типом FOR BIT DATA.

#### **LANGUAGE** *язык*

Задает язык, используемый DB2 для зависящей от языка обработки документа при индексировании. Если вы не указали локаль, при определении значения по умолчанию для параметра **LANGUAGE** будет использована территория базы данных. Если вы хотите выполнять просмотр документов для определения локали, задайте для *локали* значение AUTO.

### **FORMAT** *формат*

Задает формат текстовых документов в столбце. Поддерживаемые форматы - TEXT, XML, HTML и INSO. Если тип данных столбца - не XML, формат по умолчанию берется из производной таблицы SYSIBMTS.TSDEFAULTS, из строки со значением столбца DEFAULTNAME, равным FORMAT. Если тип данных столбца - XML, формат по умолчанию всегда совпадает с XML. Если вы хотите использовать формат INSO, должна быть установлена поддержка расширенного текста.

### **UPDATE FREQUENCY**

Задает частоту обновления индекса. Индекс обновляется, если число изменений не меньше значения, заданного для параметра **UPDATE MINIMUM**. Автоматические обновления могут быть выполнены только в том случае, если задана переменная реестра DB2\_ATS\_ENABLE и введена команда **START FOR TEXT**.

Формат по умолчанию берется из производной таблицы the SYSIBMTS.TSDEFAULTS, из строки со значением столбца DEFAULTNAME, равным UPDATEFREQUENCY.

#### **NONE**

Индекс не будет обновляться. Это значение может пригодиться для текстового столбца в таблице, данные в которой никогда не изменяются. Это полезно также, если вы собираетесь обновить индекс вручную с помощью команды **UPDATE INDEX**.

#### **D** Дни недели для обновления индекса.

\* Каждый день недели.

*целое1*

Определенные дни недели, от воскресенья до субботы (от 0 до 6).

#### **H** Часы указанных дней для обновления индекса.

\* Каждый час дня.

*целое2*

Определенные часы дня, от полуночи до 11 вечера: от 0 до 23.

#### **M** Минуты указанных часов для обновления индекса.

*целое3*

### **UPDATE MINIMUM** *минимальное\_число\_изменений*

Задает минимальное число изменений в текстовых документах перед инкрементным обновлением индекса в момент, заданный параметром **UPDATE FREQUENCY**. Допустимы только целые положительные числа. Это значение по умолчанию берется из производной таблицы SYSIBMTS.TSDEFAULTS, из строки со значением столбца DEFAULTNAME, равным UPDATEMINIMUM. Небольшое значение повышает согласованность между столбцом таблицы и индексом текстового поиска, но одновременно вызывает дополнительную нагрузку на систему. Команда **UPDATE INDEX** игнорирует значение этого параметра, пока для нее не задан параметр **USING UPDATE MINIMUM**.

### **COLLECTION DIRECTORY** *каталог*

Задает каталог, в котором хранится собрание индексов текстового поиска. Надо задать абсолютный путь; максимальная длина абсолютного пути - 215 символов. У владельца службы экземпляра сервера Text Search должен быть доступ с правом чтения и записи к этому каталогу.

Параметр COLLECTION DIRECTORY поддерживается только в случае интегрированного конфигурирования сервера текстового поиска. Дополнительную информацию о положениях собраний можно получить в замечаниях об использовании.

#### **ADMINISTRATION TABLES IN** *имя\_табличного\_пространства*

Задаёт имя существующего не временного табличного пространства для управляющих таблиц, создаваемых для индекса. По умолчанию используется табличное пространство той базовой таблицы, для которой вы создаете индекс.

При создании индекса для многораздельной таблицы или для таблиц в многораздельных базах данных этот аргумент обязателен. Для многораздельных баз данных это табличное пространство должно находиться в той же группе разделов, что и табличное пространство базовой таблицы. Управляющие таблицы текстового поиска распространяются таким же образом, как соответствующая базовая таблица.

#### **INDEX CONFIGURATION** (*значение\_опции*)

Задаёт дополнительные относящиеся к индексу опции в виде пар опция-значение. Поддерживаются следующие опции.

Таблица 14. Пары опция-значение для параметра INDEX CONFIGURATION

Опция	Значение	Тип данных	Описание
COMMENT	<i>текст</i>	Строка длиной до 512 байтов	Добавляет строковый комментарий к столбцу REMARKS в производной таблице каталога DB2 Text Search SYSIBMTS.TSINDEXES. Этот комментарий используется также как описание собрания.

Таблица 14. Пары опция-значение для параметра INDEX CONFIGURATION (продолжение)

Опция	Значение	Тип данных	Описание
UPDATEAUTOCOMMIT	<i>commitcount _number</i>	Целое	<p>Задаёт число обновлений индекса, после которого выполняется принятие для сохранения сделанной работы, для первоначальных или инкрементных обновлений индекса:</p> <ul style="list-style-type: none"> <li>Первоначальное обновление индекса обрабатывает пакеты документов из базовой таблицы после того, как будет активирован триггер для захвата изменений данных. Когда число изменённых документов достигает значения <i>счётчик_принятия</i>, сервер выполняет промежуточное принятие. Записи журнала, сгенерированные необработанными документами, удаляются из таблицы кэша. Использование опции UPDATEAUTOCOMMIT для первоначального обновления текстового индекса ведёт к значительному увеличению времени выполнения.</li> <li>Для инкрементных обновлений обработанные записи журнала удаляются из промежуточной таблицы с каждым промежуточным принятием. <i>Счётчик_принятия</i> подсчитывает число изменённых документов, а не число записей промежуточной таблицы.</li> </ul>

Таблица 14. Пары опция-значение для параметра INDEX CONFIGURATION (продолжение)

Опция	Значение	Тип данных	Описание
AUXLOG	<i>значение</i> <i>_auxlog</i>	Строка	Управляет созданием инфраструктуры дополнительных журналов для захвата изменений, которые не распознаны триггером. Параметр по умолчанию для таблиц с разделением по диапазонам - ON. Значение по умолчанию в таблице по умолчанию можно изменить, задав для таблиц без разделения по диапазонам значение AuxLogNorm, а для таблиц с разделением по диапазонам - AuxLogPart.  Для текстового индекса нельзя изменить вспомогательное свойство инфраструктуры журнала после создания этого индекса.
CJKSEGMENTATION	<i>способ</i> <i>_CJKSEGMENTATION</i>	Строка	Способ сегментации применим для документов на китайском, японском и корейском языках (наборы локалей zh_CN, zh_TW, ja_JP, ko_KR), в том числе для документов с включенным автоматическим определением языка (LANGUAGE AUTO). Если никакие опции не заданы, применяется значение CJKSEGMENTATION из таблицы значений по умолчанию. Поддерживаемые значения: <ul style="list-style-type: none"><li>• MORPHOLOGICAL</li><li>• NGRAM</li></ul> Указанный метод сегментации добавлен в SYSIBMTS. TSCONFIGURATION - управляющая производная таблица, ее нельзя изменить после создания текстового индекса.

**Напоминание:** Нечисловые значения, такие как комментарии, должны указываться в одинарных кавычках. Знак одинарной кавычки внутри строкового значения указывается как две одинарных кавычки подряд.

**Пример:**

INDEX CONFIGURATION (COMMENT 'Index on User''s Guide column')

*локаль\_сообщений*

Входной аргумент типа VARCHAR(33), задающий локаль, используемую для

всех возвращаемых сообщений об ошибках. Если аргумент - пустое значение или пустая строка, или же на сервере недоступны файлы сообщений для указанной локали, используется 'en\_US'.

#### сообщение

Выходной аргумент типа VARCHAR(32), задающий предупреждение или информационное сообщение для успешно завершенной операции.

## Примеры

*Пример 1:* В следующем примере вызывается процедура SYSTS\_CREATE для создания индекса текстового поиска с именем myTextIndex. Опция **UPDATE MINIMUM 10** указывает, что до выполнения инкрементного обновления индекса должно произойти по меньшей мере 10 изменений текстовых документов, связанных с этим индексом. Все сообщения об ошибках возвращаются по-английски. В случае успешного завершения используемой команды текстового поиска в выходном параметре **message** возвращается состояние исполнения команды.

```
CALL SYSPROC.SYSTS_CREATE('db2ts', 'myTextIndex',
    'myUserSchema.myBaseTable (myTextColumn)', 'UPDATE MINIMUM 10',
    'ru_RU', ?)
```

Пример вывода выглядит так:

```
Значение выходных параметров
-----
Имя параметра      : MESSAGE
Значение параметра: Операция выполнена успешно.
Состояние возврата = 0
```

*Пример 2:* В следующем примере вызывается процедура SYSTS\_CREATE для создания индекса текстового поиска с именем myTextIndex2. Опции не заданы. В этом примере индекс существует, и поэтому возвращается сообщение об ошибке.

```
CALL SYSPROC.SYSTS_CREATE('db2ts', 'myTextIndex2',
    'myUserSchema.myBaseTable (myTextColumn)', '', 'ru_RU', ?)
```

Пример вывода выглядит так:

```
SQL20427N Ошибка при выполнении процедуры или команды
управления текстовым поиском. Сообщение об ошибке "CIE00201 Индекс текстового поиска
"db2ts"."myTextIndex2" уже существует. "
```

## Замечания по использованию

- В процедурах администрирования текстового поиска используется существующее соединение с базой данных. Если для предыдущей транзакции не выполнено принятие до вызова операции управления текстовым поиском, может возникнуть тупиковая ситуация, так как могут быть затронуты одни и те же объекты базы данных и операция будет ожидать завершения предыдущего шага. Это может случиться, например, если при выключенной опции AUTOCOMMIT создать таблицу, а затем создать текстовый индекс, не вызвав принятие в явном виде между этими двумя транзакциями.
- Если отсутствует условие COLLECTION DIRECTORY, собрание будет помещено в подкаталог, обозначенный после сгенерированного системой идентификатора индекса в пути, определенном параметром *defaultDataDirectory* в конфигурации Text Server. Для непосредственного конфигурирования этого параметра можно использовать утилиту configTool. Если параметр *defaultDataDirectory* непосредственно не сконфигурирован, подкаталог собрания располагается в каталоге *<configPath>/config/collections*. Настоятельно рекомендуется использовать положение, отличное от домашнего каталога или пути экземпляра базы данных, таких как *defaultDataDirectory* или

COLLECTION DIRECTORY. Убедитесь, что в этом положении есть достаточное пространство хранения и оно локально (не смонтировано в NFS).

- Сконфигурируйте значение для параметра **COLLECTION DIRECTORY**, которое позволяет использовать устройство RAID с чередованием.

Действуют следующие правила, связанные с ключами:

- Для таблицы нужно определить первичный ключ.
- Максимальное количество столбцов в первичном ключе должно быть на два меньше, чем разрешено для числа столбцов в первичном ключе для других сред DB2.
- Общая длина всех столбцов первичного ключа для таблицы с индексами DB2 Text Search должна быть не больше максимальной общей длины первичного ключа, разрешенной в других средах DB2, минус 15 байтов. Смотрите ограничения для оператора DB2 CREATE INDEX.

При создании индекса с параметром **LANGUAGE**, для которого задана опция AUTO, в качестве опции допускается спецификация CJKSEGMENTATION. Заданный способ сегментации применим к документам на китайском, японском и корейском языках. После завершения создания индекса заданное для *способ\_CJKSEGMENTATION* значение изменить нельзя.

Некоторые процедуры и команды нельзя выполнять для индекса текстового поиска одновременно, так как это может привести к сообщению об ошибке из-за синхронизации конфликтующих операций.

- процедура SYSTS\_DISABLE или команда **db2ts DISABLE DATABASE FOR TEXT**
- процедура SYSTS\_CONFIGURE

Успешное выполнение команды **CREATE INDEX** приводит к следующему:

- Обновляются данные сервера DB2 Text Search. Создается собрание с именем *экземпляр\_имя\_базы\_данных\_идентификатор\_индекса\_номер*, как в следующем примере:

```
tigertail_MYTSDB_TS250517_0000
```

В многораздельных базах данных создается собрание в каждом разделе. Имена собраний можно получить из столбца COLLECTIONNAME производной таблицы SYSIBMTS.TSCOLLECTIONNAMES.

- В указанном табличном пространстве создается таблица событий индексирования. Кроме того, в указанном табличном пространстве создается промежуточная таблица индексов с соответствующими индексами DB2. Если для параметра **INDEX CONFIGURATION** была задана опция AUXLOG ON, создается вторая промежуточная таблица для захвата изменений через обработку целостности.
- Обновляется информация в каталоге DB2 Text Search. Однако вновь созданный индекс текстового поиска автоматически не заполняется. Для заполнения индекса текстового поиска нужно вызвать процедуру **SYSTS\_UPDATE** или команду **UPDATE INDEX** вручную или автоматически (в результате определения расписания обновлений для этого индекса с помощью параметра **UPDATE FREQUENCY**).
- Если DB2 Text Search используется одновременно с DB2 Net Search Extender и уже существует активный индекс Net Search Extender для данного столбца таблицы, новый индекс DB2 Text Search деактивируется.
- Если индекс текстового поиска создан с параметром **LANGUAGE**, для которого задано значение AUTO, и с параметром **CJKSEGMENTATION** со значением MORPHOLOGICAL, процедуры поиска допустимых строк в морфологическом индексе

возвращают нулевые строки. Для получения результатов добавьте опцию QUERYLANGUAGE к функции CONTAINS, как показано в следующем запросе:

```
select bookname from morphobooks
where contains (story, '△', 'QUERYLANGUAGE=zh_CN') = 1
```

---

## Процедура SYSTS\_DISABLE - Выключение текущей базы данных для текстового поиска

Эта процедура отключает текстовый поиск DB2 для текущей базы данных.

Когда возможность текстового поиска отключена, индексы и команды текстового поиска для этой базы данных становятся недоступными.

Эта процедура передает на сервер базы данных команду управления текстовым поиском **DISABLE DATABASE FOR TEXT**.

### Авторизация

Привилегии ID авторизации этого оператора должны содержать следующие полномочия:

- Полномочия DBADM с DATAACCESS.
- Роль SYSTS\_ADM

### Привилегия PUBLIC по умолчанию

None (отсутствует)

### Синтаксис

►► SYSTS\_DISABLE (—опции—, —локаль\_сообщений—, —сообщение—) ►►

Схема - SYSPROC.

### Параметры процедуры

#### опции

Входной аргумент типа VARCHAR(128), задающий используемые опции отключения базы данных. Можно задать аргумент FORCE. Если задано это значение, все индексы отбрасываются и возможность текстового поиска отключается принудительно. Никакие индексы текстового поиска не сохраняются, и никаких сообщений об ошибках и предупреждений не возвращается. Если аргумент является пустым значением или пустой строкой, выполняется попытка отключения возможности текстового поиска для базы данных.

#### локаль\_сообщений

Входной аргумент типа VARCHAR(33), задающий локаль, используемую для всех возвращаемых сообщений об ошибках. Если аргумент - пустое значение или пустая строка, или же на сервере недоступны файлы сообщений для указанной локали, используется 'en\_US'.

#### сообщение

Выходной аргумент типа VARCHAR(32K), задающий предупреждение или информационное сообщение для успешно завершенной операции.

## Примеры

*Пример 1:* В следующем примере текстовый поиск отключается для базы данных при помощи процедуры SYSTS\_DISABLE. Задана опция FORCE, чтобы отключить возможность, даже если для таблиц базы данных все еще существуют индексы текстового поиска. Задано, что сообщения об ошибках возвращаются по-русски. Для выходного параметра *сообщение* задана строка информационного сообщения.

```
CALL SYSPROC.SYSTS_DISABLE('FORCE', 'ru_RU', ?)
```

Ниже приведен пример вывода этого запроса.

```
Значение выходных параметров
-----
Имя параметра      : MESSAGE
Значение параметра: Операция выполнена успешно.

Return Status = 0
```

*Пример 2:* В следующем примере текстовый поиск отключается для базы данных с существующими индексами текстового поиска при помощи процедуры SYSTS\_DISABLE без задания опции FORCE. В результате вызывающему возвращается сообщение об ошибке. Желательно отбросить все существующие индексы текстового поиска перед отключением возможности тестового поиска, или же можно задать опцию FORCE как значение входного параметра *опции*.

```
CALL SYSPROC.SYSTS_DISABLE('', 'ru_RU', ?)
```

Ниже приведен пример вывода этого запроса.

```
SQL20427N Ошибка при выполнении процедуры или команды
управления текстовым поиском. Сообщение об ошибке "CIE00326 В указанной базе данных или
базе данных по умолчанию активен индекс текстового поиска. ". SQLSTATE 38N14
```

## Замечания по использованию

- В процедурах администрирования текстового поиска используется существующее соединение с базой данных. Рекомендуется перед выполнением этой процедуры управления текстовым поиском выполнить принятие всех изменений, внесенных транзакциями, чтобы избежать нежелательного влияния принятия или отката этой процедуры. Один из способов достигнуть этого - включить AUTOCOMMIT.
- Нельзя одновременно вызывать для индекса текстового поиска несколько процедур или команд, если они могут конфликтовать. Некоторые из конфликтующих процедур:
  - процедура SYSTS\_ALTER
  - процедура SYSTS\_CLEAR\_EVENTS
  - процедура SYSTS\_DISABLE
  - процедура SYSTS\_CONFIGURE
  - процедура SYSTS\_UPDATE

В случае конфликта процедура возвращает SQLCODE -20426 и SQLSTATE 38N13.

**Примечание:** Для процедуры **SYSTS\_DISABLE** блокировка задается на уровне базы данных.

- При выполнении этой процедуры
  - обновляется информация в каталоге DB2 Text Search. Журнал индекса и таблицы событий отбрасываются. Триггеры для пользовательской текстовой таблицы удаляются.

- если задана опция **FORCE**, вся информация текстового индекса удаляется из базы данных, и все связанные собрания удаляются. Смотрите справку в темах "Команда db2ts DROP INDEX" и "Процедура SYSTS\_DROP".
- Эта процедура не влияет на состояние включения DB2 Net Search Extender для базы данных. Она удаляет таблицы и производные таблицы каталога текстового поиска DB2, созданные процедурой SYSTS\_ENABLE или командой **ENABLE FOR TEXT**.
- Перед отбрасывание базы данных DB2 с определениями индексов текстового поиска запустите эту процедуру и убедитесь, что все текстовые индексы и собрания удалены успешно.
- Если какие-либо индексы не удается удалить при помощи опции **FORCE**, имена таких собраний записываются в файлы журнала **db2diag**.

**Примечание:** Не рекомендуется выполнять действия, которые приводят к возникновению бесхозных собраний, то есть собраний, остающихся определенными на сервере текстового поиска, но не используемых DB2. Вот некоторые ситуации, которые могут привести к появлению бесхозных собраний:

- Когда команда DROP DATABASE CLP выполняется без запуска команды DISABLE DATABASE FOR TEXT
- Когда процедура SYSTS\_DISABLE или команда **DISABLE DATABASE FOR TEXT** выполняется с использованием опции **FORCE**.

---

## Процедура SYSTS\_DROP - Отбрасывание индекса текстового поиска

Процедура SYSTS\_DROP отбрасывает существующий индекс текстового поиска, связанный с любым столбцом таблицы.

После успешного выполнения этой процедуры запросы текстового поиска нельзя выполнять для столбца, для которого был отброшен индекс текстового поиска.

Эта процедура вызывает на сервере баз данных команду администрирования текстового поиска **DROP INDEX**.

### Авторизация

У ID авторизации этого оператора должна быть роль SYSTS\_MGR, а также одна из следующих привилегий или одно из полномочий:

- Привилегия CONTROL для таблицы, на которой определен индекс
- Привилегия DROPIN для схемы, на которой определен индекс
- Если у индекса текстового поиска есть существующее расписание, ID авторизации должен принадлежать создателю индекса или у него должны быть полномочия DBADM

### Привилегия PUBLIC по умолчанию

None (отсутствует)

### Синтаксис

►► SYSTS\_DROP (—схема\_индекса—, —имя\_индекса—, —опции—) ►►

Схема - SYSPROC.

## Параметры процедуры

### *схема\_индекса*

Входной аргумент типа VARCHAR(128), задающий схему индекса текстового поиска. Значение *схема\_индекса* должно соответствовать ограничениям именования для имен схем DB2. Если аргумент является пустым значением или пустой строкой, используется значение CURRENT SCHEMA. Аргумент *схема\_индекса* регистрозависим.

### *имя\_индекса*

Входной аргумент типа VARCHAR(128), задающий имя индекса. Вместе с аргументом *схема\_индекса* уникальным образом задает индекс текстового поиска в базе данных. Аргумент *имя\_индекса* регистрозависим.

### *варианты*

Входной аргумент типа VARCHAR(32000), задающий используемые опции. Если никаких опций не требуется, в качестве аргумента можно указать пустое значение или пустую строку.

### *локаль\_сообщений*

Входной аргумент типа VARCHAR(33), задающий локаль, используемую для всех возвращаемых сообщений об ошибках. Если аргумент - пустое значение или пустая строка, или же на сервере недоступны файлы сообщений для указанной локали, используется 'en\_US'.

### *сообщение*

Выходной аргумент типа VARCHAR(32K), задающий предупреждение или информационное сообщение для успешно завершенной операции.

