

**IBM DB2 10.1
for Linux, UNIX, and Windows**

**Net Search Extender
管理およびユーザズ・ガイド
2013 年 1 月更新版**



**IBM DB2 10.1
for Linux, UNIX, and Windows**

**Net Search Extender
管理およびユーザズ・ガイド
2013 年 1 月更新版**



ご注意

本書および本書で紹介する製品をご使用になる前に、327 ページの『付録 B. 特記事項』に記載されている情報をお読みください。

本書には、IBM の専有情報が含まれています。その情報は、使用許諾条件に基づき提供され、著作権により保護されています。本書に記載される情報には、いかなる製品の保証も含まれていません。また、本書で提供されるいかなる記述も、製品保証として解釈すべきではありません。

IBM 資料は、オンラインでご注文いただくことも、ご自分の国または地域の IBM 担当員を通してお求めいただくこともできます。

- オンラインで資料を注文するには、IBM Publications Center (<http://www.ibm.com/shop/publications/order>) をご利用ください。
- ご自分の国または地域の IBM 担当員を見つけるには、IBM Directory of Worldwide Contacts (<http://www.ibm.com/planetwide/>) をお調べください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

お客様の環境によっては、資料中の円記号がバックslashと表示されたり、バックslashが円記号と表示されたりする場合があります。

原典： SC27-3895-01

IBM DB2 10.1 for Linux, UNIX, and Windows

Net Search Extender Administration and User's Guide

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2012.12

© Copyright IBM Corporation 1995, 2013.

目次

第 1 部 Net Search Extender の概要 および概念 1

第 1 章 Net Search Extender の主要な 概念 3

SQL スカラー検索関数の概要	6
ストアード・プロシージャ検索の概要	7
SQL 表値関数の概要	8
追加概念	9
列トランスフォーメーション関数	9
DB2 Net Search Extender インスタンス・サービス Windows での DB2 Net Search Extender インスタ ンス・サービス	10
外部保管データ	10
表とビューの管理	10
パーティション・データベース・サポート	10
フェデレーテッド・データベースのニックネーム に対する索引	11
ネイティブ XML サポート	12
パーティション表サポート	12
保全性処理に基づく増分更新	13
別個の fenced ユーザーのサポート (Linux および AIX).	14

第 2 章 DB2 Net Search Extender の 主要なフィーチャー 17

第 3 章 db2text コマンドの紹介 19

第 2 部 インストール 21

第 4 章 DB2 クライアント/サーバー環境 での Net Search Extender のインスト ール 23

第 5 章 Net Search Extender のインス トールのシステム要件 25

第 6 章 パーティション DB2 サーバーで の Net Search Extender のインストー ール 27

第 7 章 UNIX での Net Search Extender のインストール 29

第 8 章 Windows でのインストール . . . 31

第 9 章 Net Search Extender のディレ クトリー名とファイル名 33

第 10 章 Outside-In ライブラリーのイン ストール 35

第 11 章 インストール検査 37

UNIX でのインストール検査 37

Windows でのインストール検査 37

第 12 章 Net Search Extender のアン インストール 39

UNIX での Net Search Extender のアンインストール 39

Windows での Net Search Extender のアンインスト
ール 39

第 3 部 高可用性 (HA) のための Net Search Extender (NSE) の構成 . . . 41

第 4 部 DB2 Net Search Extender へのアップグレード 45

第 5 部 32 ビットから 64 ビットへ の DB2 Net Search Extender 索引 マイグレーション・ツール 49

第 6 部 32 ビットから 64 ビットの DB2 Net Search Extender 索引マイ グレーション・ツール (Windows) . . . 53

第 7 部 計画に関する考慮事項 57

第 13 章 Net Search Extender のディ レクトリーのロケーションおよび索引スト レンジ 59

第 14 章 パーティション化された DB2 サーバーに関するリソースの考慮事項 . . . 61

第 15 章 ストアード・プロシージャ検 索のメモリー所要量 63

AIX (64 ビット) での Net Search Extender のメモリ
所要量 63

Windows (32 ビットおよび 64 ビット) での Net
Search Extender ストアード・プロシージャのメモ
リ所要量 63

Solaris (64 ビット) での Net Search Extender のメモリー所要量	64
Linux (32 ビットおよび 64 ビット) での Net Search Extender ストアード・プロシージャのメモリー所要量	65

第 16 章 表、列、および索引の名前の考慮事項	67
---------------------------------	-----------

第 17 章 文書フォーマットおよびサポートされるコード・ページ	69
---	-----------

第 18 章 Net Search Extender 索引ファイルの破損を防ぐ方法	71
--	-----------

第 19 章 Outside-In フィルター・ソフトウェア	73
---------------------------------------	-----------

第 20 章 ユーザー・ロール	75
------------------------	-----------

第 21 章 増分更新のための拡張テキスト保持ステージング・インフラストラクチャー	77
--	-----------

第 8 部 Net Search Extender の管理	79
--------------------------------------	-----------

第 22 章 Net Search Extender インスタンス・サービス	81
---	-----------

第 23 章 Net Search Extender インスタンス・サービスの開始および停止 (コマンド行を使用)	83
--	-----------

第 24 章 Net Search Extender ロッキング・サービス	85
ロッキング・サービスの使用	85
ロック・スナップショットの表示	87

第 25 章 更新サービス	89
----------------------	-----------

第 9 部 開発: テキスト索引の作成と保守	91
-------------------------------	-----------

第 26 章 データベースを使用可能にする	93
------------------------------	-----------

第 27 章 データベースを使用不可にする	95
------------------------------	-----------

第 28 章 テキスト索引の作成	97
バイナリー・データ・タイプのテキスト索引の作成	99
サポートされないデータ・タイプのテキスト索引の作成	99

DB2 レプリケーションで、増分索引更新による、ニックネームのテキスト索引を作成する	100
ストアード・プロシージャ検索で使用できるテキスト索引の作成	102

第 29 章 ビュー上でのテキスト索引の作成	105
-------------------------------	------------

第 30 章 範囲パーティション表でのテキスト索引の作成	107
-------------------------------------	------------