## Примеры

*Пример 1:* В следующем примере выполняется отбрасывание индекса текстового поиска, созданного со значениями *схема\_индекса* 'db2ts' и *имя\_индекса* 'myTextIndex'. Указано, что сообщения об ошибках должны возвращаться по-английски. При успешном выполнении процедуры вызывающему возвращается сообщение выходного параметра.

```
CALL SYSPROC.SYSTS_DROP('db2ts', 'myTextIndex', '', 'en_US', ?)
```

Как и в предыдущих выпусках, поддерживается процедура SYSTS\_DROP без аргумента **options**, например:

```
CALL SYSPROC.SYSTS_DROP('db2ts', 'myTextIndex', 'ru_RU', ?)
```

Ниже приведен пример вывода этого запроса.

```
Значение выходных параметров
```

```
-----
```

```
Имя параметра : MESSAGE
```

```
Значение параметра: Операция выполнена успешно.
```

```
Return Status = 0
```

*Пример 2:* В следующем примере SYSTS\_DROP вызывается для отбрасывания индекса текстового поиска со значениями *схема\_индекса* 'db2ts' и *имя\_индекса* 'myTextIndex2'. Такой индекс не существует, и происходит ошибка.

```
CALL SYSPROC.SYSTS_DROP('db2ts', 'myTextIndex2', 'ru_RU', ?)
```

Ниже приведен пример вывода этого запроса.

```
SQL20427N Ошибка при выполнении процедуры или команды
управления текстовым поиском. Сообщение об ошибке "CIE00316 Индекс текстового поиска
"db2ts"."myTextIndex2" не существует. ". SQLSTATE 38N14
```

### Замечания по использованию

- Нельзя одновременно вызывать для индекса текстового поиска несколько процедур или команд, если они могут конфликтовать. Некоторые из конфликтующих процедур:

- процедура SYSTS\_ALTER
- процедура SYSTS\_CLEAR\_EVENTS
- процедура SYSTS\_DISABLE
- процедура SYSTS\_DROP
- процедура SYSTS\_UPDATE
- процедура SYSTS\_CONFIGURE

Команда STOP FOR TEXT, выполняемая одновременно с операцией DROP, не вызовет сообщения о конфликте команд, но вместо этого, если сервер текстового поиска выключится до того, как DROP удалит собрание, возвратится ошибка, что сервер текстового поиска недоступен.

- После отбрасывания индекса текстового поиска текстовый поиск по соответствующему столбцу становится невозможным. Если вы планируете создать новый индекс для того же текстового столбца, перед созданием нового индекса текстового поиска надо отсоединиться от базы данных, а затем соединиться с ней снова.
- Команда db2ts DROP INDEX производит следующие изменения в базе данных:
  - Обновляет информацию в каталоге DB2 Text Search.
  - Отбрасывает промежуточные таблицы индекса и таблицы событий.
  - Удаляет триггеры для пользовательской текстовой таблицы.
  - Уничтожает собрание, связанное с этим определением индекса текстового поиска DB2.

---

## Процедура SYSTS\_ENABLE - Включение текущей базы данных для текстового поиска

Процедура SYSTS\_ENABLE включает DB2 Text Search для текущей базы данных.

Эта процедура должна быть успешно выполнена прежде, чем можно будет создавать индексы текстового поиска по столбцам в базе данных.

Эта процедура передает на сервер базы данных команду управления текстовым поиском **ENABLE DATABASE FOR TEXT**.

### Авторизация

Привилегии ID авторизации этой процедуры должны содержать роль SYSTS\_ADM и полномочия DBADM.

### Привилегия PUBLIC по умолчанию

None (отсутствует)

## Синтаксис

►►SYSTS\_ENABLE(—опции—, —локаль\_сообщений—, —сообщение—)◄◄

### опции:

►► ADMINISTRATION TABLES IN имя\_табличного\_пространства ◄◄

Схема - SYSPROC.

## Параметры процедуры

### варианты

Входной аргумент типа VARCHAR(32K), задающий используемые опции. Если никаких опций не требуется, в качестве аргумента можно указать пустое значение или пустую строку. Поддерживаемые значения:

#### **ADMINISTRATION TABLES IN**имя\_табличного\_пространства

Задаёт имя существующего обычного табличного пространства для управляющих таблиц, создаваемых при включении базы данных для DB2 Text Search. Если это условие не задано, в качестве табличного пространства используется SYSTOOLSPACE. Независимо от того, какое имя табличного пространства используется, SYSTOOLSPACE или непосредственно заданное, табличное пространство должно отвечать определённым требованиям. Это должно быть:

- Обычное табличное пространство
- Для многораздельных баз данных рекомендуется определить пул буферов и табличное пространство с размером страниц 32 Кбайт

Для использования регистрозависимого имени табличного пространства примените разделительный идентификатор, заключив имя в двойные кавычки. По умолчанию имя рассматривается как обычный идентификатор и преобразуется в верхний регистр.

#### *локаль\_сообщений*

Входной аргумент типа VARCHAR(33), задающий локаль, используемую для всех возвращаемых сообщений об ошибках. Если аргумент - пустое значение или пустая строка, или же на сервере недоступны файлы сообщений для указанной локали, используется 'en\_US'.

#### *сообщение*

Выходной аргумент типа VARCHAR(32K), задающий предупреждение или информационное сообщение для успешно завершённой операции.

## Примеры

*Пример 1:* Включение базы данных для текстового поиска при создании управляющих таблиц в табличном пространстве; все сообщения выводятся по-английски.

```
CALL SYSPROC.SYSTS_ENABLE ('ADMINISTRATION TABLES IN TSPACE', 'en_US', ?)
```

Включите базу данных для текстового поиска, создав управляющие таблицы в табличном пространстве с выводом всех сообщений по-русски.

```
CALL SYSPROC.SYSTS_ENABLE ('ADMINISTRATION TABLES IN "tbs32k" ', 'ru_RU', ?)
```

Как и в предыдущих выпусках, поддерживается процедура SYSTS\_ENABLE без аргумента **options**, например:

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

Пример вывода для этого запроса:

```
Значение выходных параметров
-----
Имя параметра      : MESSAGE
Значение параметра: Операция выполнена успешно.

Return Status = 0
```

*Пример 2:* В следующем примере SYSTS\_ENABLE вызывается для базы данных, для которой уже включен текстовый поиск. В результате вызывающему возвращается сообщение об ошибке.

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

Пример вывода для этого запроса:

```
SQL20427N Ошибка при выполнении процедуры или команды
управления текстовым поиском. Сообщение об ошибке продукта текстового поиска
"CIE00322 В указанной базе данных или базе данных по умолчанию уже разрешен
текстовый поиск. ". SQLSTATE 38N14
```

## Замечания по использованию

- В процедурах администрирования текстового поиска используется существующее соединение с базой данных. Рекомендуется перед выполнением этой процедуры управления текстовым поиском выполнить принятие всех изменений, внесенных транзакциями, чтобы избежать нежелательного влияния принятия или отката этой процедуры. Один из способов достигнуть этого - включить AUTOCOMMIT.
- При запуске этой процедуры происходят следующие события:
  - В схеме SYSIBMTS создаются объекты базы данных, такие как управляющие таблицы и производные таблицы каталога текстового поиска.
  - Задаваемые значения по умолчанию индексов текстового поиска для базы данных можно посмотреть в производной таблице SYSIBMTS.TSDEFAULTS.
  - При успешном выполнении этой команды создаются и становятся доступными таблицы и производные таблицы каталога текстового поиска.
- Если при выполнении этой процедуры у вас нет достаточных привилегий доступа к файлам и выполнения для получения конфигурации сервера Text Search, процедура создаст каталог текстового поиска и вернет предупреждение 'Неполное включение'. В таком случае информацию о соединении сервера Text Search нужно будет изменить вручную.

---

## Процедура SYSTS\_UPDATE - Обновление индекса текстового поиска

Процедура **SYSTS\_UPDATE** обновляет индекс текстового поиска, чтобы отразить текущее содержимое текстового столбца, с которым связан индекс.

Во время выполнения изменения возможен поиск. До завершения изменения поиск работает с частично обновленным индексом.

Эта процедура вызывает на сервере баз данных команду администрирования текстового поиска **UPDATE INDEX**.

## Авторизация

Привилегии ID авторизации этого оператора должны содержать роль SYSTS\_MGR и, как минимум, один из следующих типов полномочий:

- Полномочия DATAACCESS
- Привилегия CONTROL для таблицы, на которой определен текстовый индекс
- INDEX с привилегией SELECT для базовой таблицы, для которой определен индекс

Кроме этого, для первоначального обновления применяются требования авторизации, представленные в операторе **CREATE TRIGGER**.

## Привилегия PUBLIC по умолчанию

None (отсутствует)

## Синтаксис

```
►►—SYSTS_UPDATE—(—схема_индекса—, —имя_индекса—, —————→  
►—опции_изменения—, —локаль_сообщений—, —сообщение—)————→◄◄
```

Схема - SYSPROC.

## Параметры процедуры

### *схема\_индекса*

Входной аргумент типа VARCHAR(128), задающий схему индекса текстового поиска. Значение *схема\_индекса* должно соответствовать ограничениям именования для имен схем DB2. Если аргумент является пустым значением или пустой строкой, используется значение CURRENT SCHEMA. Аргумент *схема\_индекса* регистрозависим.

### *имя\_индекса*

Входной аргумент типа VARCHAR(128), задающий имя индекса. Вместе с аргументом *схема\_индекса* уникальным образом задает индекс текстового поиска в базе данных. Аргумент *имя\_индекса* регистрозависим.

### *опции\_изменения*

Входной аргумент типа VARCHAR(32K), задающий опции обновления. Если никаких опций не требуется, в качестве аргумента можно указать пустое значение или пустую строку. Возможные значения:

- USING UPDATE MINIMUM: Это значение соответствует параметру UPDATE MINIMUM управляющей команды текстового поиска **CREATE INDEX** и процедуры SYSTS\_CREATE.
- FOR DATA REDISTRIBUTION: Задает, что индекс текстового поиска в многораздельной базе данных нужно обновить после того, как добавлены или удалены разделы данных и завершена последующая операция перераспределения данных. Результаты поиска могут быть несогласованными, пока индекс текстового поиска не обновлен с опцией FOR DATA REDISTRIBUTION.
- ALLROWS: Задает, что первоначальное обновление должно быть выполнено безусловно.
- NULL или пустая строка (""): Задает, что обновление запускается безусловно, когда вызывается процедура

### локаль\_сообщений

Входной аргумент типа VARCHAR(33), задающий локаль, используемую для всех возвращаемых сообщений об ошибках. Если аргумент - пустое значение или пустая строка, или же на сервере недоступны файлы сообщений для указанной локали, используется 'en\_US'.

### сообщение

Выходной аргумент типа VARCHAR(32K), задающий предупреждение или информационное сообщение для успешно завершенной операции.

## Примеры

*Пример 1:* В следующем примере выполняется обновление индекса текстового поиска, созданного со значениями *схема\_индекса* 'db2ts' и *имя\_индекса* 'myTextIndex'. Пустое значение на месте параметра *опции\_обновления* означает, что при вызове этой хранимой процедуры обновление запускается безусловно. Указано, что сообщения об ошибках должны возвращаться по-русски. При успешном выполнении процедуры вызывающему возвращается сообщение выходного параметра.

```
CALL SYSPROC.SYSTS_UPDATE  
( 'db2ts', 'myTextIndex', '', 'ru_RU', ?)
```

Пример вывода этого запроса:

```
Значение выходных параметров  
-----  
Имя параметра   : MESSAGE  
Значение параметра: Операция выполнена успешно.
```

*Пример 2:* Обновление текстового индекса после выполнения операции для группы разделов, связанных с базовой таблицей; все сообщения об ошибках возвращаются по-английски.

```
CALL SYSPROC.SYSTS_UPDATE  
( 'db2ts', 'myTextIndex2', 'FOR DATA REDISTRIBUTION', 'en_US', ?)
```

```
Значение выходных параметров  
-----  
Имя параметра   : MESSAGE  
Значение параметра: Операция выполнена успешно.
```

Return Status = 0

*Пример 3:* В следующем примере SYSTS\_UPDATE вызывается для обновления индекса текстового поиска со значениями *схема\_индекса* 'db2ts' и *имя\_индекса* 'myTextIndex3'. Такой индекс не существует, и происходит ошибка.

```
CALL SYSPROC.SYSTS_UPDATE('db2ts', 'myTextIndex3', 'USING UPDATE MINIMUM',  
'ru_RU', ?)
```

```
SQL20427N Ошибка при выполнении процедуры или команды  
управления текстовым поиском. Сообщение об ошибке: "CIE00316 Индекс  
текстового поиска "db2ts"."myTextIndex3" не существует".  
SQLSTATE 38N14
```

## Замечания по использованию

- В процедурах администрирования текстового поиска используется существующее соединение с базой данных. В зависимости от завершения этих процедур для текущей транзакции выполняется принятие или откат. Поэтому вам может быть полезно выполнять принятие всех изменений транзакции во избежание неожиданных последствий такого принятия или отката. Один из способов принятия всех изменений транзакции - включение AUTOCOMMIT.

- Некоторые процедуры и команды нельзя выполнять для индекса текстового поиска одновременно, так как это может привести к ошибке из-за синхронизации конфликтующих операций. Некоторые из конфликтующих процедур и команд:
  - процедура SYSTS\_ALTER или команда **db2ts ALTER INDEX**
  - Процедура SYSTS\_CLEAR\_EVENTS или команда **db2ts CLEAR EVENTS FOR INDEX**
  - процедура SYSTS\_DISABLE или команда **db2ts DISABLE DATABASE FOR TEXT**
  - процедура SYSTS\_UPDATE или команда **db2ts UPDATE INDEX**

В случае конфликта процедура возвращает SQLCODE -20426 и SQLSTATE 38H13.
- Возврат из этой процедуры не происходит, пока не будет полностью завершена обработка обновления индекса. Продолжительность зависит от числа индексируемых документов и от числа уже проиндексированных документов. Имя собрания для индекса можно получить из столбца COLLECTIONNAME производной таблицы SYSIBMTS.TSCOLLECTIONNAMES.
- Если в отдельных документах есть ошибки, эти документы надо исправить. Первичные ключи документов с ошибками можно найти в таблице событий для индекса. Если вы измените соответствующие строки в пользовательской таблице, при следующем вызове SYSTS\_UPDATE эти документы обрабатываются повторно.
- При запуске процедуры SYSTS\_UPDATE происходят следующие события:
  - В таблицу событий вставляются строки, в том числе информация об ошибках синтаксического анализа. В случае инкрементного обновления удаляется информация из промежуточной таблицы индекса. Перед первым обновлением процедура SYSTS\_UPDATE создает триггеры для пользовательской таблицы.
  - Обновляется собрание:
    - Новые или измененные документы анализируются и индексируются.
    - Удаленные документы удаляются из индекса.
- Если с текстовым индексом был связан словарь синонимов, при выполнении обновления с опциями ALLROWS или FOR DATA REDISTRIBUTION, эта связь удаляется путем отбрасывания и повторного создания собраний для текстового индекса. Словарь синонимов нужно связать с новыми собраниями текстового индекса.

---

## Процедура SYSTS\_UPGRADE\_CATALOG - Обновление каталога текстового поиска

При этой процедуре до последней версии продукта обновляется каталог DB2 Text Search, в том числе управляющие таблицы и управляющие производные таблицы.

При этой процедуре создаются новые таблицы и производные таблицы каталога, которые используются в последней версии продукта, а также изменяются существующие таблицы и производные таблицы каталога. Устаревшие таблицы и производные таблицы каталога удаляются.

### Авторизация

Привилегии ID авторизации этой процедуры должны содержать роль SYSTS\_ADM и полномочия DBADM.

### Привилегия PUBLIC по умолчанию

None (отсутствует)

## Синтаксис

►—SYSTS\_UPGRADE\_CATALOG—(—локаль\_сообщений—,—сообщение—)—————►

Схема - SYSPROC.

## Параметры процедуры

### локаль\_сообщений

Входной аргумент типа VARCHAR(33), задающий локаль, используемую для всех возвращаемых сообщений об ошибках. Если аргумент - пустое значение или пустая строка, или же на сервере недоступны файлы сообщений для указанной локали, используется 'en\_US'.

### сообщение

Выходной аргумент типа VARCHAR(32K), задающий предупреждение или информационное сообщение для успешно завершенной операции.

## Пример

*Пример 1:* В следующем примере база данных была включена для текстового поиска в более старом выпуске. При вызове процедуры **SYSTS\_UPGRADE\_CATALOG** обновляются таблицы каталога текстового поиска и, если процедура выполняется успешно, вызывающему возвращается сообщение с обозначающим это параметром вывода.

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

Значение выходных параметров

-----

Имя параметра : MESSAGE

Значение параметра : CIE0213W Каталог DB2 Text Search был обновлен до текущей версии. Теперь вам может потребоваться обновить индекс текстового поиска при помощи хранимой процедуры SYSPROC.SYSTS\_UPGRADE\_INDEX.

Нужно обновить также индекс, вызвав "SYSPROC.SYSTS\_UPGRADE\_INDEX".

Return Status = 0

*Пример 2:* В следующем примере база данных не была включена для DB2 Text Search в более старом выпуске до вызова этой процедуры. Значение NULL для переменной *локаль\_сообщения* обозначает, что по умолчанию будет использоваться локаль 'en\_US'. Все сообщения об ошибках будут выводиться по-английски.

```
CALL SYSPROC.SYSTS_UPGRADE_CATALOG('',?)
```

SQL20427N Ошибка при выполнении процедуры или команды  
управления текстовым поиском.

Сообщение об ошибке:

"CIE0323E Заданная база данных или база данных по умолчанию  
не включена для текста".

SQLSTATE=38H14

*Пример 3:* В следующем примере каталог текстового поиска и версия индексов текстового поиска уже были обновлены до текущей версии.

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

Значение выходных параметров

-----

Имя параметра : MESSAGE

Значение параметра : CIE0002I Уровень выпуска DB2 Text Search -  
это текущий уровень для базы данных.

Система не обновлялась.

Return Status = 0

## Замечания по использованию

Процедура **SYSTS\_UPGRADE\_CATALOG** интегрирована в команду **DB2 UPGRADE DATABASE** для выполнения обновления конкретного каталога текстового поиска. Если при обновлении базы данных не удастся обновить каталог текстового поиска, пользователь должен вызвать процедуру **SYSTS\_UPGRADE\_CATALOG** отдельно, чтобы завершить обновление базы данных.

Для обновления каталога индекса текстового поиска выполните следующие действия:

1. Убедитесь, что служба экземпляра DB2 Text Search остановлена.
2. Вызовите процедуру **SYSTS\_UPGRADE\_CATALOG**.

**Примечание:** Управляющие процедуры DB2 Text Search используют существующее соединение с базой данных. В зависимости от завершения этих процедур для текущей транзакции выполняется принятие или откат. Поэтому вам может быть полезно выполнять принятие всех изменений транзакции во избежание неожиданных последствий такого принятия или отката. Один из способов добиться этого - включить **AUTOCOMMIT**.

3. Процедура **SYSTS\_UPGRADE\_CATALOG** пытается заполнить каталог информацией сервера Text Search. Проверьте содержимое **SYSIBMTS.TSSERVER** и при необходимости измените информацию сервера Text Search.
4. Обновите версии индексов текстового поиска, следуя действиям, описанным для процедуры **SYSTS\_UPGRADE\_INDEX**. Значение версии в управляющей таблице **SYSIBMTS.TSDEFAULTS** не будет изменено, пока не будет успешно выполнена процедура **SYSTS\_UPGRADE\_INDEX**.

---

## SYSTS\_UPGRADE\_INDEX - Обновление индексов текстового поиска

При этой процедуре изменяется информация индекса DB2 Text Search в таблицах каталога DB2 и в таблицах каталога текстового поиска.

Собрания индексов текстового поиска управляются сервером Text Search. Более подробную информацию смотрите в публикации Обновление DB2 Text Search.

### Авторизация

Привилегии ID авторизации этой процедуры должны содержать роль **SYSTS\_ADM** и полномочия **DBADM**.

### Привилегия PUBLIC по умолчанию

None (отсутствует)

### Синтаксис

►►—SYSTS\_UPGRADE\_INDEX—(—*локаль\_сообщений*—,—*сообщение*—)————►►

Схема - SYSPROC.

### Параметры процедуры

*локаль\_сообщений*

Входной аргумент типа VARCHAR(33), задающий локаль, используемую для

всех возвращаемых сообщений об ошибках. Если аргумент - пустое значение или пустая строка, или же на сервере недоступны файлы сообщений для указанной локали, используется 'en\_US'.

#### сообщение

Выходной аргумент типа VARCHAR(32K), задающий предупреждение или информационное сообщение для успешно завершенной операции.

## Пример

Пример 1: В следующем примере база данных была включена для текстового поиска в более старом выпуске, а процедура **SYSTS\_UPGRADE\_CATALOG** уже успешно завершена. Процедура **SYSTS\_UPGRADE\_INDEX** завершит обновление метаданных текстового индекса в каталоге базы данных. При успешном выполнении процедуры вызывающему возвращается сообщение выходного параметра.

```
CALL SYSPROC.SYSTS_UPGRADE_INDEX('en_US', ?)
Имя параметра      : MESSAGE
Значение параметра : CIE00001 Операция завершена успешно.
```

Состояние возврата = 0

Пример 2: В следующем примере база данных не была включена для текстового поиска в более старом выпуске продукта. Если для параметра *локаль\_сообщения* задано значение NULL, будет использована локаль системы. Если локаль системы недоступна, будет использована локаль по умолчанию 'en\_US'.

```
CALL SYSPROC.SYSTS_UPGRADE_INDEX(' ', ?)
SQL20427N Ошибка при выполнении процедуры или команды
          управления текстовым поиском. Сообщение об ошибке:
"CIE0323E Заданная база данных или база данных по умолчанию
не включена для текста".
SQLSTATE=38H14
```

Пример 3: В следующем примере каталог DB2 Text Search и версия индексов текстового поиска уже были обновлены до текущей версии.

```
CALL SYSPROC.SYSTS_UPGRADE_INDEX('en_US', ?)
Значение выходных параметров
-----
Имя параметра      : MESSAGE
Значение параметра : CIE0002I Уровень выпуска DB2 Text Search -
это текущий уровень для базы данных.
Система не обновлялась.
```

Return Status = 0

Пример 4: В следующем примере каталог DB2 Text Search не обновляется.

```
CALL SYSPROC.SYSTS_UPGRADE_INDEX('en_US', ?)
CIE0409E Каталог текстового поиска DB2 не обновлен до текущей версии.
```

## Замечания по использованию

- Процедура **SYSTS\_UPGRADE\_INDEX** интегрирована в команду **DB2 UPGRADE DATABASE** для выполнения обновления версии индексов текстового поиска. При запуске команды **DB2 UPGRADE DATABASE** будет выполнена и эта процедура. Если обновление версии индексов текстового поиска заканчивается неудачно, эту процедуру нужно запустить вручную.