第 31 章 索引作成のためのパフォーマンスに関する考慮事項	109
---------------------------------------	------------

第 10 部 テキスト索引の保守	111
-------------------------	------------

第 32 章 テキスト索引の更新と再編成	113
-----------------------------	------------

第 33 章 テキスト索引の更新	115
-------------------------	------------

第 34 章 テキスト索引の変更	117
-------------------------	------------

第 35 章 索引イベントのクリア	119
--------------------------	------------

第 36 章 テキスト索引の妥当性検査 (Windows、AIX)	121
checknseindex コマンド (Windows、AIX)	122

第 37 章 テキスト索引のドロップ	125
---------------------------	------------

第 38 章 テキスト索引状況の表示	127
---------------------------	------------

第 39 章 テキスト索引のバックアップとリストア	129
----------------------------------	------------

第 40 章 /tmp ディレクトリーからのファイルの除去	131
--------------------------------------	------------

第 11 部 テキストを検索する方法	133
---------------------------	------------

第 41 章 SQL スカラー検索関数を使用するテキストの検索	135
照会の発行	135
検索を実行し、検出された一致の数を戻す	135
検索を実行し、検出されたテキスト文書のスコアを戻す	136

第 42 章 SQL 検索指数の指定	137
任意の順序での用語検索	137
ブール演算子 AND と OR を使用する検索	137
ブール演算子 NOT を使用する検索	138
ファジー検索	138
用語の一部の検索 (文字のマスキング)	139
マスク文字を含んでいる用語の検索	139

固定順序での用語検索	140
同じ文または同じ段落内での用語検索	140
構造化文書のセクション内での用語検索	140
シソーラス検索	141
数値属性検索	141
フリー・テキスト検索	142

第 43 章 その他の検索構文の例 143

第 44 章 ストアード・プロシージャ 検索を使用するテキスト検索 145

第 45 章 SQL 表値関数を使用するテキ スト検索 147

HIGHLIGHT 関数の使用 147

第 46 章 複数列の検索 151

第 47 章 外部結合でのテキスト検索の 使用 153

第 48 章 検索時のパフォーマンスに関 する考慮事項 155

第 49 章 ユーザー・シナリオ 157

SQL スカラー検索関数 の場合の簡単な例 157

キャッシュを使用したストアード・プロシージャ
検索 の場合の簡単な例 159

SQL 表値関数の場合の簡単な例 160

第 50 章 検索項目を拡張するためのシ ソーラスの使用 161

シソーラスの構造 161

定義済みシソーラス関係 162

ユーザー独自の関係の定義 163

シソーラスの作成およびコンパイル 163

シソーラス定義ファイルの作成 164

定義ファイルのシソーラス・ディクショナリーへ
のコンパイル 165

シソーラス・サポート 165

シソーラスでサポートされている CCSID 167

シソーラス・ツールが戻すメッセージ 168

第 51 章 Net Search Extender 索引付 け構成 171

トークン化 171

ストップワード 172

ストップワードをサポートする言語 172

構成 173

第 12 部 構造化文書の使用 175

第 52 章 ネイティブな状態で格納され た XML 文書の検索 177

デフォルト文書モデルの使用	177
カスタマイズした文書モデルの使用	178
XQuery サポート	179

第 53 章 構造化文書のサポート 183

文書モデルが構造化文書を記述する方法	183
文書モデルの例	183
文書モデル	184
テキスト・フィールド	185
文書属性	185
デフォルトの文書モデル	186
構造化プレーン・テキスト文書の文書モデルの定義	187
GPP 文書を索引付けするとどうなるか	188
HTML 文書の文書モデルの定義	188
XML 文書の文書モデルの定義	191
Outside-In フィルター文書の文書モデルの定義	195
Outside-In 文書を索引付けするとどうなるか	197

第 54 章 文書モデルの参照情報 199

文書モデル用の DTD	199
ロケーター (XPath) 式のセマンティクス	200
テキスト・フィールドおよび文書属性の制限	202
Outside-In タグ属性値	203

第 13 部 リファレンス 205

第 55 章 インスタンス所有者用の管理 コマンド 207

CONTROL コマンド 207

START コマンド 209

STOP コマンド 211

第 56 章 データベース管理者用の管理 コマンド 213

ENABLE DATABASE コマンド 213

DISABLE DATABASE コマンド 215

DB2EXTHL コマンド 217

第 57 章 テキスト表所有者用の管理コ マンド 219

ACTIVATE CACHE コマンド 220

ALTER INDEX コマンド 221

CLEAR EVENTS コマンド 225

CREATE INDEX コマンド 227

DEACTIVATE CACHE コマンド 242

DROP INDEX コマンド 243

DB2EXTTH コマンド 245

RESET PENDING コマンド 246

UPDATE INDEX コマンド 247

HELP コマンド 251

COPYRIGHT コマンド 252

第 58 章 UNIX における Net Search Extender インストールおよびアンイン ストール・コマンド・リファレンス 253

db2nse_install コマンド	253
db2nse_deinstall コマンド	254
db2nsecls コマンド	255

第 59 章 検索指数の構文.	257
検索指数の構文.	257
検索パラメーター.	260

第 60 章 SQL スカラー検索関数および SQL 表値関数	265
CONTAINS スカラー関数	265
NUMBEROFMATCHES スカラー関数	266
SCORE スカラー関数.	266
DB2EXT.TEXTSEARCH コマンド.	266
DB2EXT.HIGHLIGHT.	270

第 61 章 ストアード・プロシージャー 検索関数.	275
ストアード・プロシージャー検索 用の DB2EXT.TEXTSEARCH	275

第 62 章 Windows システム・エラー	279
--------------------------------	------------

第 63 章 Net Search Extender インフォメーション・カタログ	283
データベース・レベル情報のビュー.	283
索引レベル情報のビュー.	286
db2ext.textindexes ビュー.	287
db2ext.indexconfiguration ビュー	288
db2ext.textindexformats ビュー.	289
テキスト索引の表ビュー.	290
イベント・ビュー.	290
ログ表、ビュー、およびニックネーム	291

第 64 章 テキスト検索エンジン理由コード	293
---	------------

第 14 部 トラブルシューティング 301

第 65 章 障害のトレース.	303
--------------------------------	------------

第 66 章 正しい Net Search Extender コマンドを使用せずに DB2 オブジェクトをドロップする	305
表のドロップ	305
データベースのドロップ.	305

第 67 章 Windows でのインストール戻りコード.	307
--	------------

第 68 章 ヒント	309
-----------------------------	------------

第 15 部 付録 313

付録 A. DB2 技術情報の概説	315
DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式)	316
コマンド行プロセッサから SQL 状態ヘルプを表示する.	318
異なるバージョンの DB2 インフォメーション・センターへのアクセス	319
コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの更新	319
コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの手動更新.	321
DB2 チュートリアル.	323
DB2 トラブルシューティング情報	324
ご利用条件	324

付録 B. 特記事項	327
-----------------------------	------------

索引	331
---------------------	------------

第 1 部 Net Search Extender の概要および概念

第 1 章 Net Search Extender の主要な概念

Net Search Extender は、DB2[®] データベースやその他のデータベース、およびファイル・システムに保管されているフルテキスト文書を SQL 照会を使用して検索するための方法をユーザーとアプリケーション・プログラマーに提供します。

重要: Net Search Extender は非推奨になりました。使用は推奨されません。また将来のリリースで除去される可能性があります。DB2 データベースに保管されているフルテキスト文書を SQL ステートメントおよび XQuery ステートメントを使用して検索する高速かつ用途の広い方法として、DB2 Text Search を使用してください。詳しくは、Net Search Extender から DB2 Text Search へのマイグレーションに関するトピックを参照してください。

Net Search Extender の機能を十分に理解するには、主要な用語、および使用可能なさまざまなオプションを理解する必要があります。DB2 データベースの概念および用語の基礎的な知識も必要です。

基本的に、Net Search Extender は、データベース表の列に含まれているテキスト文書の検索を行います。

テキスト文書は、一意に識別可能でなければなりません。この目的のために、Net Search Extender は表の主キーを使用します。

文書はさまざまなフォーマットで保管できます。フォーマットは、構造化されていないプレーン・テキスト、HTML、XML などの構造化されたテキスト、または PDF、Microsoft Office 文書フォーマットなどの独自文書フォーマットのいずれかを含みます。独自文書フォーマットの場合は、追加のフィルター・ソフトウェアが必要であり、別途ライセンス交付を受けなければならない可能性があります。

膨大な時間を必要とするテキスト文書の順次検索を照会時に実行する代わりに、Net Search Extender は、文書の効率的な検索を可能にするテキスト索引を作成します。

テキスト索引は、テキスト文書から抽出された重要な用語で構成されます。

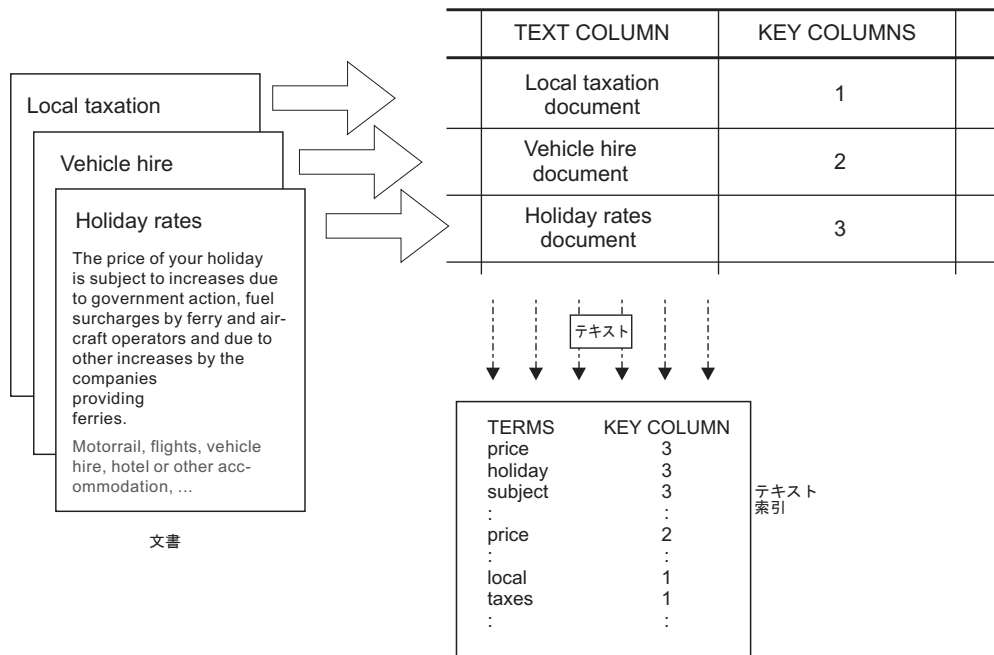


図1. テキスト索引の作成

テキスト索引作成は、索引のロケーションのような、索引のプロパティの定義および宣言を行う処理です。作成終了時点のテキスト索引にはまだデータが含まれていません。索引更新は、テキスト索引に条件および文書に関するデータを追加する処理です。最初の索引更新で、テキスト列のすべてのテキスト文書に関する情報が索引に追加されます。最初の更新は、初期更新と呼ばれます。

検索にテキスト索引を使用することによって、表とテキスト索引との間の同期を考慮する必要がでてきます。理由は、テキスト文書の追加、削除、および更新などの表に対する後からの変更が、テキスト索引に反映される必要があるためです。これらの変更は、増分更新でテキスト索引に適用されます。

Net Search Extender は、テキスト索引をソース表と同期する 2 つのオプションをサポートします。Net Search Extender での基本的な同期処理は、新規の文書、変更や削除された文書に関する情報を自動的にログ表に保管するトリガーに基づいて行われます。それぞれのテキスト索引ごとに 1 つのログ表が存在します。