- Процедура **SYSTS\_UPGRADE\_CATALOG** должна быть выполнена до процедуры **SYSTS\_UPGRADE\_INDEX**. Значение версии в управляющей таблице **SYSIBMTS.TSDEFAULTS** не будет изменено, пока не будет успешно выполнена процедура **SYSTS\_UPGRADE\_INDEX**.
- Управляющие процедуры DB2 Text Search используют существующее соединение с базой данных. В зависимости от завершения этих процедур для текущей транзакции выполняется принятие или откат. Поэтому вам может быть полезно выполнять принятие всех изменений транзакции во избежание неожиданных последствий такого принятия или отката. Один из способов добиться этого - включить **AUTOCOMMIT**.

---

## Глава 12. Управляющие производные таблицы текстового поиска

DB2 Text Search создает и обслуживает несколько управляющих производных таблиц, описывающих индексы текстового поиска в базе данных и их свойства.

Не обновляйте эти производные таблицы, если не было прямого указания сделать это.

Следующие производные таблицы отражают текущую конфигурацию вашей системы:

- Производные таблицы уровня базы данных:
  - SYSIBMTS.TSDEFAULTS
  - SYSIBMTS.TSLOCKS
  - SYSIBMTS.TSSERVERS
- Производные таблицы уровня индекса:
  - SYSIBMTS.TSINDEXES
  - SYSIBMTS.TSCONFIGURATION
  - SYSIBMTS.TSCOLLECTIONNAMES
  - SYSIBMTS.TSEVENT\_#####
  - SYSIBMTS.TSSTAGING\_#####

---

### Производная таблица SYSIBMTS.TSDEFAULTS

SYSIBMTS.TSDEFAULTS выводит все значения по умолчанию для всех индексов текстового поиска в базе данных.

Значения по умолчанию доступны в этом представлении как пары атрибут - значение.

Таблица 15. Производная таблица SYSIBMTS.TSDEFAULTS

Имя столбца	Тип данных	Допустимы пустые значения?	Описание
DEFAULTNAME	VARCHAR(30)	НЕТ	Параметры по умолчанию базы данных для текстового поиска
DEFAULTVALUE	VARCHAR(512)	НЕТ	Значения для параметров по умолчанию базы данных для текстового поиска

Следующие значения используются по умолчанию для команд db2ts **CREATE INDEX**, **ALTER INDEX**, **UPDATE INDEX** и **CLEAR EVENTS FOR INDEX**:

- AUXLOGNORM: Инфраструктура кэширования может быть включена для индекса текстового поиска путем явного задания параметра конфигурации индекса **AUXLOG ON**. Не включайте расширенную инфраструктуру кэширования с поддержкой текста для однораздельных таблиц по умолчанию.
- AUXLOGPART: Инфраструктура кэширования может быть отключена для индекса текстового поиска путем явного задания параметра конфигурации индекса **AUXLOG OFF**. По умолчанию расширенная инфраструктура кэширования с поддержкой текста включена для таблиц с разделением по диапазонам.

- **CJKSEGMENTATION**: Задает метод сегментации, используемый для индексирования документов на китайском, японском и корейском языке. Поддерживаемые значения: **MORPHOLOGICAL** и **NGRAM**. Значение по умолчанию - **NGRAM**.
- **CODEPAGE**: Начальное значение по умолчанию кодовой страницы для новых индексов - это кодовая страница базы данных.
- **DOCUMENTRESULTQUEUEUSE**: Это значение используется для ограничения объема памяти базы данных, резервируемой для каждой операции обновления для собрания. Значение по умолчанию - 30000; диапазон - от 100 до 100000. Заметим, что в многораздельной установке одно обновление текстового индекса, сконфигурированное для параллельного выполнения, зарезервирует память для каждого собрания, требующего обновления.
- **FORMAT**: Начальное значение по умолчанию для формата документа - простой текст.
- **LANGUAGE**: Начальное значение по умолчанию для индексации документов - en\_US.
- **MAXCONCURRENTUPDATES**: Управляет числом изменений собраний, которые могут выполняться параллельно в любой момент времени. Для многораздельных установок число собраний для каждого текстового индекса определяется согласно разделению таблицы. Однако при этом считаются только изменения активных разделов. Значение по умолчанию - 8.
- **MAXCONCURRENTCOLLECTIONS**: Управляет числом собраний, которые могут быть созданы. Для базы данных на одном узле число собраний равно числу текстовых индексов; для многораздельных установок число собраний на текстовый индекс соответствует разделению таблицы. Значение по умолчанию - 160.
- **MAXDOCUMENTSIZEINMB**: Управляет размером документов, приемлемых для обработки. Текст, размер которого превышает этот предел, вызовет запись сообщения предупреждения в таблицу событий. Значение равно 100.
- **UPDATEFREQUENCY**: Начальное значение по умолчанию для расписания обновления для новых индексов - **NONE**.
- **UPDATEMINIMUM**: Начальное значение по умолчанию для обновления для новых индексов - 1, то есть инкрементные обновления могут выполняться после каждого изменения.
- **UPDATEAUTOCOMMIT**: Начальное значение по умолчанию для обновления новых индексов равно 0, то есть промежуточные принятия при чтении документов из текстовых столбцов DB2 не выполняются. Это значение зарезервировано, и изменить его нельзя.

Командами **db2ts** нельзя изменить значения по умолчанию на уровне базы данных.

## Производная таблица **SYSIBMTS.TSLOCKS**

Можно просмотреть информацию блокировок команд на уровне базы данных и на уровне индекса, используя **SYSIBMTS.TSLOCKS**.

Таблица 16. Производная таблица **SYSIBMTS.TSLOCKS**

Имя столбца	Тип данных	Допустимы пустые значения?	Описание
COMMAND	VARCHAR(30)	НЕТ	Имя команды, создавшей блокировку. Возможные значения: CREATE INDEX, ALTER INDEX, DROP INDEX, UPDATE INDEX, CLEAR EVENTS, DISABLE DATABASE, CONFIGURE, CLEANUP

Таблица 16. Производная таблица SYSIBMTS.TSLOCKS (продолжение)

Имя столбца	Тип данных	Допустимы пустые значения?	Описание
LOCKSCOPE	VARCHAR(30)	НЕТ	Область действия блокировки. Возможные значения: DATABASE или INDEX.
INDSCHEMA	VARCHAR(128)	НЕТ	Имя схемы индекса текстового поиска (только при LOCKSCOPE = INDEX)
INDNAME	VARCHAR(128)	НЕТ	Неспецифицированное имя индекса текстового поиска (только при LOCKSCOPE = INDEX)
PARTITION	INTEGER	НЕТ	Номер раздела, на котором создана блокировка текстового поиска.
LOCKCREATETIME	TIMESTAMP	НЕТ	Отметка времени, когда была предоставлена блокировка

Есть три различных сценария, которые следует иметь в виду в связи со стратегиями блокировок:

- **Операция запущена и не встретила применимых блокировок:** Процедура устанавливает блокировку и продолжает выполнение. И при успешном, и при неудачном выполнении эта блокировка снимается.
- **Операция запущена и встретила применимую блокировку:** Требование возвращается с сообщением о конфликтующей команде.
- Операция запущена и встретила применимую блокировку, несмотря на то, что связанные программы сейчас не выполняются: Ошибка предыдущей операции не позволила правильным образом снять блокировку. Это могло случиться в особых ситуациях, таких как отказы диска или аварии. В таком случае блокировку надо снять при помощи операции **CLEAR COMMAND LOCKS** уровня индекса или базы данных, как требуется, после устранения причины ошибки и проверки согласованности системы.

## Производная таблица SYSIBMTS.TSSERVERS

В каждой строке, выведенной в производной таблице SYSIBMTS.TSSERVERS, показывается информация о сервере DB2 Text Search, сконфигурированном для базы данных.

Можно запросить производную таблицу для получения информации о помеченном для использования сервере текстового поиска:

```
db2 "SELECT SERVERID, HOST from SYSIBMTS.TSSERVERS where SERVERSTATUS = 0"
```

Таблица 17. Производная таблица SYSIBMTS.TSSERVERS

Имя столбца	Тип данных	Допустимы пустые значения?	Описание
SERVERID	INTEGER	НЕТ	Уникальный ID, сгенерированный для сервера текстового поиска.
HOST	VARCHAR(256)	НЕТ	Имя хоста или IP-адрес сервера текстового поиска. Для многораздельных баз данных, внедрений автономных серверов текстового поиска и в случае выполнения операций управления с удаленных клиентов убедитесь, что используется фактическое имя хоста или IP-адрес, а не 'localhost'.

Таблица 17. Производная таблица SYSIBMTS.TSSERVERS (продолжение)

Имя столбца	Тип данных	Допустимы пустые значения?	Описание
PORT	INTEGER	НЕТ	Имя порта для сервера текстового поиска. (ADMIN/SEARCH)
TOKEN	VARCHAR(256)	НЕТ	Маркер аутентификации для сервера текстового поиска.
KEY	VARCHAR(128)	НЕТ	Ключ для сервера текстового поиска.
DEFAULTLOCALE	VARCHAR(33)	НЕТ	Локаль клиента по умолчанию для сообщений с сервера текстового поиска
SERVERTYPE	INTEGER	НЕТ	Это значение указывает на тип для каждого сервера текстового поиска. <ul style="list-style-type: none"> <li>• 0 = интегрированный сервер текстового поиска (integrated text search server), значение по умолчанию</li> <li>• отличное от нуля значение = автономный сервер текстового поиска <ul style="list-style-type: none"> <li>– 1 = автономный локальный сервер текстового поиска</li> <li>– 2 = автономный удаленный сервер текстового поиска</li> </ul> </li> </ul>
SERVERSTATUS	INTEGER	НЕТ	Указывает, может ли сервер текстового поиска использоваться для создания новых индексов текстового поиска. Значение по умолчанию 0 указывает, что сервер активен и используется.

## Производная таблица SYSIBMTS.TSINDEXES

Свойства текущего индекса текстового поиска выводятся в производной таблице SYSIBMTS.TSINDEXES.

В следующем примере используется схема и имя индекса:

```
db2 "SELECT COLNAME from SYSIBMTS.TSINDEXES where INDSHEMA=имя-схемы
and INDNAME=имя-индекса"
```

Производная таблица SYSIBMTS.TSINDEXES описана в следующей таблице.

Таблица 18. Производная таблица SYSIBMTS.TSINDEXES

Имя столбца	Тип данных	Допустимы пустые значения?	Описание
INDSCHEMA	VARCHAR (128)	НЕТ	Имя схемы для индекса текстового поиска.
INDNAME	VARCHAR (128)	НЕТ	Неспецифицированное имя индекса текстового поиска.
TABSCHEMA	VARCHAR (128)	НЕТ	Имя схемы базовой таблицы.
TABNAME	VARCHAR (128)	НЕТ	Неспецифицированное имя базовой таблицы.
COLNAME	VARCHAR (128)	НЕТ	Столбец, для которого был создан индекс текстового поиска.
CODEPAGE	INTEGER	НЕТ	Кодовая страница документа для индекса текстового поиска.
LANGUAGE	VARCHAR(5)	НЕТ	Язык документа для индекса текстового поиска.

Таблица 18. Производная таблица SYSIBMTS.TSINDEXES (продолжение)

Имя столбца	Тип данных	Допустимы пустые значения?	Описание
FORMAT	VARCHAR(30)	ДА	Формат документа.
FUNCTIONSCHEMA	VARCHAR (128)	ДА	Схема для типа столбца.
FUNCTIONNAME	VARCHAR(18)	ДА	Имя функции преобразования типа столбца.
COLLECTIONDIRECTORY	VARCHAR(512)	ДА	Каталог для файлов индекса текстового поиска.
UPDATEFREQUENCY	VARCHAR(300)	НЕТ	Критерий триггера для применения обновлений к индексу.
UPDATEMINIMUM	INTEGER	ДА	Минимальное число записей в таблице журнала перед инкрементным обновлением. Меньшее значение означает лучшее соответствие между столбцом таблицы и индексом текстового поиска. Однако при меньшем значении возрастает также объем ресурсов, необходимых для индексирования текстового поиска. value also increases the resources required for text search indexing.
EVENTVIEWSCHEMA	VARCHAR (128)	НЕТ	Схема для производной таблицы событий, созданной для индекса текстового поиска (всегда SYSIBMTS).
EVENTVIEWNAME	VARCHAR (128)	НЕТ	Имя производной таблицы событий, созданной для индекса текстового поиска.
STAGINGVIEWSCHEMA	VARCHAR (128)	ДА	Схема для производной таблицы журналов, созданной для индекса текстового поиска (всегда SYSIBMTS).
STAGINGVIEWNAME	VARCHAR (128)	ДА	Имя производной таблицы журналов, созданной для индекса текстового поиска.
REORGAUTOMATIC	INTEGER	ДА	Зарезервировано (не поддерживается в этом выпуске). Значение всегда равно 1.
RECREATEONUPDATE	INTEGER	НЕТ	Зарезервировано (не поддерживается в этом выпуске). Значение всегда равно 0.
ATTRIBUTES	VARCHAR(32000)	ДА	Зарезервировано (не поддерживается в этом выпуске).
COLLECTIONNAMEPREFIX	VARCHAR (128)	НЕТ	Префикс имени собрания на сервере текстового поиска.
COMMENT	VARCHAR(512)	ДА	Комментарий, заданный для параметра, который относится к свойствам индекса команды <b>CREATE INDEX</b> .

Таблица 18. Производная таблица SYSIBMTS.TSINDEXES (продолжение)

Имя столбца	Тип данных	Допустимы пустые значения?	Описание
AUXSTAGINGSHEMA	VARCHAR(48)	ДА	Схема промежуточной текстовой таблицы.
AUXSTAGINGNAME	VARCHAR(48)	ДА	Имя промежуточной текстовой таблицы.
INDSTATUS	VARCHAR(10)	НЕТ	Состояние индекса: <ul style="list-style-type: none"> <li>• ACTIVE обозначает активный индекс.</li> <li>• INACTIVE обозначает неактивный индекс. (Это значение не используется для DB2 Text Search).</li> <li>• INVALID обозначает недопустимый индекс, обычно это побочный эффект операции DB2.</li> </ul>

## Производная таблица SYSIBMTS.TSCONFIGURATION

Информация о параметрах конфигурации индекса доступна в производной таблице SYSIBMTS.TSCONFIGURATION.

Каждая строка представляет параметр индекса текстового поиска.

Следующий пример запроса производной таблицы использует имя индекса:

```
db2 "SELECT VALUE from SYSIBMTS.TSCONFIGURATION where INDSHEMA=имя-схемы
and INDNAME=имя-индекса and PARAMETER ='параметр'"
```

Таблица 19. Производная таблица SYSIBMTS.TSCONFIGURATION

Имя столбца	Тип данных	Допустимы пустые значения?	Описание
INDSCHEMA	VARCHAR(128)	НЕТ	Имя схемы индекса текстового поиска
INDNAME	VARCHAR(128)	НЕТ	Неспецифицированное имя индекса текстового поиска
PARAMETER	VARCHAR(30)	НЕТ	Имя параметра конфигурации
VALUE	VARCHAR(512)	НЕТ	Значение параметра

Столбец PARAMETER содержит имена параметров конфигурации индекса текстового поиска, заданные при помощи оператора CREATE INDEX, и имена некоторых параметров из производной таблицы SYSIBMTS.TSDEFAULTS.

---

## SYSIBMTS.TSCOLLECTIONNAMES view

Производная таблица SYSIBMTS.TSCOLLECTIONNAMES содержит имена собраний.

Каждая строка представляет собрание для индекса текстового поиска.

Таблица 20. SYSIBMTS.TSCOLLECTIONNAMES view

Имя столбца	Тип данных	Допустимы пустые значения?	Описание
INDSCHEMA	VARCHAR(128)	НЕТ	Имя схемы индекса текстового поиска
INDNAME	VARCHAR(128)	НЕТ	Неспецифицированное имя индекса текстового поиска
COLLECTIONNAME	VARCHAR(132)	НЕТ	Имя связанного собрания на сервере текстового поиска. В системах многораздельных баз данных каждый раздел текстового индекса представляется как собрание. Имя собрания включает номер этого раздела в качестве суффикса.

---

## Производная таблица SYSIBMTS.TSEVENT

Производная таблица содержит информацию о состоянии индексации и о событиях ошибок.

В базе данных может быть несколько производных таблиц с префиксом SYSIBMTS.TSEVENT. У каждой производной таблицы свое значение *nnnnnn*; это внутренний идентификатор, указывающий на соответствующий текстовый индекс, с которым связана производная таблица. Чтобы выяснить, какой индекс текстового поиска связан с конкретной производной таблицей, запросите производную таблицу SYSIBMTS.TSINDEXES, задав поиск имени схемы и имени производной таблицы в столбцах EVENTVIEWSHEMA и EVENTVIEWNAME. Этот запрос возвращает одну строку, в которой описываются нужные индекс текстового поиска и пользовательская таблица.

Число столбцов в этом представлении зависит от числа столбцов первичного ключа в пользовательской таблице. Столбцы PK1..PK*nn* соответствуют столбцам первичного ключа пользовательской таблицы; у них соответствующие типы данных и определения длин. Тип данных в каждом из столбцов производной таблицы точно соответствует типу данных соответствующего столбца первичного ключа.

Каждая строка в этой производной таблице представляет сообщение команды **UPDATE INDEX** для индекса текстового поиска. Например, в строке может быть указано, что команда **UPDATE INDEX** была запущена или завершилась. Строка может также описывать ошибку, которая возникла при индексации текстового документа. Этот текстовый документ можно идентифицировать, получив значения из столбцов первичного ключа в строке этой производной таблицы и найдя их в пользовательской таблице.

Можно очистить события командой db2ts **CLEAR EVENTS FOR INDEX**.

Таблица 21. Производная таблица событий

Имя столбца	Тип данных	Допустимы пустые значения?	Описание
OPERATION	INTEGER	ДА	Операция (вставка, изменение или удаление) для базовой таблицы, которая отражается в индексе текстового поиска
TIME	TIMESTAMP	ДА	Отметка времени создания записи о событии
SEVERITY	INTEGER	ДА	Если сообщение относится к одному документу, здесь будет одно из следующих значений: <ul style="list-style-type: none"> <li>• 1 = Информационное</li> <li>• 4 = Части документа были проиндексированы, но с предупреждением, как указано в сообщении</li> <li>• 8 = Документ не был проиндексирован, как указано в сообщении</li> <li>• 0 = Иное</li> </ul>
SQLCODE	INTEGER	ДА	Код SQL соответствующей ошибки, если есть
MESSAGE	VARCHAR(1024)	ДА	Текстовая информация о конкретной ошибке
PARTITION	INTEGER	ДА	Зарезервировано для внутреннего использования IBM.
PK01	Тип данных первого столбца первичного ключа базовой таблицы	ДА	Значение первого столбца первичного ключа базовой таблицы индекса текстового поиска для строки, которая обрабатывалась, когда возникла ошибка
...	...	...	...
PK $m$	Тип данных последнего столбца первичного ключа базовой таблицы	ДА	Значение последнего столбца первичного ключа базовой таблицы индекса текстового поиска для строки, которая обрабатывалась, когда возникла ошибка

Кроме того, в этой производной таблице доступны информационные события, такие как запуск, принятие и завершение обработки изменений. В этом случае значения PK01, PK $m$  и OPERATION - пустые. Кодовая страница и локаль MESSAGE соответствуют параметрам базы данных.

## Производная таблица SYSIBMTS.TSSTAGING

Промежуточная таблица хранит операции изменения пользовательской таблицы, для которой требуется синхронизация с индексом текстового поиска.

Триггеры создаются для пользовательской таблицы, чтобы передавать информацию об изменениях в промежуточную таблицу. Кроме того, с дополнительными опциями записи в журнал обработка целостности обнаруживает изменения в пользовательской таблице. Команда **UPDATE INDEX** читает эти записи и, после успешной синхронизации, удаляет их.

В базе данных может быть несколько производных таблиц с префиксом SYSIBMTS.TSSTAGING\_. У каждой производной таблицы свое значение  $nnnnnn$ ; это внутренний идентификатор, указывающий на соответствующий текстовый индекс, с которым связана производная таблица. Чтобы выяснить, какой индекс текстового поиска связан с конкретной производной таблицей, запросите производную таблицу SYSIBMTS.TSINDEXES, задав поиск имени схемы и имени производной таблицы в столбцах STAGINGVIEWSCHEMA и STAGINGVIEWNAME. Запрос возвращает одну строку, в которой описываются нужные индекс текстового поиска и пользовательская таблица.

Число столбцов в этом представлении зависит от числа столбцов первичного ключа в пользовательской таблице. Столбцы PK1..PK*n* соответствуют столбцам первичного ключа пользовательской таблицы; у них соответствующие типы данных и определения длин. Тип данных в каждом из столбцов производной таблицы точно соответствует типу данных соответствующего столбца первичного ключа.

Каждая строка в этой производной таблице представляет операцию вставки, удаления или изменения для строки пользовательской таблицы или текстового документа. Этот текстовый документ можно идентифицировать, получив значения из столбцов первичного ключа в строке этой производной таблицы и найдя их в пользовательской таблице.

Для получения информации о производной таблице можно использовать следующий запрос:

```
db2 "SELECT STAGINGVIEWSHEMA, STAGINGVIEWNAME from SYSIBMTS.TSINDEXES
where INDSHEMA=имя-схемы and INDNAME=имя-индекса"
```

Таблица 22. Производная таблица SYSIBMTS.TSSTAGING

Имя столбца	Тип данных	Допустимы пустые значения?	Описание
OPERATION	INTEGER	НЕТ	Операция (вставка, изменение или удаление) для базовой таблицы, которая отражается в индексе текстового поиска
TIME	TIMESTAMP	НЕТ	Порядковый ID строки (когда сработал триггер вставки, изменения или удаления). Это отметка времени, которая, однако, не задает точного времени операции.
STATUS	INTEGER	НЕТ	Состояние обработки для строки: -1 не обработано
PK01	Тип данных первого столбца первичного ключа базовой таблицы	ДА	Первый столбец первичного ключа базовой таблицы.
...	...	...	...
PK <i>n</i>	Тип данных последнего столбца первичного ключа базовой таблицы	ДА	Последний столбец первичного ключа базовой таблицы.



## Приложение А. Сравнение DB2 Text Search и Net Search Extender

Прежде чем перенастроиться с Net Search Extender (NSE) на DB2 Text Search, необходимо принять во внимание различия в синтаксисе, семантике и наборах результатов для запросов полнотекстового поиска, которые выглядят аналогичными в этих двух решениях.

Прочитайте Табл. 23 and Табл. 24 на стр. 220, чтобы узнать, возможен ли для вас перенос из NSE в DB2 Text Search.

DB2 Text Search поддерживается в тех же операционных системах, где поддерживался Net Search Extender, кроме 64-битных операционных систем Linux в System z. В следующей таблице содержится список функций установки, доступных в NSE и DB2 Text Search:

Таблица 23. Функции установки, доступные в NSE и DB2 Text Search

Функция	NSE	Текстовый поиск DB2 (DB2 Text Search)	Комментарии и ссылки на дополнительную информацию
Локальная установка для текстового механизма	Да	Да	
Удаленная установка для текстового механизма	Нет	Да	“Сценарии внедрения сервера DB2 Text Search” на странице <a href="http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0058598.html">http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0058598.html</a>
Разделение базы данных	Да	Да	“DB2 Text Search в среде многораздельных баз данных” на странице <a href="http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0058524.html">http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0058524.html</a>
Индекс для однораздельных базовых таблиц	Да	Да	Создание, обновление и изменение свойств индексов текстового поиска на странице <a href="http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c_textindexcreation.html">http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c_textindexcreation.html</a>

Таблица 23. Функции установки, доступные в NSE и DB2 Text Search (продолжение)

Функция	NSE	Текстовый поиск DB2 (DB2 Text Search)	Комментарии и ссылки на дополнительную информацию
Индекс для многораздельных базовых таблиц (с разделением по диапазонам)	Да	Да	Расширенная инфраструктура промежуточного обслуживания текста для инкрементных обновлений индекса текстового поиска на странице <a href="http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0057426.html">http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0057426.html</a>
Индекс для псевдонимов (с репликацией)	Объявлены устаревшими	Нет	Объявленные устаревшими в Версии 9.7
Индекс для производных таблиц	Да	Нет	

DB2 Text Search предоставляет аналогичные Net Search Extender функциональные возможности. В следующей таблице показаны функциональные возможности, доступные в NSE и DB2 Text Search:

Таблица 24. Функциональные возможности, доступные в NSE и DB2 Text Search

Функциональные элементы	NSE	DB2 Text Search	Комментарии и ссылки на дополнительную информацию
Пересоздание при изменении	Да	Да	
Пользовательские функции преобразований	Да	Да	
Кэширование	Нет	Нет	
Множественные индексы	Да	Нет	
Предварительно сортированные индексы	Нет	Нет	
Словарь синонимов	Да	Да	“Словари синонимов для DB2 Text Search” на странице <a href="http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0052652.html">http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0052652.html</a>
Тезаурус (ассоциативный, иерархический, определяемый пользователем)	Нет	Нет	
Текст, HTML, XML	Да	Да	“Форматы документов, поддерживаемые для DB2 Text Search”, на странице <a href="http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/r0053096.html">http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/r0053096.html</a>

Таблица 24. Функциональные возможности, доступные в NSE и DB2 Text Search (продолжение)

Функциональные элементы	NSE	DB2 Text Search	Комментарии и ссылки на дополнительную информацию
INSO	Да	Да	DB2 Text Search поддерживает INSO с использованием пакета дополнительных компонентов DB2. Подробности смотрите в разделе “Поддержка документов Rich Text Format (RTF)” на странице <a href="http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0054766.html">http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0054766.html</a> .
GPP	Да	Нет	Вы можете создать функцию в DB2 Text Search для поддержки GPP
Документные модели	Да	Нет	
Лингвистическая обработка	Да	Да <sup>+</sup>	Лингвистический процесс NSE ограничен простым поиском морфологических форм (только в английском).  DB2 Text Search поддерживает лингвистическую обработку для 20 языков, причем для таких языков, как китайский, японский и корейский, поддерживается как морфологическая сегментация, так и сегментация энграмм. Подробности смотрите в разделе “Лингвистическая обработка для DB2 Text Search”.
функция CONTAINS	Да	Да	“Функция CONTAINS на странице <a href="http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/r_contains.html">http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/r_contains.html</a> ”
функция SCORE	Да	Да	DB2 Text Search использует другой алгоритм, который может выдавать другие результаты. Подробности смотрите в разделе “Функция SCORE на странице <a href="http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/r_score.html">http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/r_score.html</a> ”.
Число соответствий	Нет	Нет	
Выделения	Нет	Нет	

Таблица 24. Функциональные возможности, доступные в NSE и DB2 Text Search (продолжение)

Функциональные элементы	NSE	DB2 Text Search	Комментарии и ссылки на дополнительную информацию
Обработка стоп-слов	Да	Да	“Синтаксис инструмента стоп-слов для DB2 Text Search на странице <a href="http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/r0058492.html">http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/r0058492.html</a> ”
Предельное число результатов	Да	Да	Для функций CONTAINS и SCORE предусмотрен параметр <b>RESULTLIMIT</b> , который указывает максимальное число возвращаемых результатов.
Нормализация символов	Да	Да	
Escape-символы	Да	Да	Настройка недоступна в DB2 Text Search.
Поиск с логическими условиями	Да	Да	“Синтаксис аргументов текстового поиска на странице <a href="http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/r0052651.html">http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/r0052651.html</a> ”
Символы подстановки	Да	Да	“Синтаксис аргументов текстового поиска на странице <a href="http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/r0052651.html">http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/r0052651.html</a> ”
Поиск морфологических форм	Да	Да	Поиск морфологических форм включен по умолчанию для DB2 Text Search
Точный поиск	Да	Да	Текстовый поиск DB2 не зависит от регистра. Подробности смотрите в разделе “Точный поиск на странице <a href="http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/t_searchingwiththetextindex.html">http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/t_searchingwiththetextindex.html</a> ”.
Поиск неполных соответствий	Да	Да	“Нечеткий поиск на странице <a href="http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0058557.html">http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0058557.html</a> ”
Контекстный поиск	Да	Да	“Поиск с заданием близости слов на странице <a href="http://www.ibm.com/support/.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0058673.html">http://www.ibm.com/support/.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0058673.html</a> ”

Таблица 24. Функциональные возможности, доступные в NSE и DB2 Text Search (продолжение)

Функциональные элементы	NSE	DB2 Text Search	Комментарии и ссылки на дополнительную информацию
Поиск с использованием диапазонов	Да	Да, для XML	DB2 Text Search поддерживает поиск с использованием диапазонов на основе выражений XPath в XML. Net Search Extender поддерживает поиск с использованием диапазонов при помощи документной модели.
Поиск свободного текста	Да	Нет	
Поиск по полю	Да	Да, для XML	Поддержка DB2 Text Search использует выражения XPath в XML. Поддержка NSE использует документную модель. Подробности смотрите в разделе “Конфигурирование поиска XML для DB2 Text Search на странице <a href="http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0052709.html">http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0052709.html</a> ” и “Поиск в документах XML с использованием DB2 Text Search на странице <a href="http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0052708.html">http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/c0052708.html</a> ”.
Поиск атрибутов	Да	Нет	
Вес/повышение значимости	Да	Да	У DB2 Text Search и NSE разные алгоритмы. Подробности смотрите в разделе “Поиск в индексах тестового поиска при помощи SCORE на странице <a href="http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/t_searchingandreturningscore.html">http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ts.doc/doc/t_searchingandreturningscore.html</a> ”.



---

## Приложение В. Локали, поддерживаемые DB2 Text Search

В следующей таблице перечислены локали, поддерживаемые DB2 Text Search для обработки документов.

Таблица 25. Поддерживаемые локали

Код локали	Язык	Регион
ar_AA	Арабский	Арабские страны или регионы
cs_CZ	Чешский	Чехия
da_DK	Датский	Дания
de_CH	Немецкий	Швейцария
de_DE	Немецкий	Германия
el_GR	Греческий	Греция
en_AU	Английский	Австралия
en_GB	Английский	Великобритания
en_US	Английский	Соединенные Штаты
es_ES	Испанский	Испания
fi_FI	Финский	Финляндия
fr_CA	Французский	Канада
fr_FR	Французский	Франция
it_IT	Итальянский	Италия
ja_JP	Японский	Япония
ko_KR	корейский	Республика Корея
nb_NO	Норвежский (букмол)	Норвегия
nl_NL	Голландский	Нидерланды
nn_NO	Норвежский (нюнорск)	Норвегия
pl_PL	Польский	Польша
pt_BR	Португальский	Бразилия
pt_PT	Португальский	Португалия
ru_RU	Русский	Россия
sv_SE	Шведский	Швеция
zh_CN	китайский	Китай
zh_TW	китайский	Тайвань



---

## Приложение С. Команды DB2

---

### db2iupgrade - обновление экземпляра

Обновляет экземпляр до копии DB2 текущего выпуска от копии DB2 предыдущего выпуска. Копия DB2, из которой вы запускаете команду **db2iupgrade**, должна поддерживать обновление экземпляра от копии DB2, которую вы хотите обновить.

В операционных системах Linux и UNIX эта команда расположена в каталоге *DB2DIR/instance*, где *DB2DIR* - это положение установки, где установлен новый выпуск системы баз данных DB2. Эта команда не поддерживает обновления экземпляра для установки без полномочий root.

В операционных системах Windows эта команда расположена в каталоге **DB2PATH\bin**, где **DB2PATH** - это положение, где установлена копия DB2. Чтобы переместить профиль вашего экземпляра из его текущего положения в другое, используйте опцию **/p** и задайте путь к профилю экземпляра. В противном случае профиль экземпляра останется после обновления в своем исходном положении.

#### Авторизация

Полномочия пользователя root в операционных системах Linux и UNIX или локального администратора в операционных системах Windows.

#### Синтаксис команды

Для операционных систем Linux и UNIX

```
db2iupgrade [-d] [-k]
             [-j "TEXT_SEARCH" [, имя_службы] [, номер_порта]]
             [-a тип_аутентификации] [-u ID_изолированного_пользователя]
             имя_экземпляра
```

Для операционных систем Windows

```
db2iupgrade имя_экземпляра /u: имя-пользователя,пароль
             /p: путь-профиля-экземпляра /q /a: authType
             /j "TEXT_SEARCH" [, имя_службы] [, номер_порта] [/?]
```

#### Параметры команды

Для операционных систем Linux и UNIX

- d Включает режим отладки. Используйте эту опцию только после получения инструкций от службы поддержки баз данных DB2.
- k Сохраняет прежний тип экземпляра, если он поддерживается в копии DB2, из которой вы запускаете команду **db2iupgrade**. Если этот параметр не задан, тип экземпляра изменяется на поддерживаемый тип экземпляра по умолчанию.

**-j "TEXT\_SEARCH"**

Конфигурирует сервер DB2 Text Search с использованием сгенерированных значений по умолчанию для имени службы и номера порта TCP/IP. Этот параметр нельзя использовать, если тип экземпляра - клиент.

**-j "TEXT\_SEARCH,имя\_службы"**

Конфигурирует сервер DB2 Text Search с использованием заданного имени службы и автоматически сгенерированного номера порта. Если для заданного имени службы в файле `services` задан номер порта, используется этот номер порта.

**-j "TEXT\_SEARCH,имя\_службы,номер\_порта"**

Конфигурирует сервер DB2 Text Search с использованием заданного имени службы и номера порта.

**-j "TEXT\_SEARCH,номер\_порта"**

Конфигурирует сервер DB2 Text Search с использованием имени службы по умолчанию и заданного номера порта. Допустимые номера порта должны быть в диапазоне 1024 - 65535.

**-a тип\_аутентификации**

Задает тип аутентификации (SERVER, CLIENT или SERVER\_ENCRYPT) для этого экземпляра. Значение по умолчанию - SERVER.

**-u ID\_изолированного\_пользователя**

Задает имя ID пользователя, применяемое для запуска изолированных пользовательских функций и изолированных хранимых процедур. Эта опция требуется при обновлении экземпляра клиента DB2 до экземпляра сервера DB2.

*InstName*

Задает имя экземпляра.

**Для операционных систем Windows**

*InstName*

Задает имя экземпляра.

**/u:имя-пользователя,пароль**

Задает имя учетной записи и пароль для службы DB2. Эта опция обязательна при обновлении многораздельного экземпляра.

**/p:путь-профиля-экземпляра**

Задает новый путь профиля экземпляра для обновляемого экземпляра.

**/q** Запускает команду **db2iupgrade** в режиме без вывода сообщений.

**/a:authType**

Задает тип аутентификации (SERVER, CLIENT или SERVER\_ENCRYPT) для этого экземпляра.

### **/j "TEXT\_SEARCH"**

Конфигурирует сервер DB2 Text Search с использованием сгенерированных значений по умолчанию для имени службы и номера порта TCP/IP. Этот параметр нельзя использовать, если тип экземпляра - клиент.

### **/j "TEXT\_SEARCH, имя\_службы"**

Конфигурирует сервер DB2 Text Search с использованием заданного имени службы и автоматически сгенерированного номера порта. Если для заданного имени службы в файле `services` задан номер порта, используется этот номер порта.

### **/j "TEXT\_SEARCH, имя\_службы, номер\_порта"**

Конфигурирует сервер DB2 Text Search с использованием заданного имени службы и номера порта.

### **/j "TEXT\_SEARCH, номер\_порта"**

Конфигурирует сервер DB2 Text Search с использованием имени службы по умолчанию и заданного номера порта. Допустимые номера порта должны быть в диапазоне 1024 - 65535.

**/?** Выводит информацию об использовании для команды **db2iupgrade**.

## **Замечания по использованию**

При помощи команды **db2iupgrade** можно обновлять только экземпляры DB2 Enterprise Server Edition (тип экземпляра - *ese*).

Команда **db2iupgrade** вызывает команду **db2ckupgrade** с параметром **-not1** и задает файл `upgrade.log` как файл журнала для **db2ckupgrade**. Создаваемый для **db2iupgrade** файл журнала по умолчанию - это `/tmp/db2ckupgrade.log.processID`. Перед обновлением экземпляра проверьте, готовы ли локальные базы данных для обновления. Параметр **-not1** отключает проверку наличия индексов типа 1. Файл журнала создается в домашнем каталоге экземпляра для операционных систем Linux и UNIX или в текущем каталоге для операционных систем Windows. Если команда **db2ckupgrade** вернет какую-либо ошибку, обновление экземпляра будет прервано.

Для сред многораздельных баз данных до запуска команды **db2iupgrade** введите команду **db2ckupgrade**. Команда **db2ckupgrade** проверит все разделы и возвратит ошибки, найденные в этих разделах. Если вы не проверите готовность всех разделов базы данных для обновления, последующие обновления баз данных могут окончиться неудачно даже при успешном обновлении экземпляра. Подробности смотрите в описании команды **db2ckupgrade**.

### **Для операционных систем Linux и UNIX**

- Если вы используете команду **db2iupgrade** для обновления экземпляра DB2 от предыдущей версии до текущей версии системы баз данных DB2, глобальные переменные профиля DB2, определенные в старом каталоге установки базы данных DB2, не будут обновлены в новое положение установки. Переменные профиля экземпляра DB2, специфичные для экземпляра, который будет обновляться, будут перенесены в нужное положение после завершения обновления экземпляра.

- Если для получения полномочий root вы используете команду **su** вместо команды **login**, надо ввести команду **su** с опцией **-**, которая указывает задание такой же среды процесса, что и при регистрации в системе при помощи команды **login**.
- Не используйте среду экземпляра DB2 как источник для пользователя root. Если среда экземпляра DB2 не поддерживается в качестве источника, запустите команду **db2iupgrade**.
- В AIX 6.1 (или в более новых версиях), когда эта команда выполняется из копии DB2 совместного использования в глобальной среде системных разделов рабочей нагрузки (workload partition, WPAR), эта команда должна быть запущена от имени пользователя root. WPAR не поддерживается в среде DB2 pureScale.

## db2icrt - Создание экземпляра

Создайте экземпляр DB2, включая экземпляр DB2 pureScale. С помощью этой команды можно также создать начальный DB2 элемент и механизм кэширования кластера как часть создания экземпляра DB2 pureScale.

В операционных системах Linux и UNIX **db2icrt** находится в каталоге *DB2DIR/instance*, где *DB2DIR* представляет каталог установки, в котором установлена система баз данных DB2. В операционных системах Windows **db2icrt** находится в каталоге *DB2PATH\bin*, где *DB2PATH* - каталог, в котором установлена копия DB2.

Команда **db2icrt** создает экземпляр DB2 в домашнем каталоге владельца экземпляра. Можно создать только один экземпляр DB2 pureScale на среду DB2 pureScale.

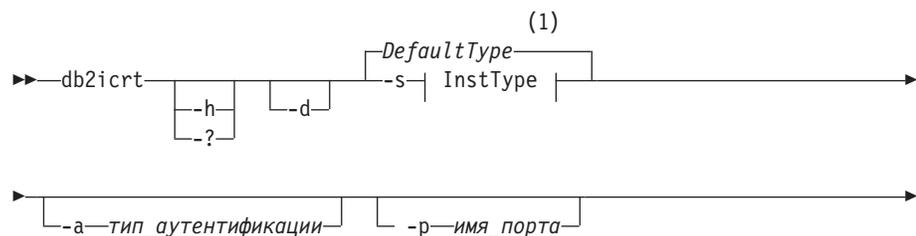
**Примечание:** Эта команда недоступна для установки продуктов баз данных DB2 без полномочий root.

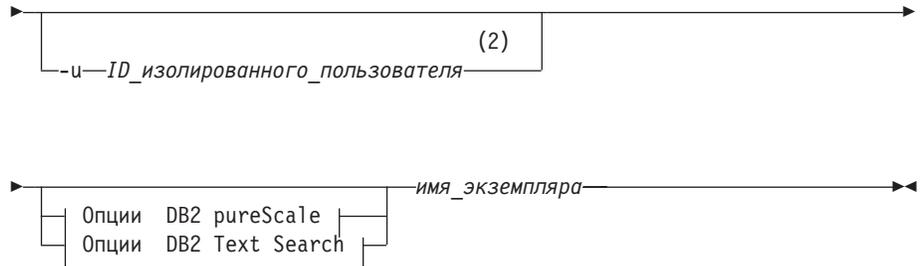
### Авторизация

Требуются полномочия пользователя root в операционных системах Linux и UNIX или локального администратора в операционных системах Windows.

### Синтаксис команды

Для операционных систем Linux и UNIX

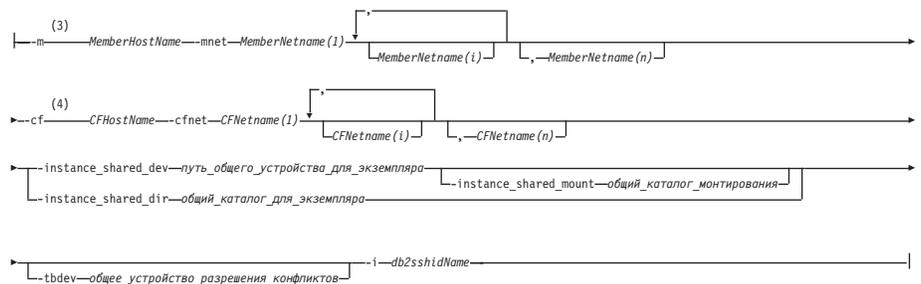




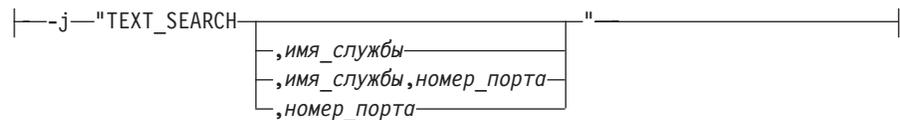
### InstType:



### DB2 pureScale опции:



### DB2 Text Search опции:

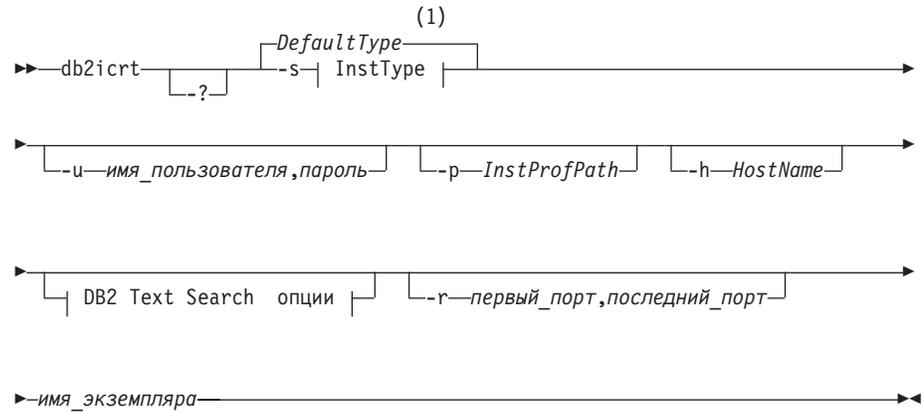


### Примечания:

- 1 Если тип экземпляра не задан с опцией -s, созданный в результате тип экземпляра по умолчанию будет типом самого высокого ранга, указанным в *InstType*, на создание которого у копии DB2 есть лицензия.
- 2 При создании клиентских экземпляров опция -u *FencedID* недопустима.
- 3 Формат *MemberHostName:MemberNetname* для опции -m является устаревшим, и в будущем его поддержка может быть прекращена. Для поддержки IPv6 с DB2 pureScale Feature требуется новый формат с обеими опциями: -m и -mnet.
- 4 Формат *CFHostName:CFNetNames* для опции -cf объявлен устаревшим, и

в будущем его поддержка может быть прекращена. Для поддержки IPv6 с DB2 pureScale Feature требуется новый формат с обеими опциями: `-sf` и `-cfnet`.