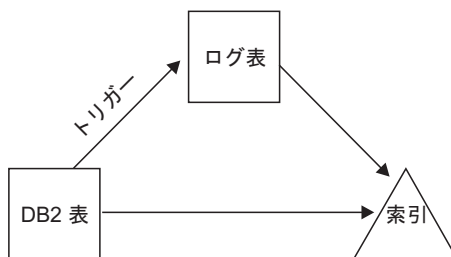


図2. トリガーによる増分更新処理

基本オプションは、トリガーのみに基づいて行われます。そのため、トリガーによって認識されない更新は無視されます。例えば、LOAD コマンドを使用してデータを

ロードする場合や、レンジ・パーティション表に対して範囲をアタッチ、またはデタッチする場合です。拡張同期オプションを使用すると、そのような変更も取り込むことができます。これは、新規文書および削除された文書についての情報を保管するためのテキスト保持ステージング表を追加することによって、保全性処理により可能になります。一方、ログ表には、トリガーによって取り込まれた変更済み文書が保管されます。

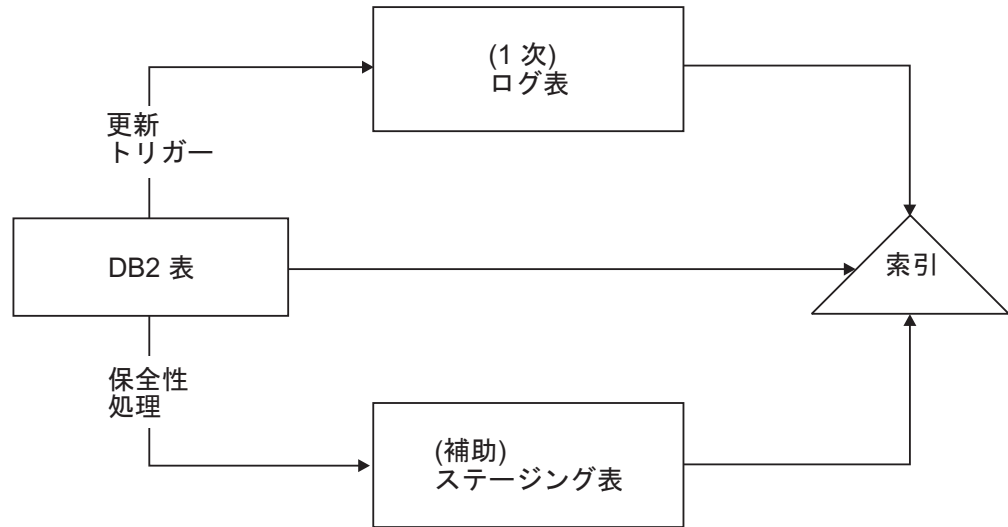


図3. トリガーと保全性処理による増分更新

手動または自動オプションを使用して、テキスト索引を更新できます。自動オプションでは、指定された日付および時刻による更新スケジュールを使用します。

これらのどちらのオプションを使用しても、テキスト文書の更新、削除、および挿入を行うトランザクションの範囲内で、テキスト索引が同期化されることはありません。Net Search Extender の非同期テキスト索引作成を使用すると、パフォーマンスおよび並行性が向上します。別々のトランザクション内で、索引の非常に小さな部分のコピーに対して更新が行われます。オリジナルに代わってコピーが置かれる非常に短い間だけ、索引に読み取りアクセスのロックがかけられるに過ぎません。検索操作ではこのことは意識されません。詳しくは、81 ページの『第 22 章 Net Search Extender インスタンス・サービス』を参照してください。

テキスト索引は、索引ファイルのロケーションおよび自動更新プロパティなどの、特定のプロパティを持っています。必要なら、プロパティの一部を変更できます。これは索引の変更と呼ばれます。索引を変更しても索引データは変更されません。

そのようなプロパティの 1 つに、ORDER BY 句で表の列のテキスト索引を事前ソートするかどうかというものがあります。このような場合、初期更新を行うと、指定した順序でテキスト文書に索引が付けられて、検索結果がこの順序で戻されます。

例えば、価格で事前ソートされた本の要約を指定できます。リレーショナル・データベース・システムに関する本で一番安いものを探す場合は、テキスト検索を制限して一番安い最初の数冊の本だけを戻すようにすることができます。ただし、事前

ソートされた索引がないと、すべての本を検索し、これらの本と一番安い本とを結合する必要があります。この場合は、操作に多くの時間がかかります。

Net Search Extender では、テキスト列ごとに事前ソート済み索引を複数もつことができます。例えば、1 つ目の索引で、出版の日付に従って本の事前ソートを行い、2 つ目の索引で、価格に従って本の事前ソートを行うことができます。

通常、テキスト索引を作成した後の最初の更新が初期更新になり、それ以降は増分更新になります。ただし、事前ソート済み索引を使用して作業を行う場合は、更新の際に順序を維持する必要があります。RECREATE INDEX ON UPDATE オプションを使用すると、この処理を行うことができます。このオプションでは、更新が行われるたびに索引がすべて再作成されます。

テキスト索引の更新が終わると、以下のいずれかのオプションを使用して、検索を行うことができます。

- SQL スカラー検索関数
- ストアド・プロシージャ検索
- SQL 表値関数

検索オプションにはいろいろな操作特性があります。以下のセクションでこれらのオプションを説明します。

SQL スカラー検索関数の概要

Net Search Extender には、SQL 内でシームレスに統合される 3 つのスカラー・テキスト検索関数 (CONTAINS、NUMBEROFMATCHES、および SCORE) があります。