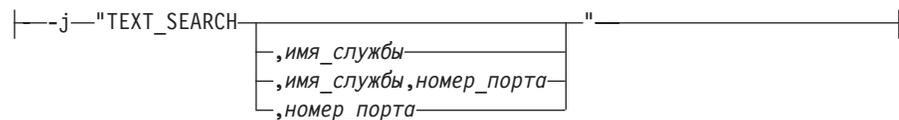
### Для операционных систем Windows



#### InstType:



#### DB2 Text Search опции:



#### Примечания:

- 1 Если тип экземпляра не задан с `-s`, созданный в результате тип экземпляра по умолчанию будет типом самого высокого ранга, указанным в *InstType*, на создание которого у копии DB2 есть лицензия.

## Параметры команды

### Для операционных систем Linux и UNIX

- `-?` Выводит информацию об использовании.
- `-h` Выводит информацию об использовании.
- `-d` Включает режим отладки. Сохраняет файл трассировки с именем по умолчанию в `/tmp` как `db2icrt.trc.ID_процесса`. Используйте эту опцию только по указанию службы поддержки баз данных DB2
- `-a тип_аутентификации`  
Задаёт тип аутентификации (SERVER, CLIENT или SERVER\_ENCRYPT) для этого экземпляра. Значение по умолчанию - SERVER.

#### **-j "TEXT\_SEARCH"**

Конфигурирует сервер DB2 Text Search с использованием сгенерированных значений по умолчанию для имени службы и номера порта TCP/IP. Этот параметр нельзя использовать, если тип экземпляра - клиент.

#### **-j "TEXT\_SEARCH,имя\_службы"**

Конфигурирует сервер DB2 Text Search с использованием заданного имени службы и автоматически сгенерированного номера порта. Если для заданного имени службы в файле `services` задан номер порта, используется этот номер порта.

#### **-j "TEXT\_SEARCH,имя\_службы,номер\_порта"**

Конфигурирует сервер DB2 Text Search с использованием заданного имени службы и номера порта.

#### **-j "TEXT\_SEARCH,номер\_порта"**

Конфигурирует сервер DB2 Text Search с использованием имени службы по умолчанию и заданного номера порта. Допустимые номера порта должны быть в диапазоне 1024 - 65535.

#### **-p <имя порта TCP/IP>**

Задаёт имя или номер порта TCP/IP, используемого этим экземпляром. Эта опция также задаёт параметр конфигурации менеджера баз данных **SVCENAME** для экземпляра DB2.

#### **-m MemberHostName:NetName1**

Задаёт хост для конфигурирования в качестве DB2 элемент при создании экземпляра. Это обязательный параметр в среде DB2 pureScale. С помощью команды **db2icrt** можно сконфигурировать только один DB2 элемент. Дополнительные DB2 элементы можно добавить при помощи команды **db2iupdt -add**. Синтаксис *NetName1* объявлен устаревшим и может быть исключен из будущего выпуска. Используйте вместо него параметр **-mnet**.

Переменная *MemberHostName* должна быть каноническим именем хоста (взятым, например, из вывода команды 'hostname', запущенной на локальном хосте). Заданное здесь значение *NetName1* должно принадлежать той же подсети, как указано в параметре **-cf**.

#### **-mnet MemberNetName**

Этот параметр заменяет устаревший синтаксис *:NetName1* параметра **-m MemberHostName:NetName1**. Задаёт сетевое имя межсоединения кластера, которым служит имя хоста соединения, используемого для высокоскоростной связи между участниками и механизмами кэширования кластера (другое название - CF) в экземпляре DB2 pureScale.

Переменная *MemberNetName* должна принадлежать к одной из тех же подсетей, указанных в параметре **-cf**, и соответствовать сетевому имени взаимодействия кластера (например, *db2\_<имя\_хоста\_ib0>*).

#### **-cf CFHostName:NetName2**

Задаёт хост для конфигурирования в качестве механизм кэширования кластера (другое название - CF) при создании экземпляра. Это обязательный параметр в среде DB2 pureScale. С помощью команды **db2icrt** можно сконфигурировать только один CF. Дополнительные CF можно добавить при помощи команды **db2iupdt -add**. Синтаксис

*NetName2* объявлен устаревшим и может быть исключен из будущего выпуска. Используйте вместо него параметр **-cfnet**.

**-cfnet** *CFNetName*

Этот параметр заменяет устаревший синтаксис *:NetName2* параметра **-cf CFHostName:NetName2**. Задает сетевое имя межсоединения кластера, которым служит имя хоста соединения, используемого для высокоскоростной связи между участниками и механизмами кэширования кластера в экземпляре DB2 pureScale.

Переменная *CFNetName* должна принадлежать к той же подсети, что задана в параметре **-m**, и соответствовать сетевому имени взаимодействия кластера, например, *db2\_<имя\_хоста\_ib0>*).

**-instance\_shared\_dev** *путь\_общего\_устройства\_для\_экземпляра*

Задает путь совместно используемого диска, необходимый, чтобы сконфигурировать экземпляр DB2 pureScale для хранения совместно используемых файлов экземпляра и пути базы данных по умолчанию. Например, */dev/hdisk1*. Совместно используемый каталог должен быть доступным на всех хостах экземпляра DB2 pureScale. Значение этой опции не может совпадать со значением опции **-tbdev**.

Если задан параметр **-instance\_shared\_dev**, программа установки DB2 создаст кластерная файловая система DB2.

Параметры **-instance\_shared\_dev** и **-instance\_shared\_dir** - взаимоисключающие.

**-instance\_shared\_mount** *общий\_каталог\_монтирования*

Задает точку монтирования для новой файловой системы IBM General Parallel File System ( GPFS). Указанный путь должен быть новым и пустым путем, не вложенным в существующую файловую систему GPFS.

**-instance\_shared\_dir** *общий\_каталог\_для\_экземпляра*

Задает каталог в совместно используемой файловой системе (GPFS), необходимый, чтобы сконфигурировать экземпляр DB2 pureScale для хранения совместно используемых файлов экземпляра и пути базы данных по умолчанию. Например, */sharedfs*. Этот диск должен быть доступным на всех хостах экземпляра DB2 pureScale. Значение этой опции не может совпадать со значением опции **-tbdev** или с путем установки.

Если задан параметр **-instance\_shared\_dir**, программа установки DB2 будет использовать файловую систему, управляемую пользователем. Управляемая пользователем файловая система должна быть доступна на всех хостах и должна быть файловой системой GPFS.

Параметры **-instance\_shared\_dir** и **-instance\_shared\_dev** - взаимоисключающие.

**-tbdev** *общее\_устройство\_разрешения\_конф.ликов*

Задает путь совместно используемого устройства, которое будет действовать как устройство разрешения конфликтов в среде DB2 pureScale для поддержания целостности данных. Значение этой опции не может совпадать со значением опции **-instance\_shared\_dev**, а также опции **-instance\_shared\_dir**. Это обязательная опция, если устройство разрешения конфликтов служб кластера создается впервые. С этим дисковым устройством не может быть связана

файловая система. Если равноправный домен служб кластера уже существует, это недопустимая опция.

**-i db2sshidName**

Задаёт ID пользователя без полномочий root, необходимый для использования протокола сети защищенной оболочки (secure shell, SSH) между хостами. Необходимо задать ID пользователя без специальных привилегий. Допустима только для файловой системы GPFS, управляемой DB2.

**-s тип\_экземпляра**

Задаёт тип создаваемого экземпляра. Используйте опцию **-s** только при создании экземпляра, отличного от экземпляра по умолчанию, связанного с установленным продуктом, из которого запускается команда **db2icrt**. Допустимые значения:

**dsf** Используется для создания экземпляра DB2 pureScale для сервера баз данных DB2 с локальными и удаленными клиентами. Это тип экземпляра по умолчанию для Возможность IBM DB2 pureScale.

**ese** Используется для создания экземпляра сервера базы данных с локальными и удаленными клиентами. Это тип экземпляра по умолчанию для DB2 Enterprise Server Edition или DB2 Advanced Enterprise Server Edition.

**wse** Используется для создания экземпляра сервера базы данных с локальными и удаленными клиентами. Это тип экземпляра по умолчанию для DB2 Workgroup Server Edition, DB2 Express Edition или DB2 Express-C и DB2 Connect Enterprise Edition.

**standalone**

Используется для создания экземпляра сервера баз данных с локальными клиентами.

**client** Используется для создания экземпляра клиента. Это тип экземпляра по умолчанию для IBM Data Server, IBM Data Server и DB2 Connect Personal Edition.

Продукты базы данных DB2 поддерживают свои экземпляры по умолчанию и более младшие типы экземпляров. Для экземпляра DB2 Enterprise Server Edition поддерживает типы экземпляров ese, wse, standalone и client.

**-u ID\_изолир\_пользователя**

Задаёт имя ID пользователя, применяемое для запуска изолированных пользовательских функций и изолированных хранимых процедур. Опция **-u** требуется, если вы не создаете экземпляр клиента.

*InstName*

Задаёт имя экземпляра, совпадающее с именем существующего пользователя в операционной системе. Имя экземпляра должно быть последним аргументом команды **db2icrt**.

**Для операционных систем Windows**

*InstName*

Задаёт имя экземпляра.

**-s** *тип\_экземпляра*

Задает тип создаваемого экземпляра. В настоящее время существует четыре типа экземпляров DB2. Допустимые значения:

**client** Используется для создания экземпляра клиента. Это тип экземпляра по умолчанию для IBM Data Server, IBM Data Server и DB2 Connect Personal Edition.

**standalone**

Используется для создания экземпляра сервера баз данных с локальными клиентами.

**ese** Используется для создания экземпляра сервера баз данных с локальными и удаленными клиентами и поддержкой среды многораздельных баз данных. Откроется окно

`-s ese -u имя_пользователя, пароль`

Опции надо использоваться с командой **db2icrt** для создания типа экземпляра ESE и экземпляра среды многораздельных баз данных.

**wse** Используется для создания экземпляра сервера базы данных с локальными и удаленными клиентами. Это тип экземпляра по умолчанию для DB2 Workgroup Server Edition, DB2 Express Edition или DB2 Express-C и DB2 Connect Enterprise Edition.

Продукты базы данных DB2 поддерживают свои экземпляры по умолчанию и более младшие типы экземпляров. Для экземпляра DB2 Enterprise Server Edition поддерживает типы экземпляров *ese*, *wse*, *standalone* и *client*.

**-u** *имя\_пользователя, пароль*

Задает имя пользователя и пароль для службы DB2. Эта опция обязательна при создании экземпляра распределенной базы данных.

**-p** *InstProfPath*

Задает путь профиля экземпляра.

**-h** *HostName*

Переопределяет имя хоста TCP/IP по умолчанию, если для текущего компьютера есть несколько имен хостов. Это имя хоста TCP/IP используется при создании раздела базы данных по умолчанию (раздел базы данных 0). Эта опция действительна только для экземпляров многораздельных баз данных.

**-r** *PortRange*

Задает диапазон портов TCP/IP, которые используются экземпляром распределенной базы данных при работе в режиме MPP. Например, `-r 50000,50007`. Если эта опция задана, в файле *services* на локальном компьютере будут обновлены следующие записи:

```
DB2_InstName      начальный_порт/tcp
DB2_InstName_END  конечный_порт/tcp
```

**/j** **"TEXT\_SEARCH"**

Конфигурирует сервер DB2 Text Search с использованием сгенерированных значений по умолчанию для имени службы и номера порта TCP/IP. Этот параметр нельзя использовать, если тип экземпляра - клиент.

`/j "TEXT_SEARCH, имя_службы"`

Конфигурирует сервер DB2 Text Search с использованием заданного имени службы и автоматически сгенерированного номера порта. Если для заданного имени службы в файле `services` задан номер порта, используется этот номер порта.

```
/j "TEXT_SEARCH, имя_службы, номер_порта"
```

Конфигурирует сервер DB2 Text Search с использованием заданного имени службы и номера порта.

```
/j "TEXT_SEARCH, номер_порта"
```

Конфигурирует сервер DB2 Text Search с использованием имени службы по умолчанию и заданного номера порта. Допустимые номера порта должны быть в диапазоне 1024 - 65535.

-? Выводит информацию об использовании.

## Примеры

1. Чтобы создать экземпляр DB2 pureScale для владельца экземпляра `db2sdin1` и изолированного пользователя `db2sdfel`, введите команду:

```
DB2DIR/instance/db2icrt
-cf host1.domain.com -cfnet host1.domain.com-ib0
-m host2.domain.com -mnet host2.domain.com-ib0
-instance_shared_dev /dev/hdisk1
-tbdev /dev/hdisk2
-u db2sdfel
db2sdin1
```

где *каталог*DB2 - положение установки вашей копии DB2. У экземпляра DB2 pureScale `db2sdin1` CF будет на хосте `host1`, а участник - на хосте `host2`. Эта команда также использует путь `/dev/hdisk1`, чтобы создать совместно используемую файловую систему для хранения совместно используемых файлов экземпляра и конфигурирует путь файла `/dev/hdisk2` как путь совместно используемого устройства разрешения конфликтов.

2. Чтобы создать экземпляр DB2 Enterprise Server Edition для ID пользователя `db2inst1`, введите команду:

```
DB2DIR/instance/db2icrt -s ese -u db2fenc1 db2inst1
```

где *каталог*DB2 - положение установки вашей копии DB2.

3. Чтобы создать экземпляр DB2 pureScale, использующий существующую файловую систему (GPFS), управляемую продуктом DB2 для владельца экземпляра `db2sdin1` и изолированного пользователя `db2sdfel`, введите команду:

```
каталог_DB2/instance/db2icrt
-cf host1.domain.com -cfnet host1.domain.com-ib0
-m host2.domain.com -mnet host2.domain.com-ib0
-tbdev /dev/hdisk2
-u db2sdfel
db2sdin1
```

где *каталог*\_DB2 - положение установки вашей копии DB2.

4. Чтобы создать экземпляр DB2 pureScale с существующей файловой системой GPFS, управляемой пользователем (`/gpfs_shared_dir`), для владельца экземпляра `db2sdin1` и изолированного пользователя `db2sdfel`, введите команду:

```
каталог_DB2/instance/db2icrt
-cf host1.domain.com -cfnet host1.domain.com-ib0
-m host2.domain.com -mnet host2.domain.com-ib0
```

```
-instance_shared_dir /gpfs_shared_dir
-tbdev /dev/hdisk2
-u db2sdfel
db2sdin1
```

где *каталог\_DB2* - положение установки вашей копии DB2.

5. На компьютере AIX для создания экземпляра для ID пользователя `db2inst1` введите следующую команду:

На компьютере клиента:

```
DB2DIR/instance/db2icrt db2inst1
```

На компьютере сервера:

```
DB2DIR/instance/db2icrt -u db2fenc1 db2inst1
```

где `db2fenc1` - ID пользователя, от имени которого будут запускаться изолированные пользовательские функции и изолированные хранимые процедуры.

## Замечания по использованию

- Пользователь экземпляра должен существовать на всех хостах с одним и тем же UID, GID, именем группы и путем домашнего каталога. То же правило относится и к изолированному пользователю. После успешного выполнения команды **db2icrt** программа установки DB2 сконфигурирует SSH для пользователя экземпляра на всех хостах.
- При использовании команды **db2icrt** имя экземпляра должно совпадать с именем существующего пользователя.
- В среде DB2 pureScale может быть только один экземпляр.
- При создании экземпляров DB2 учитывайте следующие ограничения:
  - Если для создания экземпляров DB2 используются существующие ID, убедитесь, что эти ID не заблокированы и сроки их паролей не истекли.
- При помощи команды **db2isetup** можно также создать или изменить экземпляры DB2 и добавить несколько хостов с графическим интерфейсом.
- Если для получения полномочий root вы используете команду **su** вместо команды **login**, надо ввести команду **su** с опцией **-**, которая указывает задание такой же среды процесса, что и при регистрации в системе при помощи команды **login**.
- Не надо брать в качестве исходной среду экземпляра DB2 для пользователя root. Выполнение команды **db2icrt**, когда среда экземпляра DB2 используется как источник, не поддерживается.
- Если вы ранее создали экземпляр DB2 pureScale, который был отброшен, его нельзя воссоздать с помощью параметра `-instance_shared_dev`, поскольку кластерная файловая система DB2 уже могла быть создана. Чтобы задать ранее созданную совместно используемую файловую систему:
  - Если существующая совместно используемая файловая система GPFS создана и управляется DB2 pureScale Feature, использовать параметры `-instance_shared_dev` и `-instance_shared_dir` не нужно.
  - Если существующая совместно используемая файловая система GPFS не была создана и не управляется DB2 pureScale Feature, воспользуйтесь параметром `-instance_shared_dir`.
- В AIX 6.1 (или в более новых версиях), когда эта команда выполняется из копии DB2 совместного использования в глобальной среде системных разделов рабочей нагрузки (workload partition, WPAR), эта команда должна быть запущена от имени пользователя root. WPAR не поддерживается в среде DB2 pureScale.

## db2idrop - удаление экземпляра

Удаляет экземпляр DB2, созданный командой **db2icrt**.

Вы можете отбросить только экземпляры, которые выводятся в списке командой **db2ilist** для той же копии DB2, откуда введена команда **db2idrop**. Кроме того, командой **db2idrop** можно отбросить экземпляр DB2 pureScale.

В операционных системах Linux и UNIX эта утилита расположена в каталоге *DB2DIR/instance*, где *DB2DIR* - это положение установки, где установлена текущая версия системы баз данных DB2. В операционных системах Windows эта утилита расположена в каталоге *DB2PATH\bin*, где *DB2PATH* - это положение, где установлена копия DB2.

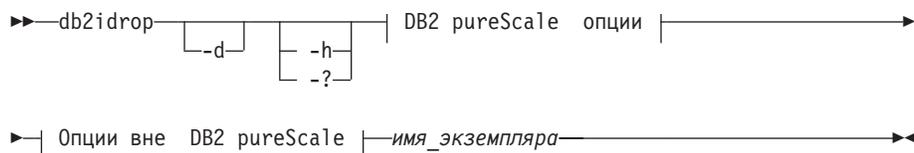
**Примечание:** С помощью этой команды нельзя отбросить экземпляр DB2 установки без полномочий root в операционных системах Linux и UNIX. Единственная возможность - деинсталлировать копию DB2, установленную без полномочий root. Подробную информацию смотрите в разделе *Замечания об использовании*.

### Авторизация

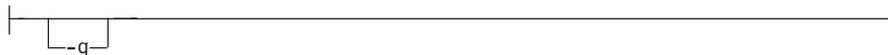
Полномочия пользователя root в операционных системах Linux и UNIX или локального администратора в операционных системах Windows.

### Синтаксис команды

Для операционных систем Linux и UNIX



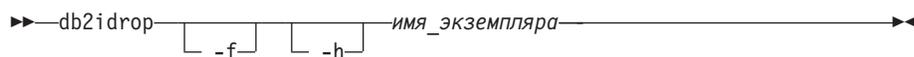
**DB2 pureScale опции:**



**Опции вне DB2 pureScale:**



Для операционных систем Windows



### Параметры команды

Для операционных систем Linux и UNIX

- d** Вводит режим отладки для использования службой поддержки баз данных DB2.

- h | -?** Выводит информацию об использовании.
- g** Этот параметр - обязательный, если **db2idrop** используется с экземпляром DB2 pureScale. Указывает, что вы хотите отбросить экземпляр DB2 pureScale на всех хостах. Этот параметр требует, чтобы все элементы DB2 и все механизмы кэширования кластера были остановлены на всех хостах в экземпляре DB2 pureScale. Эта опция будет проигнорирована, если отбрасывается любой другой тип экземпляра
- f** Этот параметр устарел.  
Задает флаг принудительного завершения. Если этот флаг задан, все программы, использующие этот экземпляр, будут принудительно завершены. Этот параметр не поддерживается в среде DB2 pureScale.

*InstName*

Задает имя экземпляра.

#### Для операционных систем Windows

- f** Задает флаг принудительного завершения. Если этот флаг задан, все программы, использующие этот экземпляр, будут принудительно завершены.
- h** Выводит информацию об использовании.

*InstName*

Задает имя экземпляра.

### Пример

Если вы создали `db2inst1` в операционных системах Linux или UNIX, запустив следующую команду:

```
/opt/IBM/db2/copy1/instance/db2icrt -u db2fenc1 db2inst1
```

Чтобы отбросить `db2inst1`, надо ввести следующую команду:

```
/opt/IBM/db2/copy1/instance/db2idrop db2inst1
```

### Замечания по использованию

- Перед отбрасыванием экземпляра убедитесь, что менеджер баз данных DB2 остановлен на всех хостах, а обращающиеся к базе данных DB2 прикладные программы отсоединены и прерваны. Должно быть выполнено резервное копирование связанных с экземпляром баз данных DB2, а данные о конфигурации сохранены для возможного использования в будущем.
- Команда **db2idrop** не удаляет базы данных. Сначала удалите базы данных, если они больше не требуются. Если базы данных не удалены, их всегда можно каталогизировать под другой копией DB2 того же выпуска и продолжать их использование.
- Если вы хотите сохранить конфигурации DB2 Text Search и планируете повторно использовать базы данных экземпляра, необходимо предпринять некоторые дополнительные действия по сохранению каталога `config` (в UNIX: *домашний\_каталог\_экземпляра/sql1ib/db2tss/config*, в Windows: *путь\_к\_профилю\_экземпляра\имя\_экземпляра\db2tss\config*) или содержимого каталога `config` перед запуском команды **db2idrop**. После создания нового экземпляра каталог `config` можно восстановить. Однако восстановление каталога `config` применимо только в том случае, если новый экземпляр создан в том же выпуске и на том же уровне пакета Fix Pack.

- Экземпляр установки без полномочий root нельзя отбросить в операционных системах Linux и UNIX. Единственная доступная для пользователя возможность удалить такой экземпляр DB2 - деинсталлировать копию DB2, установленную без полномочий root, введя команду **db2\_deinstall -a**.
- В операционных системах Linux и UNIX, если получаете полномочия root при помощи команды **su** вместо команды **login**, вводите команду **su** с опцией **-**, которая задает ту же среду процесса, как и при регистрации в системе при помощи команды **login**.
- В операционных системах Linux и UNIX не используйте среду экземпляра DB2 как источник для пользователя root. Выполнение команды **db2idrop**, когда среда экземпляра DB2 используется как источник, не поддерживается.
- В среде DB2 pureScale параметр **-g** обязателен. В этом случае экземпляр будет отброшен на всех хостах. Однако файловая система IBM General Parallel File System (GPFS) в хост, иницирующий установку (ИИ) не будет удалена, как и файловая система GPFS. Необходимо вручную удалить файловую систему и деинсталлировать GPFS.
- В операционных системах Windows, если экземпляр включен в кластер при помощи Microsoft Cluster Service (MSCS), перед отбрасыванием экземпляра его можно исключить из кластера, введя команду **db2mcs** или **db2iclus**.
- В AIX 6.1 (или в более новых версиях), когда эта команда выполняется из копии DB2 совместного использования в глобальной среде системных разделов рабочей нагрузки (workload partition, WPAR), эта команда должна быть запущена от имени пользователя root. WPAR не поддерживается в среде DB2 pureScale.

---

## db2iupdt - Обновление экземпляров

Командой **db2iupdt** можно обновить другой экземпляр до более высокого уровня в выпуске, обновить экземпляр, отличный от экземпляра DB2 pureScale, до экземпляра DB2 pureScale или же масштабировать экземпляр DB2 pureScale.

Перед командой **db2iupdt** необходимо остановить экземпляр и все запущенные для него процессы. При использовании этой команды для обновления экземпляра, отличного от экземпляра DB2 pureScale, добавьте участника или механизм кэширования кластера, а также проверьте, что установлены все необходимые программы и выполнен список предварительных требований к экземпляру и хостам.

**Примечание:** В Экземпляр DB2 pureScale нельзя изменять модель ресурсов без *конфигурационного кворума*, то есть абсолютного большинства узлов в подключенном состоянии. При конфигурировании с двумя хостами команду **db2iupdt** нельзя использовать, если один из этих хостов отключен.

### Авторизация

Полномочия пользователя root в операционных системах UNIX и Linux или локального администратора в операционных системах Windows.

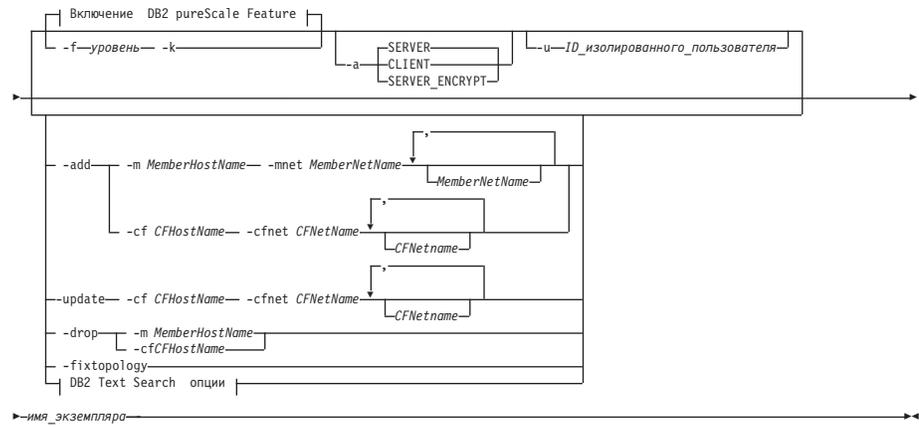
### Синтаксис команды

Для операционных систем UNIX и Linux

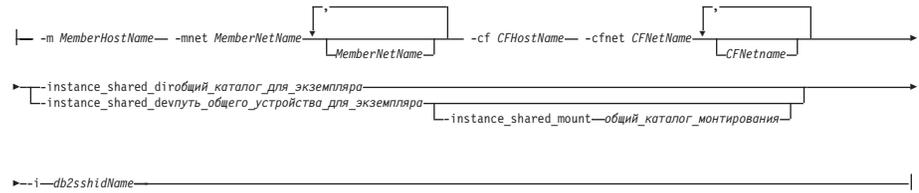
```

→ db2iupdt [-h] [-d] [-D]

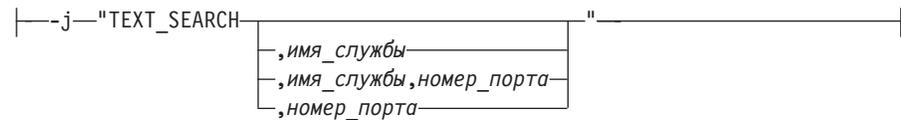
```



### Включение DB2 pureScale Feature:



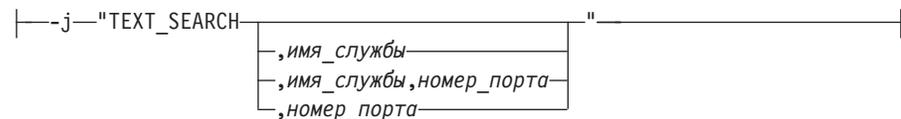
### DB2 Text Search опции:



### Для операционных систем Windows



### DB2 Text Search опции:



## Параметры команды

### Для операционных систем UNIX и Linux

- h | -?** Выводит информацию об использовании.
- a тип\_аутентификации**  
Задает тип аутентификации (SERVER, SERVER\_ENCRYPT или CLIENT) для этого экземпляра. Значение по умолчанию - SERVER.
- d** Включает режим отладки.
- k** Сохраняет текущий тип экземпляра при обновлении.
- D** Этот параметр объявлен устаревшим, и в одном из следующих выпусков он будет удален. Замените его параметром **-f level**. Переносит экземпляр из кода высшего уровня, установленного по одному пути, в код более низкого уровня, установленный по другому пути.
- f level**  
Изменяет тип экземпляра с более новой версии DB2 на более раннюю версию DB2 для обеспечения совместимости.
- add -m MemberHostName:Netname1 | -cf CFHostName:Netname2**  
Задает хост (элемент или механизм кэширования кластера), добавляемый в экземпляр DB2 pureScale.  
  
Чтобы задать тип добавляемого хоста, воспользуйтесь параметром **-m** для элемента или параметром **-cf** для механизма кэширования кластера. Надо выбрать только один, но не оба варианта.  
  
Команду **db2iupdt -add** надо вводить с хоста, который уже представляет собой часть экземпляра DB2 pureScale.  
  
Синтаксис **-m MemberHostName:NetName1** и **-cf CFHostName:NetName2** устарел и может не поддерживаться в будущих выпусках. Используйте вместо него синтаксис **-add -m MemberHostName -mnet MemberNetName** или **-add -cf CFHostName -cfnet CFNetName**.
- add -m MemberHostName -mnet MemberNetName | -add -cf CFHostName -cfnet CFNetName**  
Задает сетевое имя межсоединения кластера, которым служит имя хоста соединения, используемого для высокоскоростной связи между участниками и механизмом кэширования кластера DB2. Сетевое имя каждого хоста, который вы хотите добавить, должно принадлежать к той же подсети, что и все существующие участники или CF в экземпляре.  
  
Чтобы задать тип добавляемого хоста, воспользуйтесь параметром **-m MemberHostName -mnet MemberNetName** для элемента или **-cf CFHostName -cfnet CFNetName** для механизма кэширования кластера. Надо выбрать только один, но не оба варианта.  
  
Команду **db2iupdt -add** надо вводить с хоста, который уже представляет собой часть экземпляра DB2 pureScale.
- add** Задает имя хоста и сетевое имя межсоединения кластера или сетевые имена хоста, добавляемые в экземпляр DB2 pureScale Feature. Команду **db2iupdt -add** надо вводить с хоста, который уже представляет собой часть экземпляра DB2 pureScale.  
  
**-m MemberHostName -mnet MemberNetName**  
Хост с именем *MemberHostName* добавляется в экземпляр

DB2 pureScale Feature сетевым именем взаимодействия кластера *MemberNetName*. Если для *MemberHostName* есть несколько сетевых портов взаимодействия кластеров, укажите для *MemberHostName* список имен взаимодействия кластеров, разделенных запятыми.

**-cf** *CFHostName* **-cfnet** *CFNetName*

Хост с именем *CFHostName* добавляется в экземпляр DB2 pureScale Feature как механизм кэширования кластера с сетевым именем взаимодействия кластера *CFNetName*. Если для *CFHostName* есть несколько сетевых портов взаимодействия кластеров, укажите для *CFNetName* список имен взаимодействия кластеров, разделенных запятыми.

**-drop -m** *MemberHostName* | **-cf** *CFHostName*

Задаёт хост (элемент или механизм кэширования кластера), отбрасываемый из экземпляра DB2 pureScale. Чтобы задать тип отбрасываемого хоста, укажите опцию **-m**, если тип - элемент, или **-cf**, если тип - механизм кэширования кластера. Надо выбрать только один, но не оба варианта. Этот параметр нельзя использовать, чтобы отбросить последний элемент и последний CF из экземпляра DB2 pureScale. Этот параметр нельзя использовать вместе с параметром **-add**.

**-instance\_shared\_dev** *путь\_общего\_устройства\_для\_экземпляра*

Задаёт путь совместно используемого диска, необходимый, чтобы сконфигурировать экземпляр DB2 pureScale для хранения совместно используемых файлов экземпляра и пути базы данных по умолчанию. Например, путь устройства */dev/hdisk1*. Совместно используемый каталог должен быть доступным на всех хостах экземпляра DB2 pureScale. Значение этого параметра не может совпадать со значением параметра **-tbdev**. Этот параметр и параметр **-instance\_shared\_dir** - взаимоисключающие.

Этот параметр обязателен, только если вы обновляете экземпляр, отличный от экземпляра DB2 pureScale до экземпляра DB2 pureScale.

**-instance\_shared\_mount** *общий\_каталог\_монтирования*

Задаёт точку монтирования для новой файловой системы IBM General Parallel File System ( GPFS). Указанный путь должен быть новым и пустым путем, не вложенным в существующую файловую систему GPFS.

**-instance\_shared\_dir** *общий\_каталог\_для\_экземпляра*

Задаёт каталог в совместно используемой файловой системе (GPFS), необходимый, чтобы сконфигурировать экземпляр DB2 pureScale для хранения совместно используемых файлов экземпляра и пути базы данных по умолчанию. Например, */sharedfs*. Этот диск должен быть доступным на всех хостах экземпляра DB2 pureScale. Значение этого параметра не может совпадать со значением параметра **-tbdev**. Этот параметр и параметр **-instance\_shared\_dev** - взаимоисключающие.

Этот параметр обязателен, только если вы обновляете экземпляр, отличный от экземпляра DB2 pureScale до экземпляра DB2 pureScale.

**-tbdev** *общее\_устройство\_разрешения\_конф.ликов*

Задаёт путь совместно используемого устройства, которое будет действовать как устройство разрешения конфликтов в среде DB2 pureScale для поддержания целостности данных. Значение этого

параметра не может совпадать со значением параметра **-instance\_shared\_dev** или параметра **-instance\_shared\_dir**. Этот параметр обязателен, когда устройство разрешения конфликтов служб кластера создано впервые, или при изменении экземпляра, отличного от экземпляра DB2 pureScale, до экземпляра DB2 pureScale. Если равноправный домен служб кластера существует, этот параметр недопустим.

**-i db2sshidName**

Задаёт ID пользователя без полномочий root, необходимый для использования протокола сети защищённой оболочки (secure shell, SSH) между хостами. Необходимо задать ID пользователя без специальных привилегий. Допустим только для файловой системы GPFS, управляемой DB2.

**-j "TEXT\_SEARCH"**

Конфигурирует сервер DB2 Text Search с использованием сгенерированных значений по умолчанию для имени службы и номера порта TCP/IP. Этот параметр нельзя использовать, если тип экземпляра - клиент или dsf.

**-j "TEXT\_SEARCH,имя\_службы"**

Конфигурирует сервер DB2 Text Search с использованием заданного имени службы и автоматически сгенерированного номера порта. Если для заданного имени службы в файле `services` задан номер порта, используется этот номер порта.

**-j "TEXT\_SEARCH,имя\_службы,номер\_порта"**

Конфигурирует сервер DB2 Text Search с использованием заданного имени службы и номера порта.

**-j "TEXT\_SEARCH,номер\_порта"**

Конфигурирует сервер DB2 Text Search с использованием имени службы по умолчанию и заданного номера порта. Допустимые номера порта должны быть в диапазоне 1024 - 65535.

**-u Fenced ID**

Задаёт имя ID пользователя, применяемое для запуска изолированных пользовательских функций и изолированных хранимых процедур. Это единственный параметр, необходимый при преобразовании типа экземпляра `client` в отличный от `client` тип. Для определения текущего типа экземпляра посмотрите параметр типа узла в выводе команды **GET DBM CFG**. Если тип экземпляра уже отличен от `client` или тип `client` остаётся типом экземпляра (например, при указании параметра **-k**), параметр **-u** не нужен. Параметр **-u** может изменить изолированного пользователя для существующего экземпляра.

**-fixtopology**

Используется для исправления вручную неудачно завершившейся операции добавления или отбрасывания. При операциях добавления этот параметр вызовет откат всех изменений с возвратом к предыдущей топологии. Операции отбрасывания с этим оператором будут выполнены. Этот параметр нельзя использовать в сочетании с любыми другими параметрами, кроме **-d**.

*InstName*  
Задает имя экземпляра.

#### Для операционных систем Windows

*InstName*  
Задает имя экземпляра.

*/u:имя-пользователя,пароль*  
Задает имя пользователя и пароль для службы DB2.

*/p:путь-профиля-экземпляра*  
Задает новый путь профиля экземпляра для обновляемого экземпляра.

*/r:начальный\_порт,конечный\_порт*  
Задает диапазон портов TCP/IP, которые используются экземпляром распределенной базы данных при работе в режиме MPP. Когда задана эта опция, файл служб на локальном компьютере будет дополнен следующими записями:

```
DB2_InstName      начальный_порт/tcp  
DB2_InstName_END  конечный_порт/tcp
```

*/h:имя\_хоста*  
Переопределяет имя хоста TCP/IP по умолчанию, если для текущего компьютера есть несколько имен хостов TCP/IP.

*/s* Изменяет экземпляр в многораздельный экземпляр.

*/q* Запускает команду **db2iupdt** в режиме без вывода сообщений.

*/a:authType*  
Задает *authType* - тип аутентификации (SERVER, CLIENT или SERVER\_ENCRYPT) для этого экземпляра.

*/j "TEXT\_SEARCH"*  
Конфигурирует сервер DB2 Text Search с использованием сгенерированных значений по умолчанию для имени службы и номера порта TCP/IP. Этот параметр нельзя использовать, если тип экземпляра - клиент.

*/j "TEXT\_SEARCH, имя\_службы"*  
Конфигурирует сервер DB2 Text Search с использованием заданного имени службы и автоматически сгенерированного номера порта. Если для заданного имени службы в файле `services` задан номер порта, используется этот номер порта.

*/j "TEXT\_SEARCH, имя\_службы, номер\_порта"*  
Конфигурирует сервер DB2 Text Search с использованием заданного имени службы и номера порта.

*/j "TEXT\_SEARCH, номер\_порта"*  
Конфигурирует сервер DB2 Text Search с использованием имени службы по умолчанию и заданного номера порта. Допустимые номера порта должны быть в диапазоне 1024 - 65535.

*/?* Выводит информацию об использовании для команды **db2iupdt**.

## Пример

### Для операционных систем UNIX и Linux

Экземпляр `db2inst2` связан с копией DB2 продукта баз данных DB2, установленного в каталоге `DB2DIR1`. У вас есть другая копия продукта баз данных DB2 на том же компьютере в каталоге `DB2DIR2` для той же версии продукта баз данных DB2, что установлена в каталоге `DB2DIR1`. Для изменения экземпляра, чтобы он запускался не из копии DB2, установленной в каталоге `DB2DIR1`, а из копии DB2, установленной в каталоге `DB2DIR2`, введите команду:

```
DB2DIR2/instance/db2iupdt db2inst2
```

Если копия DB2, установленная в каталоге `DB2DIR2`, более старой версии по сравнению с копией DB2, установленной в каталоге `DB2DIR1`, введите команду:

```
DB2DIR2/instance/db2iupdt -D db2inst2
```

### Обновление экземпляра до более высокого уровня того же выпуска

Чтобы изменить экземпляр DB2 до более новой версии или изменить его путь установки DB2, введите следующую команду:

```
DB2DIR/instance/db2iupdt db2inst1
```

где `каталогDB2` - положение установки вашей копии DB2. Если эта команда запускается из копии DB2 pureScale Feature, существующий экземпляр `db2inst1` должен быть типа `dsf`. Если экземпляр `db2inst1` является экземпляром DB2 pureScale, описанная в этом примере команда может изменить уровень его версии ESE DB2 с помощью DB2 pureScale Feature. Этот пример неприменим к изменению типа экземпляра `ese` на экземпляр DB2 pureScale. Данная процедура описана в следующем примере.

### Обновите экземпляр (не экземпляр DB2 pureScale) до экземпляра DB2 pureScale

Чтобы изменить экземпляр на экземпляр DB2 pureScale:

```
DB2DIR/instance/db2iupdt
-cf host2
-cfnet host2-ib0
-m host1
-mnet host1-ib0
-instance_shared_dev /dev/hdisk1
-tbdev /dev/hdisk2
-u db2fenc1
db2inst1
```

где `каталогDB2` - положение установки вашей копии DB2.

Эта команда также использует путь `/dev/hdisk1`, чтобы создать совместно используемую файловую систему для хранения совместно используемых файлов экземпляра и конфигурирует путь файла `/dev/hdisk2` как путь совместно используемого устройства для разрешения конфликтов. Значение параметра `-tbdev` должно отличаться от значения параметра `-instance_shared_dev`.

### Промасштабируйте экземпляр DB2 pureScale (при помощи команды `db2iupdt -add` или `db2iupdt -drop`)

Для среды DB2 pureScale применяются следующие примеры:

- Измените экземпляр DB2 pureScale, чтобы добавить элемент. Чтобы добавить элемента с именем `host1` и сетевым именем `host1-ib0` в экземпляр DB2 pureScale `db2sdin1`, введите команду:

```
DB2DIR/instance/db2iupdt -d -add -m host1 -mnet host1-ib0 db2sdin1
```

где *каталог* DB2 - положение установки вашей копии DB2.

- Измените экземпляр DB2 pureScale, чтобы добавить второй механизм кэширования кластера. Чтобы добавить механизм кэширования кластера с именем *host2* и сетевым именем *host2-ib0* в экземпляр DB2 pureScale *instance db2sdin1*, введите команду:

```
DB2DIR/instance/db2iupdt -d -add -cf host2 -cfnet host2-ib0 db2sdin1
```

где *каталог*DB2 - положение установки вашей копии DB2.

- Отбросьте элемент из экземпляра DB2 pureScale. Чтобы отбросить элемент с именем *host1* из экземпляра DB2 pureScale *db2sdin1*, введите команду:

```
DB2DIR/instance/db2iupdt -d -drop -m host1 db2sdin1
```

где *каталог*DB2 - положение установки вашей копии DB2. Если у хоста *host1* нет в этом же экземпляре роли механизма кэширования кластера, эту команду надо вводить с другого хоста (не *host1*).

## Замечания по использованию

### Для всех поддерживаемых операционных систем

- Если вы изменяете с помощью команды **db2iupdt** экземпляр DB2, с одной копии DB2 на другую копию DB2 одной и той же версии продукта баз данных DB2, глобальные переменные профиля DB2, определенные в пути установки старой копии DB2, не будут приведены в соответствие с новым путем установки. Переменные профиля экземпляра DB2, специфичные для этого экземпляра, будут перенесены в нужное положение после завершения изменения экземпляра.

### Для операционных систем UNIX и Linux

- Командой **db2iupdt** можно обновить только DB2DB2 Enterprise Server Edition.
- Если в топологию участников вносятся какие-либо изменения, например, добавляется или отбрасывается участник, необходимо сделать автономную резервную копию базы данных, прежде чем обращаться к ней. При попытке доступа к базе данных до получения автономной резервной копии база данных переводится в состояние отложенного резервного копирования.

Можно добавить или отбросить сразу несколько участников без необходимости резервного копирования после каждого изменения. Например, если добавляется три участника, резервное копирование нужно выполнить только после завершения всех операций добавления. Но если вы добавляете двух участников, а потом отбрасываете одного участника, или отбрасываете двух участников, а потом добавляете одного участника, резервное копирование необходимо выполнять до очередного изменения топологии.