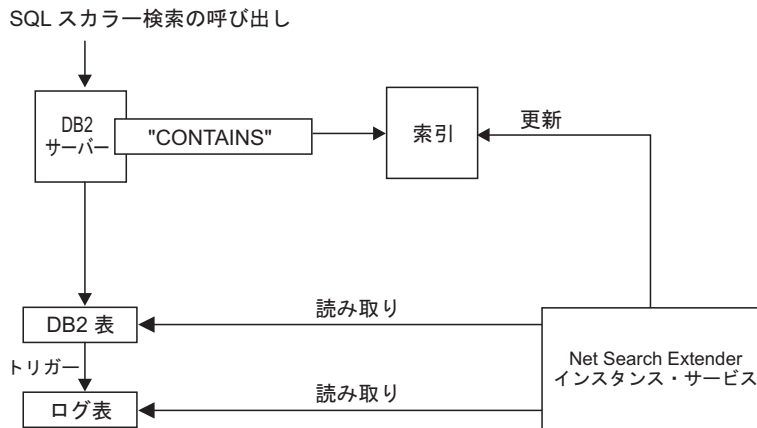


図 4. 検索での SQL スカラー検索関数の使用

検索関数は、SQL 照会内で標準の SQL 式を使用する場合と同じ場所で使用できます。典型的な照会は以下のとおりです。

```
SELECT * FROM books WHERE CONTAINS (abstract, 'relational databases') = 1  
AND PRICE <10
```

```
SELECT ISBN, SCORE (abstract, 'relational databases') as SCORE
```

```
from BOOKS
where NUMBEROFMATCHES (abstract, "relational databases")
>5 AND PRICE <10
order by SCORE
```

この例の SQL スカラー関数は、テキスト文書が、指定されたテキスト検索条件にどの程度一致したのかを示す指標を戻します。SQL 照会の SELECT 段階で、ユーザーに戻す情報が決定されます。

SQL スカラー検索関数を、デフォルトの検索方式として使用してください。これらの検索関数は、ほとんどの状況に適合するはずですが、テキストの検索式が他の条件と結合している場合には、特にそう言えます。

DB2 のオプティマイザーは、CONTAINS 述部に一致するテキスト文書の個数を予測し、代替りの別のアクセス・プランで、どの程度の時間が必要になるかを認識できます。オプティマイザーは、最短の時間のアクセス・プランを選択します。

ストアード・プロシージャ検索の概要

ストアード・プロシージャ検索は主に、テキスト検索のみの照会を行うハイパフォーマンスでスケーラビリティの高いアプリケーションで使用します。テキスト検索のみの照会とは、テキスト検索の結果とその他の複雑な SQL 条件の結果とを結合する必要のない検索です。

検索結果をエンド・ユーザーに提示する場合は、通常、ユーザー表に対する結合操作を後に続けて、検索関数そのものを呼び出します。結果データのソートも行う場合があります。この操作は、時間やリソースを多く必要とする場合があります。一方、メモリーに格納され、事前ソートされたデータに対して結合操作を実行することによって、アプリケーションが大量の時間やリソースを必要とするディスク操作を回避できる場合があります。以下のような状態が考えられます。

- ユーザーに提示するデータ・サブセットが少量である
- データのサブセットが事前にわかっている
- 対象とするソート順序が固定されていて、事前にわかっている
- 検索結果のランク済みサブセットが十分である

テキスト索引の作成時に、表またはビューのどの列をエンド・ユーザーに戻すのかを指定する必要があります。データはメイン・メモリーのキャッシュに保管されます。このようにすると、ストアード・プロシージャ検索で、非常に素早く検索結果を戻すことができます。キャッシュは、使用する前にアクティブ化する必要があります。また、対応する非アクティブ化コマンドがあります。

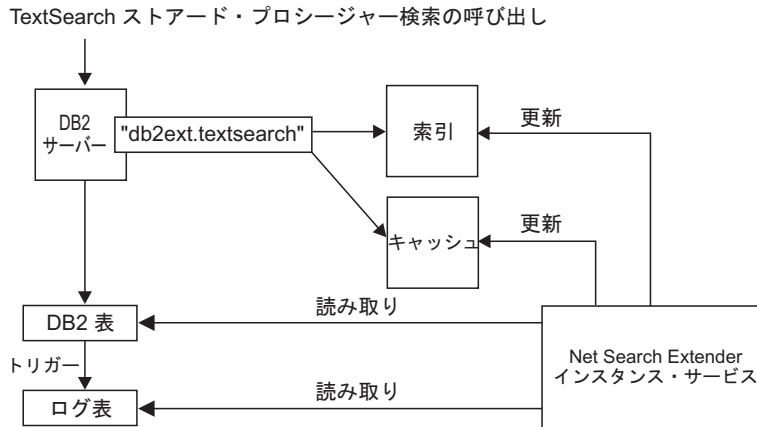


図5. ストアド・プロシージャ検索の使用

ACTIVATE コマンドを使用すると、データは一時キャッシュ (アクティブ化の際に最初から作成される) か、ディスク上に保持される永続キャッシュにロードされます。

検索にストアド・プロシージャを使用するかどうかを決定する場合は、どれだけのメモリが必要なのか、および索引の更新にどれだけの空きメモリが残っていないかなど、メモリーの計算を注意して行う必要があります。詳細は、63 ページの『第 15 章 ストアド・プロシージャ検索のメモリー所要量』を参照してください。

ストアド・プロシージャは、ビュー上に作成されたテキスト索引を処理できます。ただし、ビュー上にトリガーを作成できないため、すべての変更が自動的に認識されるわけではありません。変更情報を手動でログ表に追加するか、RECREATE オプションを使用する必要があります。

ストアド・プロシージャ検索は主に、テキスト検索のみの照会を行うハイパフォーマンスでスケーラビリティの高いアプリケーションで使用します。テキスト検索のみの照会とは、テキスト検索の結果とその他の複雑な SQL 条件の結果とを結合する必要のない検索です。

SQL スカラー検索関数との主な機能上の相違点は以下のとおりです。

- ストアド・プロシージャ検索は、任意の SQL 照会には使用できず、定義済みのキャッシュ表に対する照会である。
- ストアド・プロシージャ検索は、ビューの索引を活用することができる。
- ストアド・プロシージャ検索は、列の事前ソートされた複数のテキスト索引を利用できる。