- Команда **db2iupdt** расположена в каталоге *DB2DIR/instance*, где *DB2DIR* - это положение, где установлена текущая версия продукта баз данных DB2.
- Если вы хотите изменить экземпляр установки без полномочий *root*, посмотрите команду изменения таких экземпляров **db2nrupdt**. Команда **db2iupdt** не поддерживает изменение экземпляров установки без полномочий *root*.

- Если для получения полномочий root вы используете команду **su** вместо команды **login**, надо ввести команду **su** с опцией **-**, которая указывает задание такой же среды процесса, что и при регистрации в системе при помощи команды **login**.
- Не надо брать в качестве исходной среду экземпляра DB2 для пользователя root. Выполнение команды **db2iupdt**, когда среда экземпляра DB2 используется как источник, не поддерживается.
- В AIX 6.1 (или в более новых версиях), когда эта команда выполняется из копии DB2 совместного использования в глобальной среде системных разделов рабочей нагрузки (workload partition, WPAR), эта команда должна быть запущена от имени пользователя root. WPAR не поддерживается в среде DB2 pureScale.
- Если команда **db2iupdt** используется для обновления экземпляра до более высокого уровня в выпуске, подпрограммы и библиотеки будут скопированы из каждого участника в совместно используемое положение. Если у библиотеки совпадают имена, но не совпадает содержимое на разных хостах, содержимое этой библиотеки в совместно используемом положении будет определяться последним хостом, с которого была введена команда **db2iupdt**.

#### Для операционных систем Windows

- Команда **db2iupdt** расположена в каталоге **DB2PATH\bin**, где **DB2PATH** - это положение, где установлена текущая версия продукта баз данных DB2.
- Экземпляр обновляется до копии DB2, из которой вы запускаете команду **db2iupdt**. Но чтобы переместить профиль вашего экземпляра из его текущего положения в другое, используйте опцию **/p** и задайте путь к профилю экземпляра. В противном случае профиль экземпляра останется после изменения в своем исходном положении. Вместо обновления от предыдущего выпуска до текущего используйте команду **db2iupgrade**.



---

## Приложение D. Обзор технической информации DB2

Техническая информация DB2 доступна в различных форматах, и к ней можно обращаться различными способами.

Для получения технической информации по DB2 служат следующие инструменты и методы:

- DB2 Информационный центр
  - Темы (Задачи, принципы работы и справочные темы)
  - Примеры программ
  - Обучающие программы
- Книги по DB2
  - Файлы PDF (загружаемые)
  - Файлы PDF (с диска DVD DB2 PDF)
  - Печатные книги
- Справка командной строки
  - Справка по командам
  - Справка по сообщениям

**Примечание:** Темы Информационного центра DB2 обновляются чаще, чем книги в формате PDF или бумажные книги. Чтобы получить наиболее свежую информацию, устанавливайте обновления сразу после их появления или используйте Информационный центр DB2 на сайте [ibm.com](http://ibm.com).

Дополнительную техническую информацию о DB2, такую как технические замечания (technotes), оригинальные публикации (white papers) и технические руководства (IBM Redbooks) можно найти на сайте [ibm.com](http://ibm.com). Обратитесь к библиотеке на сайте DB2 Information Management по адресу <http://www.ibm.com/software/data/sw-library/>.

### Отзывы о документации

Мы ценим ваши отзывы о документации по DB2. Если у вас есть предложения по совершенствованию документации по DB2, присылайте электронные сообщения на адрес [db2docs@ca.ibm.com](mailto:db2docs@ca.ibm.com). Коллектив разработчиков документации DB2 читает все ваши отзывы, но не может отвечать напрямую. Чтобы мы могли лучше понять ваши трудности, по возможности представьте конкретные примеры. Если ваш отзыв касается конкретной темы или файла справки, приложите заголовок справки и URL.

Не обращайтесь по этому адресу с вопросами, требующими службы поддержки заказчиков DB2. Если у вас есть технический вопрос по DB2, который не удается разрешить по документации, обратитесь за помощью в ваш местный центр обслуживания IBM.

---

## Техническая библиотека DB2 в печатном виде или в формате PDF

Ниже в таблице описан состав библиотеки DB2, доступной в Центре публикаций IBM по адресу [www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss](http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss). Английские и переведенные версии руководств DB2 V10.1 в формате PDF можно скачать с сайта [www.ibm.com/support/docview.wss?rs=71&uid=swg27009474](http://www.ibm.com/support/docview.wss?rs=71&uid=swg27009474).

В таблицах описаны книги, доступные в печатном виде, тем не менее, в вашей стране или регионе они могут не быть доступны в печатном виде.

Номер формы увеличивается при каждом обновлении руководства. Убедитесь, что вы читаете самую свежую версию документации, как указано ниже.

**Примечание:** Информационный центр DB2 обновляется чаще, чем книги в формате PDF или бумажные книги.

Таблица 26. Техническая информация по DB2

Имя	Номер формы	Доступна в печатном виде	Последнее изменение
<i>Administrative API Reference</i>	SC27-3864-00	Да	апрель 2012 г.
<i>Administrative Routines and Views</i>	SC27-3865-01	Нет	Январь, 2013
<i>Справочное руководство по интерфейсу уровня вызовов, Том 1</i>	SC27-3866-01	Да	Январь, 2013
<i>Справочное руководство по интерфейсу уровня вызовов, Том 2</i>	SC27-3867-01	Да	Январь, 2013
<i>Command Reference</i>	SC27-3868-01	Да	Январь, 2013
<i>Database Administration Concepts and Configuration Reference</i>	SC27-3871-01	Да	Январь, 2013
<i>Data Movement Utilities Guide and Reference</i>	SC27-3869-01	Да	Январь, 2013
<i>Database Monitoring Guide and Reference</i>	SC27-3887-01	Да	Январь, 2013
<i>Справочное руководство по восстановлению данных и высокой доступности</i>	SC27-3870-01	Да	Январь, 2013
<i>Database Security Guide</i>	SC27-3872-01	Да	Январь, 2013
<i>Справочное руководство по DB2 Workload Management</i>	SC27-3891-01	Да	Январь, 2013
<i>Developing ADO.NET and OLE DB Applications</i>	SC27-3873-01	Да	Январь, 2013
<i>Developing Embedded SQL Applications</i>	SC27-3874-01	Да	Январь, 2013
<i>Developing Java Applications</i>	SC27-3875-01	Да	Январь, 2013
<i>Developing Perl, PHP, Python, and Ruby on Rails Applications</i>	SC27-3876-00	Нет	апрель 2012 г.
<i>Разработка программ RDF для IBM Data Servers</i>	SC27-4462-00	Да	Январь, 2013

Таблица 26. Техническая информация по DB2 (продолжение)

Имя	Номер формы	Доступна в печатном виде	Последнее изменение
<i>Developing User-defined Routines (SQL and External)</i>	SC27-3877-01	Да	Январь, 2013
<i>Getting Started with Database Application Development</i>	GI13-2046-01	Да	Январь, 2013
<i>Начинаем работу по установке и управлению DB2 в Linux и Windows</i>	GI13-2047-00	Да	апрель 2012 г.
<i>Globalization Guide</i>	SC27-3878-00	Да	апрель 2012 г.
<i>Установка серверов DB2</i>	GC89-1240-01	Да	Январь, 2013
<i>Установка клиентов IBM Data Server</i>	GC89-1242-00	Нет	апрель 2012 г.
<i>Message Reference Volume 1</i>	SC27-3879-01	Нет	Январь, 2013
<i>Message Reference Volume 2</i>	SC27-3880-01	Нет	Январь, 2013
<i>Net Search Extender Administration and User's Guide</i>	SC27-3895-01	Нет	Январь, 2013
<i>Partitioning and Clustering Guide</i>	SC27-3882-01	Да	Январь, 2013
<i>Preparation Guide for DB2 10.1 Fundamentals Exam 610</i>	SC27-4540-00	Нет	Январь, 2013
<i>Preparation Guide for DB2 10.1 DBA for Linux, UNIX, and Windows Exam 611</i>	SC27-4541-00	Нет	Январь, 2013
<i>pureXML Guide</i>	SC27-3892-01	Да	Январь, 2013
<i>Spatial Extender User's Guide and Reference</i>	SC27-3894-00	Нет	апрель 2012 г.
<i>SQL Procedural Languages: Application Enablement and Support</i>	SC27-3896-01	Да	Январь, 2013
<i>Справочник по SQL, Том 1</i>	SC27-3885-01	Да	Январь, 2013
<i>Справочник по SQL, Том 2</i>	SC27-3886-01	Да	Январь, 2013
<i>Руководство по текстовому поиску</i>	SC89-1259-01	Да	Январь, 2013
<i>Troubleshooting and Tuning Database Performance</i>	SC27-3889-01	Да	Январь, 2013
<i>Обновление до DB2 Версии 10.1</i>	SC27-3881-01	Да	Январь, 2013

Таблица 26. Техническая информация по DB2 (продолжение)

Имя	Номер формы	Доступна в печатном виде	Последнее изменение
<i>Что нового для DB2 Версии 10.1</i>	SC89-1246-01	Да	Январь, 2013
<i>XQuery Reference</i>	SC27-3893-01	Нет	Январь, 2013

Таблица 27. Техническая информация о DB2 Connect

Имя	Номер формы	Доступна в печатном виде	Последнее изменение
<i>DB2 Connect Установка и конфигурирование DB2 Connect Personal Edition</i>	SC89-1245-00	Да	апрель 2012 г.
<i>DB2 Connect Установка и конфигурирование серверов DB2 Connect</i>	SC89-1244-01	Да	Январь, 2013
<i>DB2 Connect. Руководство пользователя</i>	SC89-1243-01	Да	Январь, 2013

## Вызов справки по SQLSTATE из командной строки

Продукты DB2 возвращают SQLSTATE - значения, описывающие ошибки, которые могут возникнуть при выполнении оператора SQL. Справка по SQLSTATE объясняет значение состояний SQL и коды классов состояний SQL.

### Процедура

Для вызова справки по SQLSTATE откройте процессор командной строки и введите:

```
? sqlstate или ? код класса
```

где *sqlstate* - допустимый пятизначный код SQL, а *код класса* - первые две цифры sqlstate.

Например, ? 08003 выводит справку по состоянию SQL 08003, а ? 08 выводит справку по коду класса 08.

## Доступ к различным версиям Информационного центра DB2

Документацию для других версий продуктов DB2 можно найти в отдельных информационных центрах на [ibm.com](http://ibm.com).

### Об этой задаче

Темы DB2 V10.1 смотрите в *Информационном центре DB2* по адресу <http://publib.boulder.ibm.com/infocenter/db2luw/v10r1>.

Темы DB2 Версии 9.8 смотрите в *Информационном центре DB2* по адресу <http://publib.boulder.ibm.com/infocenter/db2luw/v9r8/>.

Темы DB2 Версии 9.7 смотрите в *Информационном центре DB2* по адресу <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/>.

Темы DB2 Версии 9.5 смотрите в *Информационном центре DB2* по адресу <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5>.

Темы DB2 Версии 9.1 смотрите в *Информационном центре DB2* по адресу <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>.

Для тем DB2 Версии перейдите по URL *Информационного центра DB2*: <http://publib.boulder.ibm.com/infocenter/db2luw/v8/>.

---

## Изменение Информационного центра DB2, установленного на вашем компьютере или на сервере интранета

Локально установленный Информационный центр DB2 надо периодически обновлять.

### Прежде чем начать

Информационный центр DB2 Версия 10.1 должен быть уже установлен. Подробности смотрите в теме “Установка Информационного центра DB2 при помощи мастера по установке DB2” в книге *Установка серверов DB2*. Все предварительные требования и ограничения для установки Информационного центра касаются также и обновления Информационного центра.

### Об этой задаче

Существующий Информационный центр DB2 можно обновлять автоматически или вручную:

- Автоматическое обновление обновляет существующие возможности и языки Информационного центра. Одно из преимуществ автоматического обновления состоит в том, что Информационный центр недоступен более короткий период времени в ходе обновления по сравнению с ручным обновлением. Кроме того, можно задать периодический запуск автоматического обновления в составе других пакетных работ.
- Ручное обновление можно использовать для изменения существующих возможностей и языковой поддержки Информационного центра. Автоматическое обновление сокращает время простоя в процессе обновления, но для добавления возможностей и языковой поддержки требуется использовать ручное обновление. Например, локальный Информационный центр исходно был установлен на английском и на французском языке, но теперь вы хотите установить также русский язык; при обновлении вручную будет установлен русский язык и обновлена информация на английском и на французском языке. Однако при ручном обновлении от вас требуется вручную остановить, обновить и перезапустить Информационный центр. Информационный центр недоступен в ходе всего процесса обновления. При автоматическом обновлении Информационный центр находится в простое и может восстановиться только после завершения обновления.

В данной теме подробно описан процесс автоматического обновления. Инструкции по обновлению вручную смотрите в теме “Обновление Информационного центра DB2, установленного на вашем компьютере или на сервере интранет, вручную”.

### Процедура

Чтобы автоматически обновить DB2, установленный на вашем компьютере или на сервере интранета:

1. В операционных системах Linux:

- a. Перейдите в каталог, где установлен Информационный центр. По умолчанию Информационный центр DB2 установлен в каталоге /opt/ibm/db2ic/V10.1.
  - b. Перейдите из каталога установки в каталог doc/bin.
  - c. Запустите сценарий update-ic:  
update-ic
2. В операционных системах Windows:
- a. Откройте командное окно.
  - b. Перейдите в каталог, где установлен Информационный центр. По умолчанию Информационный центр DB2 установлен в каталоге <Program Files>\IBM\DB2 Information Center\Версия 10.1, где <Program Files> соответствует положению каталога Program Files.
  - c. Перейдите из каталога установки в каталог doc\bin.
  - d. Запустите файл update-ic.bat:  
update-ic.bat

## Результаты

Информационный центр DB2 будет перезапущен автоматически. Если обновления доступны, в Информационном центре будут показаны новые и обновленные темы. Если обновления Информационного центра недоступны, в журнал будет записано сообщение. Файл журнала находится в каталоге doc\eclipse\configuration. Имя файла журнала - случайным образом сгенерированное число, например, 1239053440785.log.

---

## Обновление Информационного центра DB2, установленного на вашем компьютере или на сервере интранет, вручную

Если вы установили Информационный центр DB2 локально, можно получить обновления документации от IBM и установить их.

### Об этой задаче

Для обновления вашего локально установленного *Информационного центра DB2* вручную:

1. Остановите *Информационный центр DB2* на вашем компьютере и перезапустите его в автономном режиме. Запуск Информационного центра в автономном режиме предотвращает доступ к нему других пользователей по сети и дает возможность применить исправления. Версия Информационного центра DB2 для рабочей станции всегда работает в автономном режиме. .
2. Чтобы посмотреть доступные обновления, воспользуйтесь возможностью Изменить. При помощи этой возможности можно получить и установить нужные вам обновления.

**Примечание:** Если в вашей среде необходимо установить обновления *Информационного центра DB2* на компьютере, не подключенном к интернету, создайте зеркальную копию сайта обновлений в локальной файловой системе, используя компьютер, который подключен к Интернету и на котором установлен *Информационный центр DB2*. Кроме того, создание зеркальной копии и прокси-сервера сайта обновлений полезно, если в вашей сети многие пользователи будут устанавливать обновления документации, и вы хотите сократить время, затраченное на отдельные обновления.

Если пакеты доступны, используйте эту возможность для получения пакетов. Но возможность обновления доступна только в автономном режиме.

3. Остановите Информационный центр в автономном режиме и перезапустите *Информационный центр DB2* на вашем компьютере.

**Примечание:** В Windows 2008 и Windows Vista (и новее) перечисленные далее в этом разделе команды должен вводить администратор. Чтобы открыть окно командной строки или графическое средство со всеми привилегиями администратора, щелкните по ярлыку правой кнопкой и выберите **Выполнить от имени администратора**.

## Процедура

Чтобы обновить *Информационный центр DB2*, установленный на вашем компьютере или на сервере интранета:

1. Остановите *Информационный центр DB2*.
  - В Windows выберите **Пуск > Панель управления > Администрирование > Службы**. Затем щелкните правой кнопкой мыши по службе **Информационный центр DB2** и выберите **Стоп**.
  - В Linux введите команду:  
`/etc/init.d/db2icdv10 stop`
2. Запустите Информационный центр в автономном режиме.
  - В Windows:
    - a. Откройте командное окно.
    - b. Перейдите в каталог, где установлен Информационный центр. По умолчанию *Информационный центр DB2* установлен в каталоге `Program Files\IBM\DB2 Information Center\Версия 10.1`, где `Program Files` соответствует положению каталога Program Files.
    - c. Перейдите из каталога установки в каталог `doc\bin`.
    - d. Запустите файл `help_start.bat`:  
`help_start.bat`
  - В Linux:
    - a. Перейдите в каталог, где установлен Информационный центр. По умолчанию *Информационный центр DB2* установлен в каталоге `/opt/ibm/db2ic/V10.1`.
    - b. Перейдите из каталога установки в каталог `doc/bin`.
    - c. Запустите сценарий `help_start`:  
`help_start`

В системном браузере по умолчанию откроется Информационный центр в автономном режиме.
3. Нажмите кнопку **Изменить** (🔧). (В вашем браузере должен быть разрешен JavaScript.) На правой панели Информационного центра выберите **Найти изменения**. Будет выведен список изменений для существующей документации.
4. Чтобы начать процесс установки, пометьте документы, которые вы хотите установить, затем нажмите кнопку **Установить изменения**.
5. Когда процесс установки будет завершен, нажмите кнопку **Готово**.
6. Остановите Информационный центр в автономном режиме:
  - В Windows перейдите в подкаталог `doc\bin` каталога установки и запустите файл `help_end.bat`:  
`help_end.bat`

**Примечание:** Пакетный файл `help_end` содержит команды, необходимые для корректной остановки процессов, запущенных пакетным файлом `help_start`. Не используйте сочетание клавиш `Ctrl-C` или другие методы для остановки работы `help_start.bat`.

- В Linux перейдите в подкаталог `doc/bin` каталога установки и запустите сценарий `help_end`:  
`help_end`

**Примечание:** Сценарий `help_end` содержит команды, необходимые для корректной остановки процессов, запущенных сценарием `help_start`. Не используйте другие методы для остановки работы сценария `help_start`.

7. Перезапустите *Информационный центр DB2*.

- В Windows выберите **Пуск > Панель управления > Администрирование > Службы**. Затем щелкните правой кнопкой мыши по службе **Информационный центр DB2** и выберите **Пуск**.
- В Linux введите команду:  
`/etc/init.d/db2icdv10 start`

## Результаты

В обновленном *Информационном центре DB2* будут показаны новые и измененные темы.

---

## Учебные материалы DB2

Обучающие программы DB2 помогают познакомиться с различными аспектами продуктов DB2. В уроках содержатся пошаговые инструкции.

### Прежде, чем вы начнете

XHTML-версию обучающей программы можно просмотреть в Информационном центре на сайте <http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/>.

В некоторых уроках используются примеры данных или кодов программ. Описание необходимых условий для выполнения задач обучающей программы смотрите в самой обучающей программе.

### Учебные материалы DB2

Для просмотра Обучающей программы щелкните по заголовку.

**“pureXML” в книге *pureXML Guide***

Конфигурирование базы данных DB2 для хранения данных XML и выполнение основных операций со складом собственных данных XML.

---

## Информация об устранении неисправностей DB2

Для продуктов баз данных DB2 доступна разнообразная информация по устранению неисправностей и диагностике ошибок.

### Документация DB2

Информацию по устранению неисправностей можно найти в публикации *Troubleshooting and Tuning Database Performance* или в разделе Основные понятия баз данных *Информационного центра DB2*, который содержит:

- Информация о способах поиска и определения проблем с помощью инструментов диагностики и утилит DB2.
- Решения некоторых наиболее общих проблем.
- Помощь при других проблемах, которые могут встретиться в ваших продуктах баз данных DB2.

#### **Портал поддержки IBM**

Смотрите портал поддержки IBM если вы сталкиваетесь с проблемами и вам нужна помощь в поиске причин и решений. На сайте технической поддержки есть ссылки на самые свежие публикации по DB2, TechNotes, исправления программ (APAR), пакеты корректировки и прочие ресурсы. Для решения ваших проблем можно воспользоваться поиском по этой базе знаний.

Обратитесь к portalу поддержки IBM по адресу [http://www.ibm.com/support/entry/portal/Overview/Software/Information\\_Management/DB2\\_for\\_Linux\\_UNIX\\_and\\_Windows](http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/DB2_for_Linux_UNIX_and_Windows)

## **Положения и условия**

Разрешения на использование этих публикаций предоставляются в соответствии со следующими положениями и условиями.

**Применимость:** Данные положения и условия дополняют любые условия применения для сайта IBM.

**Личное использование:** Вам предоставляется право воспроизводить эти публикации в личных некоммерческих целях при условии, что будут воспроизведены все замечания об авторских правах. Вам запрещается распространять эти публикации, полностью или по частям, демонстрировать их или создавать из них производные продукты без явного на то согласия от IBM.

**Коммерческое использование:** Вам предоставляется право воспроизводить эти публикации исключительно в пределах своего предприятия при условии, что будут воспроизведены все замечания об авторских правах. За пределами вашего предприятия вам запрещается распространять эти публикации, полностью или по частям, демонстрировать их или создавать из них производные продукты без явного на то согласия от IBM.

**Права:** На публикации или любую информацию, данные, программные средства или иную интеллектуальную собственность, содержащуюся в публикациях, предоставляются только права, явным образом предоставленные настоящим документом; никакие иные разрешения, лицензии и права не предоставляются.

IBM оставляет за собой право отозвать разрешения, предоставленные этим документом, если, по мнению IBM, использование публикаций наносит ущерб IBM или, как это установлено IBM, вышеприведенные инструкции не соблюдаются должным образом.

Вы имеете право загружать, экспортировать или реэкспортировать эту информацию только при условии соблюдения всех применимых законов и нормативных актов, включая все законы и нормативные акты США, касающиеся экспорта.