SQL 表値関数の概要

SQL 表値関数は、SQL スカラー検索関数とストアド・プロシージャ検索の中間的な方式です。SQL 表値関数とともに db2ext.highlight 関数を使用することにより、検索結果としてその文書が選ばれた理由に関する情報を入手することができます。

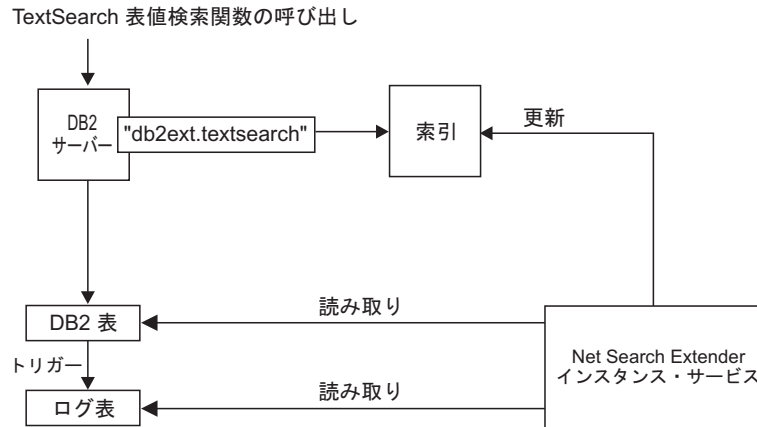


図 6. 検索での SQL 表値関数の使用

ストアード・プロシージャ検索との主な機能上の相違点は以下のとおりです。

- キャッシュが必要ない (キャッシュが利用されない)。
- 任意の SQL ステートメントに表値関数を使用できる。
- キャッシュ表の内容の事前記憶に、大容量のメモリが必要ない。

SQL スカラー検索関数との主な機能上の相違点は以下のとおりです。

- SQL 表値関数はビューの索引を利用できる。

SQL 表値関数は、通常 SQL スカラー関数を使用する場合で、ビューのテキスト索引も使用する場合に使用してください。

追加概念

列トランスフォーメーション関数

ユーザー独自の関数を使用して、サポートされていないフォーマットまたはデータ・タイプを、サポートされている形式またはデータ・タイプに変換することができます。ユーザー定義関数 (UDF) を指定することによって、オリジナルのテキスト文書を入力として入手できます。

ユーザー独自の関数を使用して、サポートされていないフォーマットまたはデータ・タイプを、サポートされている形式またはデータ・タイプに変換することができます。ユーザー定義関数 (UDF) を指定することによって、オリジナルのテキスト文書を入力として入手できます。UDF からの出力は、索引作成の際に処理できるサポートされたフォーマットになります。

直接サポートされていない外部データ・ストアに保管されている文書の索引作成にも、このフィーチャーを使用できます。この場合、DB2 列には文書参照が含まれているため、関数は、適切な文書参照をもった文書の内容を戻します。

DB2 Net Search Extender インスタンス・サービス

Net Search Extender インスタンス・サービスは、索引固有のロッキング・サービスおよびテキスト索引の更新サービス (自動および手動の両方) を行います。

Net Search Extender インスタンス・サービスは、索引固有のロッキング・サービスおよびテキスト索引の更新サービス（自動および手動の両方）を行います。

Windows での DB2 Net Search Extender インスタンス・サービス

新規 DB2 インスタンスが作成されると、最初のパーティションに対して DB2EXT サービスが自動的に作成されます。

db2ncrt コマンドを使用して新規パーティションが追加されると、さらに DB2EXT サービスが追加されます。**db2start add dbpartitionnum** コマンドが実行される時にも、DB2EXT サービスが追加されます。同じように、**db2ndrop** や **db2nchg** も DB2EXT サービスを適切にドロップまたは変更します。

外部保管データ

外部に保管されているテキスト文書はサポートされますが、テキスト索引を作成するには、DB2 ニックネーム表とともに使用しなければなりません。

ほとんどの場合、テキスト索引が作成されるデータは、CLOB または VARCHAR 型などの、ネイティブ DB2 表の列内に保管されています。

ただし、その他のデータベースなどの外部に保管されるテキスト文書もサポートされています。その他のデータベースに保管される文書の場合、テキスト索引を作成する DB2 ニックネーム表を使用してください。

サポートされていない外部データ・ストアに保管されているデータに対して、列トランスフォーメーション関数を使用することもできます。

表とビューの管理

Net Search Extender の表およびビューにより、テキスト索引とプロパティ情報が提供されます。

Net Search Extender には複数の使用可能な表およびビューがあります。それらのビューは、テキスト索引およびそれらのプロパティに関する情報を提供します。

パーティション・データベース・サポート

パーティション・データベース環境において NSE を使用可能にし、管理するときは、データベースに影響を及ぼしかねないいくつかの点について考慮してください。

パーティション・データベース環境において NSE を使用可能にし、管理するときは、次の点を考慮してください。

- DB2 の文書で説明されているとおりに、DB2 がセットアップされていることを確認してください。NFS マウントにルート・アクセスおよび `setuid` が構成されている必要があります。
- **db2text start** の処理中に問題が発生した場合は、使用可能なパーティションのうち、どのパーティションが影響を受けるかを示す詳細メッセージは戻されません。**db2text start** をもう一度発行すると、システムは各パーティション上でサ

ービスを開始しようとしています。**db2text start** コマンドが正常に終了すると、次のメッセージが表示されます。「CTE0185 更新およびロック・サービスは既にアクティブになっています。」

- fenced ユーザー ID は、インスタンス所有者 ID と同じである必要があります。
- NSE インスタンス・サービスの実行中は、db2nodes.cfg に新しいパーティション番号を挿入したり、既存のパーティション番号を削除したりすることはできません。これは、db2nodes.cfg に変更をもたらす可能性があるすべてのコマンドに適用されます。
- Windows プラットフォームでは、パーティション・データベース環境で NSE を使用している間、db2nodes.cfg は同じホストに対して複数の IP アドレスおよびホスト名を使用すべきではありません。
- 予期しない結果やエラー・メッセージを避けるため、データ再分散を実行する前に既存のテキスト索引をドロップしてください。初めにそのデータベース内のテキスト索引をドロップしないで再分散操作を実行した場合でも、再分散操作が完了した後に、テキスト索引をドロップすることは可能です。