**IBM НЕ ДАЕТ НИКАКИХ ГАРАНТИЙ ОТНОСИТЕЛЬНО СОДЕРЖАНИЯ ЭТИХ ПУБЛИКАЦИЙ. ДАННЫЕ ПУБЛИКАЦИИ ПРЕДОСТАВЛЕНЫ "КАК ЕСТЬ", БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, ЯВНЫХ ИЛИ ПОДРАЗУМЕВАЕМЫХ, ВКЛЮЧАЯ (НО НЕ ОГРАНИЧИВАЯСЬ ТАКОВЫМИ) ПРЕДПОЛАГАЕМЫЕ**

ГАРАНТИИ СОБЛЮДЕНИЯ АВТОРСКИХ ПРАВ, РЫНОЧНОЙ ПРИГОДНОСТИ ИЛИ СООТВЕТСТВИЯ ОПРЕДЕЛЕННОЙ ЦЕЛИ.

**Товарные знаки IBM:** IBM, логотип IBM и [ibm.com](http://ibm.com) - товарные знаки или зарегистрированные товарные знаки International Business Machines Corp в Соединенных Штатах и/или других странах. Другие названия продуктов и услуг могут быть товарными знаками IBM или других компаний. Текущий список товарных знаков IBM доступен в Web под заголовком по адресу [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)

---

## Приложение Е. Замечания

Эта информация описывает продукты и услуги, предлагаемые в США. Информация о продуктах других фирм (не IBM) основана на данных, доступных к моменту первой публикации этого и может быть изменена.

IBM может не предлагать описанные продукты, услуги и возможности в других странах. Сведения о продуктах и услугах, доступных в настоящее время в вашей стране, можно получить в местном представительстве IBM. Любые ссылки на продукты, программы или услуги IBM не означают явным или неявным образом, что можно использовать только продукты, программы или услуги IBM. Разрешается использовать любые функционально эквивалентные продукты, программы или услуги, если при этом не нарушаются права IBM на интеллектуальную собственность. Однако ответственность за оценку и проверку работы любых продуктов, программ и услуг других фирм лежит на пользователе.

Фирма IBM может располагать патентами или рассматриваемыми заявками на патенты, относящимися к предмету данного документа. Получение этого документа не означает предоставления каких-либо лицензий на эти патенты. Запросы по поводу лицензий следует направлять в письменной форме по адресу:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

По поводу лицензий, связанных с использованием наборов двухбайтных символов (DBCS), обращайтесь в отдел интеллектуальной собственности IBM в вашей стране или направьте запрос в письменной форме по адресу:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
1623-14, Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 Japan

**Следующий абзац не применяется в Великобритании или в любой другой стране/регионе, где подобные заявления противоречат местным законам:** КОРПОРАЦИЯ INTERNATIONAL BUSINESS MACHINES ПРЕДСТАВЛЯЕТ ДАННУЮ ПУБЛИКАЦИЮ “КАК ЕСТЬ” БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, ЯВНЫХ ИЛИ ПОДРАЗУМЕВАЕМЫХ, ВКЛЮЧАЯ ПРЕДПОЛАГАЕМЫЕ ГАРАНТИИ СОВМЕСТИМОСТИ, РЫНОЧНОЙ ПРИГОДНОСТИ И СООТВЕТСТВИЯ ОПРЕДЕЛЕННОЙ ЦЕЛИ, НО НЕ ОГРАНИЧИВАЯСЯ ИМИ. В некоторых странах для определенных сделок подобные оговорки не допускаются; таким образом, это утверждение может не относиться к вам.

Данная информация может содержать технические неточности и типографские опечатки. Периодически в информацию вносятся изменения, они будут включены в новые издания этой публикации. Фирма IBM может в любое время без уведомления вносить изменения и усовершенствования в продукты и программы, описанные в этой публикации.

Все ссылки в этой информации на сайты, не принадлежащие IBM, приводятся только для вашего удобства и ни в коей мере не означают одобрения этих сайтов. Материалы на этих сайтах не входят в состав материалов для данного продукта IBM; используйте их на свою ответственность.

IBM может использовать или распространять присланную вами информацию любым способом, как фирма сочтет нужным, без каких-либо обязательств перед вами.

Если обладателю лицензии на данную программу понадобятся сведения о возможности: (i) обмена данными между независимо разработанными программами и другими программами (включая данную) и (ii) совместного использования таких данных, он может обратиться по адресу:

IBM Canada Limited  
U59/3600  
3600 Steeles Avenue East  
Markham, Ontario L3R 9Z7  
CANADA

Такая информация может быть предоставлена на определенных условиях (в некоторых случаях к таким условиям может относиться оплата).

Лицензированная программа, описанная в данном документе, и все лицензированные материалы, доступные с ней, предоставляются IBM на условиях IBM Customer Agreement (Соглашения IBM с заказчиком), Международного соглашения о лицензиях на программы IBM или эквивалентного соглашения.

Приведенные данные о производительности измерены в контролируемой среде. Таким образом, результаты, полученные в других операционных средах, могут существенно отличаться от них. Некоторые показатели получены в системах разработки, и нет никаких гарантий, что в общедоступных системах эти показатели будут теми же. Более того, некоторые результаты могут быть получены путем экстраполяции. Реальные результаты могут отличаться от них. Пользователи должны проверить данные для своих конкретных сред.

Информация о продуктах других фирм получена от поставщиков этих продуктов, из их опубликованных объявлений или из других общедоступных источников. Фирма IBM не проверяла эти продукты и не может подтвердить точность измерений, совместимость или прочие утверждения о продуктах других фирм. Вопросы о возможностях продуктов других фирм следует направлять поставщикам этих продуктов.

Все утверждения о будущих планах и намерениях IBM могут быть изменены или отменены без уведомлений, и описывают исключительно цели фирмы.

Эта информация может содержать примеры данных и отчетов, иллюстрирующие типичные деловые операции. Чтобы эти примеры были правдоподобны, в них включены имена лиц, названия компаний и товаров. Все эти имена и названия вымышлены и любое их сходство с реальными именами и адресами полностью случайно.

#### ЛИЦЕНЗИЯ НА КОПИРОВАНИЕ:

Эта информация содержит примеры прикладных программ на языках программирования, иллюстрирующих приемы программирования для различных операционных платформ. Разрешается копировать, изменять и распространять эти

примеры программ в любой форме без оплаты фирме IBM для целей разработки, использования, сбыта или распространения прикладных программ, соответствующих интерфейсу прикладного программирования операционных платформ, для которых эти примера программ написаны. Эти примеры не были всесторонне проверены во всех возможных условиях. Поэтому IBM не может гарантировать их надежность, пригодность и функционирование. Программы примеров предоставляются "КАК ЕСТЬ", без каких-либо гарантий. IBM не несет ответственности за какой-либо ущерб, возникший в результате применения вами программ примеров.

Каждая копия программ примеров или программ, созданных на их основе, должна содержать следующее замечание об авторских правах:

© (название вашей фирмы) (год). Части этого кода построены на основе примеров программ IBM Corp. © Copyright IBM Corp. *\_вставьте год или годы\_*. Все права защищены.

## Товарные знаки

IBM, логотип IBM и [ibm.com](http://ibm.com) - товарные знаки или зарегистрированные товарные знаки International Business Machines Corp в Соединенных Штатах и/или других странах. Другие названия продуктов и услуг могут быть товарными знаками IBM или других компаний. Текущий список товарных знаков IBM доступен в Web под заголовком "Copyright and trademark information" по адресу [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Следующие термины - товарные знаки или зарегистрированные товарные знаки других компаний.

- Linux - зарегистрированный товарный знак Linus Torvalds в Соединенных Штатах и/или других странах.
- Java и все товарные знаки и логотипы на основе Java - товарные знаки или зарегистрированные товарные знаки Oracle и/или ее филиалов.
- UNIX - зарегистрированный товарный знак The Open Group в Соединенных Штатах и в других странах.
- Intel, логотип Intel, Intel Inside, логотип Intel Inside, Celeron, Intel SpeedStep, Itanium и Pentium - товарные знаки или зарегистрированные товарные знаки Intel Corporation или ее филиалов в Соединенных Штатах и в других странах.
- Microsoft, Windows, Windows NT и логотип Windows - товарные знаки корпорации Microsoft в США и/или других странах.

Названия других компаний, продуктов и услуг могут быть товарными знаками или марками сервиса других фирм.



---

# Индекс

## D

DB2 Text Search  
SQL 127  
вызов команд 79  
индексы  
    обслуживание 96  
настройка системы 35  
обзор 19  
поиск  
    индексы 107  
полномочия  
    администратор баз данных 24  
роли  
    администратор баз данных 24  
словари синонимов  
    обзор 86  
типы данных  
    поддерживаемые 19  
требования к портам TCP/IP 35  
функции 107  
функции поиска XML 127  
функция RESULTLIMIT 39

## L

Linux  
    установка  
        серверы DB2 53  
        файл ответов 57

## S

SYSIBMTS.TSSERVERS 211

## T

TCP/IP  
    обновление файла services 67

## U

UNIX  
    установка  
        мастер по установке DB2 53  
установка с помощью файла ответов 57

## W

Windows  
    установка  
        серверов DB2 при помощи мастера по установке DB2 50  
файлы ответов  
    установка при помощи 56

## X

XML  
    текстовый поиск DB2  
        грамматика EBNF 114  
        синтаксис поиска 121

## V

внесение в каталог  
    узлы TCP/IP 67

## Д

диагностика ошибок  
    доступная информация 258  
    учебные материалы 258  
документация  
    обзор 251  
    печатные 251  
    положения и условия использования 259  
    файлы PDF 251

## З

замечания 261

## И

изменения  
    Информационный центр DB2 255, 256  
Информационный центр DB2  
    версии 254  
    изменение 255, 256

## К

команда db2icrt  
    подробности 230  
команда db2idrop  
    подробности 239  
команда db2iupdt  
    подробности 241  
команда db2iupgrade  
    подробности 227  
команда HELP  
    Текстовый поиск 159  
команда обновления экземпляра 227  
команда обновления экземпляров 241  
команда создания экземпляра 230  
Команда текстового поиска ALTER INDEX 138  
Команда текстового поиска CLEANUP FOR TEXT 143  
Команда текстового поиска CLEAR COMMAND LOCKS 143  
команда текстового поиска CLEAR EVENTS FOR INDEX 145  
Команда текстового поиска CREATE INDEX 147  
команда текстового поиска DB2 RESET PENDING 161  
Команда текстового поиска DISABLE DATABASE FOR TEXT 154  
Команда текстового поиска DROP INDEX 156

Команда текстового поиска ENABLE DATABASE FOR TEXT 158

Команда текстового поиска START FOR TEXT 162

Команда текстового поиска STOP FOR TEXT 164

Команда текстового поиска UPDATE INDEX 165

команда удаления экземпляра 239

команды

db2icrt

подробности 230

db2idrop

подробности 239

db2iupdt

подробности 241

db2iupgrade

подробности 227

db2ts ALTER INDEX 138

db2ts CLEANUP FOR TEXT 143

db2ts CLEAR COMMAND LOCKS 143

db2ts CLEAR EVENTS FOR INDEX 145

db2ts CREATE INDEX 147

db2ts DISABLE DATABASE FOR TEXT 154

db2ts DROP INDEX 156

db2ts ENABLE DATABASE FOR TEXT 158

db2ts HELP 159

db2ts RESET PENDING 161

db2ts START FOR TEXT 162

db2ts STOP FOR TEXT 164

db2ts UPDATE INDEX 165

вызов из процедуры 169

команды db2ts

ALTER INDEX 138

CLEANUP FOR TEXT 143

CLEAR COMMAND LOCKS 143

CLEAR EVENTS FOR INDEX 145

CREATE INDEX 147

DISABLE DATABASE FOR TEXT 154

DROP INDEX 156

ENABLE DATABASE FOR TEXT 158

HELP 159

RESET PENDING 161

START FOR TEXT 162

STOP FOR TEXT 164

UPDATE INDEX 165

## M

мастер по установке DB2

установка

серверы DB2 (Linux) 53

серверы DB2 (UNIX) 53

модуль Net Search Extender

сравнение с DB2 Text Search 219

## O

Операторы SQL

справка

вывод 254

## P

положения и условия

публикации 259

Производная таблица SYSIBMTS.TSINDEXES 212

производные таблицы для DB2 Text Search

информация уровня базы данных

SYSIBMTS.TSDEFAULTS 209

SYSIBMTS.TSLOCKS 210

обзор 209

информация уровня индекса

SYSIBMTS.TSCOLLECTIONNAMES 215

SYSIBMTS.TSCONFIGURATION 214

SYSIBMTS.TSEVENT 215

SYSIBMTS.TSSTAGING 216

обзор 209

производных таблиц

текстовый поиск DB2

SYSIBMTS.TSINDEXES 212

информация уровня индекса 212

Пространства имен XML 40

процедура SYSTS\_ADMIN\_CMD 169

процедура SYSTS\_CREATE 185

процедура SYSTS\_DISABLE 195

процедура SYSTS\_START 183

процедуры

SYSTS\_ADMIN\_CMD 169

SYSTS\_CREATE 185

SYSTS\_START 183

## C

серверы DB2

установка

Windows 50

словари синонимов

добавление 87

обзор 86

удаление 87

справка

Операторы SQL 254

столбцов XML

текстовый поиск 132

## T

текстовые индексы

контекстный поиск 110

текстовый поиск

текстовый поиск DB2 80

Текстовый поиск

смотрите текстовый поиск DB2 1

текстовый поиск DB2

SQL 108

XQuery

xmlcolumn-contains 132

способы полнотекстового поиска 108

автономная установка 49

аргументы поиска

влияние на производительность 37

синтаксис 117

асинхронное индексирование 31

базовый поиск 109

библиотеки фильтров 68

включение базы данных 83

включение поддержки расширенного текста 80

восстановление

процесс 104

выключение баз данных 84

выключение поддержки расширенного текста 81

деинсталляция DB2 Accessories Suite 69

- текстовый поиск DB2 *(продолжение)*
  - деинсталляция сервера 60
  - дескрипторы файлов 36
  - добавление словаря синонимов 87
  - документы XML 114, 121
  - задача планирования 106
  - запросы 36
  - запуск 81
  - изменение индексов 102
  - изменение информации о сервере 64
  - изменение положения собрания 103
  - изменение характеристик обновления 102
  - индексы
    - двоичные типы данных 90
    - изменение 7, 102
    - инкрементные изменения 12, 99
    - оптимизация 30, 31
    - отбрасывание 104
    - параметры обновления, зависящие от индекса 34
    - планирование 30
    - поиск 108
    - положение 33
    - производительность 30
    - создание 7, 88
    - создание (двоичные типы данных) 90
    - создание (неподдерживаемые типы данных) 91
    - специальные символы 113
  - инкрементные изменения индексов 99
  - инструмент конфигурирования 63
  - инструменты командной строки 79
  - интегрированный сервер 4
  - команда ALTER INDEX 138
  - команда CLEAR COMMAND LOCKS 143
  - команда CLEAR EVENTS FOR TEXT 145
  - команда CREATE INDEX 147
  - команда DISABLE DATABASE FOR TEXT 154
  - команда DROP INDEX 156
  - команда ENABLE DATABASE FOR TEXT 158
  - команда HELP 159
  - команда RESET PENDING 161
  - команда START FOR TEXT 162
  - команда STOP FOR TEXT 164
  - команда UPDATE INDEX 165
  - команды
    - ALTER INDEX 138
    - CLEANUP FOR TEXT 143
    - CLEAR COMMAND LOCKS 143
    - CLEAR EVENTS FOR TEXT 145
    - CREATE INDEX 147
    - DISABLE DATABASE FOR TEXT 154
    - DROP INDEX 156
    - ENABLE DATABASE FOR TEXT 158
    - HELP 159
    - RESET PENDING 161
    - START FOR TEXT 162
    - STOP FOR TEXT 164
    - UPDATE INDEX 165
  - команды управления 96, 137
  - контекстный поиск 110
  - конфигурация сервера 26
  - конфигурирование
    - инструмент конфигурирования 63
    - мастер по установке DB2 48
    - отдельный сервер 58, 59, 66
    - способы 61
    - файл ответов 48
  - конфигурирование синтаксического анализа 39
- текстовый поиск DB2 *(продолжение)*
  - лингвистическая обработка 14
  - локали 38
  - максимальный размер кучи 27
  - менеджер индексов 24
  - множественные предикаты 37
  - модуль Net Search Extender
    - сравнение 219
  - морфологическая индексация 91, 93
  - настройка конфигурации 26
  - неподдерживаемые типы данных 91
  - обзор 1, 3
  - обзор защиты 22, 24
  - обновление 71, 74, 76
  - обновление от имени рядового пользователя (не root) 74
  - обновление текстового индекса 98
  - остановка служб экземпляра 82
  - отбрасывание индексов 104
  - отдельный сервер
    - внедрение 4
    - конфигурирование 66
  - очистка событий индекса текстового поиска 101
  - переконфигурирование 61, 63
  - планирование мощности и оптимизация 25
  - повышение производительности поиска 126
  - поддерживаемые кодовые страницы 20
  - поиск
    - специальные символы 111
    - функция SCORE 116
  - поиск неполных соответствий 109
  - полномочия
    - владелец экземпляра 23
    - пользователь, выполняющий запросы текстового поиска 24
    - роли 23
  - положение собрания 103
  - потоки индексирования 28
  - потребление дискового пространства 32
  - потребление памяти кучей 27
  - производительность 30, 37
  - просмотр состояния индекса 103
  - Пространства имен XML 40
  - размер памяти очереди 29
  - расширенный текст
    - DB2 Accessories Suite 68
    - активизация 80
    - обзор 18
  - резервное копирование 104
  - роли
    - владелец экземпляра 23
    - пользователь, выполняющий запросы 24
  - роли пользователей 23
  - словари синонимов
    - добавление 87
    - удаление 87
  - собрания текстового поиска
    - идентификация бесхозных 84
    - удаление бесхозных 84
  - специальные символы
    - восточные языки 114
    - примыкающие к термам запроса 113
  - среда многораздельной базы данных 10
  - столбцов XML 132
  - сценарий 15
  - таблицы журналов 88
  - таблицы событий
    - обзор 88

- текстовый поиск DB2 *(продолжение)*
  - таблицы событий *(продолжение)*
    - удаление событий 101
  - типы данных
    - преобразование неподдерживаемых 19
  - требования к аппаратному обеспечению 46
  - требования к программному обеспечению 46
  - триггеры 31, 88
  - удаление словаря синонимов 87
  - управляющие подпрограммы 97, 169
  - управляющие производные таблицы
    - SYSIBMTS.TSCOLLECTIONNAMES 215
    - SYSIBMTS.TSCONFIGURATION 214
    - SYSIBMTS.TSDEFAULTS 209
    - SYSIBMTS.TSEVENT 215
    - SYSIBMTS.TSINDEXES 212
    - SYSIBMTS.TSLOCKS 210
    - SYSIBMTS.TSSERVERS 211
    - SYSIBMTS.TSSTAGING 216
  - промежуточная таблица 216
  - таблица журнала 216
  - таблица событий 215
  - уровня базы данных 209, 210
  - уровня индекса 209, 212, 214, 215, 216
- усечение документов 21
- установка
  - библиотеки фильтров DB2 Accessories Suite 68
  - команда db2\_install 49
  - мастер по установке DB2 48
  - отдельный сервер 58, 59
  - требования к месту на диске 58
  - файл ответов 48
- форматы документов
  - поддерживаемые 19
  - преобразование неподдерживаемых 19
- функции поиска 107
- функция CONTAINS 107, 127
- функция SCORE 38, 107
- Функция xmlcolumn-contains 107
- эскейп-символы для специальных символов 111
- языки программирования 20, 38

Текстовый поиск DB2 (DB2 Text Search)

- конфигурирование
  - обзор 43
- установка
  - обзор 43
- функция SCORE 130

## У

- установка
  - без вывода сообщений
    - Linux 57
    - UNIX 57
    - Windows 56
- установка без вывода сообщений
  - Linux 57
  - UNIX 57
  - Windows 56
- устранение неисправностей
  - оперативная информация 258
  - учебные материалы 258
- учебные материалы
  - pureXML 258
  - диагностика ошибок 258
  - список 258
  - устранение неисправностей 258

## Ф

- файл служб
  - обновление для связи TCP/IP 67
- файлы ответов
  - установка
    - Linux 57
    - UNIX 57
    - Windows 56
- функции
  - хранимые процедуры
    - SYSTS\_ALTER 171
    - SYSTS\_CLEANUP 177
    - SYSTS\_CLEAR\_COMMANDLOCKS 178
    - SYSTS\_CLEAR\_EVENTS 180
    - SYSTS\_DROP 197
    - SYSTS\_ENABLE 199
    - SYSTS\_UPDATE 201
    - SYSTS\_UPGRADE\_CATALOG 204
    - SYSTS\_UPGRADE\_INDEX 206
- функции XQuery
  - xmlcolumn-contains 132
- функция SCORE
  - поиск в индексах текстового поиска 130
- функция xmlcolumn-contains 132

## Х

- Хранимая процедура SYSTS\_ALTER 171
- хранимая процедура SYSTS\_CLEANUP 177
- Хранимая процедура SYSTS\_CLEAR\_COMMANDLOCKS 178
- Хранимая процедура SYSTS\_CLEAR\_EVENTS 180
- Хранимая процедура SYSTS\_DROP 197
- Хранимая процедура SYSTS\_ENABLE 199
- Хранимая процедура SYSTS\_UPDATE 201
- хранимая процедура SYSTS\_UPGRADE\_CATALOG 204
- Хранимая процедура SYSTS\_UPGRADE\_INDEX 206





Напечатано в Дании

SC89-1259-01



Spine information:

IBM DB2 10.1 для Linux, UNIX и Windows

Руководство по текстовому поиску