Net Search Extender の検索機能は、パーティション・データベース環境サポートを次の方法で使用します。

- ストアード・プロシージャ検索および SQL 表値関数は、パーティション・データベース上のコーディネーター・パーティションに対してローカルである表でのみ機能します。
- SQL スカラー検索関数 (CONTAINS、NUMBEROFMATCHES、および SCORE) は、まれなケースを除いて、複数のパーティションにわたる表で使用することができます。

パーティション・データベース環境で検索を行う場合、以下の点について考慮してください。

- 検索中に RESULT LIMIT がパーティションごとに評価されます。これは、RESULT LIMIT に 3 を指定して、パーティションを 4 個使用する場合、12 個までの結果が戻る可能性があることを意味します。
- SCORE 値は、照会が複数のパーティションにアクセスする場合でも、単一パーティションにおけるすべての文書の SCORE 値と比較した際の文書の関連性を反映します。

フェデレーテッド・データベースのニックネームに対する索引

フェデレーテッド・データベース内のニックネームにはテキスト索引を作成することができます。これは、リモート・データベース内の表を指し示すものとなります。

この場合、ログ表が担う役割 (増分索引更新の場合) は、通常の表の索引の場合と異なります。通常の表とは異なり、DB2 トリガーをニックネームに対して作成することはできません。したがって、トリガーを使用して文書の変更情報をログ表に挿入することはできません。また、拡張テキスト保持ステージング・インフラストラクチャーを使用して変更を取り込むこともできません。そのため、増分更新でニックネームの索引を作成するには、以下に挙げる 2 つの異なる方法があります。

- ログ表をフェデレーテッド・データベース内でローカルに作成し、アプリケーションの側で、ニックネームに関する正しい変更情報が確実にログ表に含まれるよ

うにする方法。DB2 ビューの場合、これは増分索引更新と似ています。このオプションはデフォルト・オプションです。

- ニックネームによって参照される表に加えられた変更を、いわゆる「変更データ表 (CD 表: DB2 リモート・データベースの場合)」、または「整合変更データ表 (CCD 表: DB2 以外のリレーショナル・データベースの場合)」にキャプチャーするよう、DB2 レプリケーションをセットアップしておく方法。そうすれば、DB2 Net Search Extender は、ニックネームの索引用にログ表を作成する代わりに、CD または CCD 表を使用できます。この場合、**DB2TEXT CREATE INDEX** コマンドにキャプチャー表の特性を指定する必要があります。

ネイティブ XML サポート

コード・ページ UTF-8 を使用したデータベースでは SQL XML データ・タイプを完全にサポートしているため、データベースにネイティブな状態で格納された XML 文書に対してすべての Net Search Extender 検索関数を使用できます。UTF-8 以外を使用するデータベースでの SQL XML データ・タイプに対するテキスト検索はサポートされていないため、結果が返されない場合がある点に注意してください。

セクションを使用することによる XML 文書に対する構造化されたテキスト検索 (セクションの検索に関する詳細については 260 ページの『検索パラメーター』を参照してください) は、検索結果を強力な XQuery で処理することによって拡張できます。DB2 データベース・サーバーの XQuery 言語サポートを使用して Net Search Extender のテキスト検索機能を活用し、XML 文書の処理を最適化できます。

db2-fn:sqlquery() XQuery 入力関数でフルテキスト検索を使用すると XML 文書内を検索してから、結果の XML 文書を XQuery を使用して処理できます。

```
FOR $dept in db2-fn:sqlquery('select Department from MyTable
  where contains(Department, 'sections(/dept/employee/resume) "DB2 XML" '
    = 1')/dept
RETURN $dept/employee/name
```

この例では、列「Department」は「XML」データ・タイプです。詳細については、175 ページの『第 12 部 構造化文書の使用』を参照してください。

XML データの照会方法を示すサンプルが使用できます。 *sqllib/samples/* *extenders/db2ext* を参照してください。データベースにデータを設定して、索引の作成および更新を行うには *xmlsample database* を呼び出してください。データベースに接続したら、**db2 -tvf xmlsearch** を発行することによってデータを検索できます。

パーティション表サポート

範囲パーティション表、または単一パーティションおよび複数パーティション・データベース環境でマルチディメンション・クラスタリング・フィーチャーを使用する表に、テキスト索引を作成することができます。

テキスト索引は、あらゆるパーティション・フィーチャーの組み合わせでサポートされます。

注: バージョン 9.7 では、テキスト索引は、複数のデータベース・パーティションにまたがる表のパーティション化に応じてパーティション化されます。範囲パーテ

ィションまたはマルチディメンション・クラスタリングのような他のパーティション・フィーチャーは、テキスト索引のパーティション化に影響を与えません。

健全性処理に基づく増分更新

Net Search Extender **CREATE INDEX** コマンドの **AUXLOG** オプションを使用することによって、テキスト索引に補助ログ・インフラストラクチャー (テキスト保持ステージング表) を使用するかどうかを制御することができます。

この補助ステージング表は、健全性処理による新規文書および削除された文書に関する情報を取り込みます。一方、文書更新は、**INDEX CREATE** コマンドで参照される基本表列の更新トリガーによって取り込まれます。

このオプションの使用には、以下の制約事項が適用されます。

- テキスト索引を作成した対象が、ビューまたはニックネームではなく、基本表でなければなりません。
- **CACHE** オプションは指定できません。
- **RECREATE INDEX ON UPDATE** オプションは使用できません。
- レプリケーションを使用して更新を制御することはできません。

デフォルトでは、この構成オプションは、範囲パーティション表に対して **ON**、非パーティション表に対して **OFF** に設定されています。

健全性処理によるテキスト索引の増分更新についての変更を取り込む場合、基本表に対してデータベース操作を行った後に、追加の管理用タスクを行わなければならない場合があります。基本表またはその従属表の健全性を設定するためには、コマンドについてポストプロセッシング・タスクを実行するか、テキスト索引更新についてプリプロセッシング・タスクを実行しなければならない場合があります。

```
db2 "create table test.simple (pk integer not null primary key,
comment varchar(48))"
```

```
db2 "insert into test.simple values (1, 'blue and red')"
```

```
db2text "create index test.simpleix for text on test.simple(comment) index
configuration(auxlog on) connect to mydb"
```

```
db2text "update index test.simpleix for text connect to mydb"
```

```
db2 "load from loaddata4.sql of del insert into test.simple"
```

ロード操作の後、基本表はブロックされます。例えば、選択操作の結果は、

「SQL0668N 理由コード 1 のため、表 TEST.SIMPLE に対する操作は許可されません。」(SQLSTATE=57016) になります。

ステージング表はアクセス可能ですが、変更されたデータに関する情報はまだ含まれていません。

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

これは、「SQL3601W ステートメントにより 1 つ以上の表が自動的に SET INTEGRITY ペンディング状態になりました。」(SQLSTATE=01586) を戻します。

この時点で、ステージング表はブロックされ、基本表に対する変更操作はリジェクトされます。

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


これは、「DB21034E コマンドが、有効なコマンド行プロセッサ・コマンドでないため、SQL ステートメントとして処理されました。SQL 処理中に、次のエラーが返されました。SQL0668N 理由コード "1" のため、表 "SYSIBMTS"."SYSTSAUXLOG_IX114555" に対する操作は許可されません。」(SQLSTATE=57016) を戻します。

```
db2text "reset pending for table test.simple for text connect to mydb"
```

```
db2text "update index test.simpleix for text connect to mydb"
```

詳しくは、57 ページの『第 7 部 計画に関する考慮事項』を参照してください。

別個の fenced ユーザーのサポート (Linux および AIX)

Net Search Extender では、DB2 データベース・サーバーによって使用されるアドレス (メモリー) スペースの外部にあるユーザー定義関数 (UDF) およびストアド・プロシージャを実行するために fenced ユーザー・アカウントが必要です。

別個の fenced ユーザー ID が Net Search Extender でサポートされるようになりました。この ID は DB2 インスタンス所有者とは異なります。インスタンス所有者と fenced ユーザーの両方に対して、共通のグループが 2 次グループとして追加されます。

fenced ユーザー ID を使用する場合、以下の点を考慮に入れてください。

- インスタンス所有者と fenced ユーザーが異なる場合、Net Search Extender の管理コマンドが表所有者または表に対するコントロール特権を持つユーザーによって実行されると、そのコマンドは失敗します。例えば、インスタンス所有者と fenced ユーザーが異なり、共通の 2 次グループを共有していない場合、**db2text START** を実行すると次のエラー・メッセージが返されます。

```
CTE0312E fenced ユーザーおよびインスタンス所有者に対する共通 2 次グループが存在しません。
```

共通の 1 次グループを使用することはできますが、セキュリティ上の理由で共通の 2 次グループを使用することが推奨されています。システム、root、および管理グループは共通 2 次グループとして使用しないでください。そのグループによる NSE ファイル所有権のセキュリティ・リスクが生じるためです。この目的のためには、新しく 2 次グループを作成する必要があります。

注: インスタンス所有者および fenced ユーザーの共通 2 次グループがゼロに設定されていると、NSE サービスは開始されません。

- fenced ユーザーは索引ファイルとシソーラス・ファイルへのアクセス権限を持つ必要があります。

注: 新しい 2 次グループはこれを実現するのに役立ちます。umask 制限が、fenced ユーザーに対してグループ読み取り権限と書き込み権限を許可するようになっていることを確認してください。umask 値は 0002 に設定する必要があります。

- 以下の管理コマンドは、インスタンス所有者によってのみ発行できます。
 - CREATE INDEX
 - UPDATE INDEX
 - ALTER INDEX

- DROP INDEX
 - ACTIVATE CACHE
 - DEACTIVATE CACHE
 - RESET PENDING
 - CLEAR EVENTS
 - DB2EXTTH
 - HELP
 - COPYRIGHT
- NSE 索引ディレクトリーには、NSE 索引ファイル以外の外部ファイルは存在しません。
 - HPUX の場合、シソーラス検索を機能させるには、シソーラス定義ファイルを再コンパイルする必要があります。
 - **NUMBER_DOCS** ルーチンおよび **REORG_SUGGESTED** ルーチンは、パーティション・データベース環境ではサポートされません。

例

インスタンス所有者の ID が db2inst1 で、fencedユーザーの ID が db2fenc1 である場合に、「video」という新しいグループを作成します。このグループをインスタンス所有者と fenced ユーザーの両方に対する 2 次グループにします。

これは、以下のステップを使用して検証することができます。

```
>id db2inst1
uid=44049(db2inst1) gid=204(search) groups=33(video)
```

```
>id db2fenc1
uid=44048(db2fenc1) gid=100(users) groups=33(video)
```

第 2 章 DB2 Net Search Extender の主要なフィーチャー

Net Search Extender バージョン 9.7 には、索引付け、検索および検索結果に関する主要なフィーチャーが備わっています。

Net Search Extender バージョン 9.7 には、次の主要なフィーチャーが備わっています。

- 索引付け
 - 大規模データ・ボリュームの高速索引付け
 - 索引の動的更新
 - (オプション) 索引作成時にメイン・メモリーに表の列を保管することによる、検索時の過度の物理読み取り操作の回避
 - HTML、および XML など構造化されたテキスト・フォーマットのサポート
 - サード・パーティーのフィルター・ソフトウェア「Outside-In」のサポート
 - ニックネーム表サポート
 - 事前ソートされたテキスト索引のサポート
 - パーティション・データベース・サポート
 - ネイティブ XML サポート
 - 範囲パーティション表とクラスター表 (MDC) のサポート
- 検索
 - ブール演算
 - 同じ文または同じ段落内でのワードに対する近接検索
 - 検索語と似たスペルを持つワードを検索する「ファジー」検索
 - ワード全体および単一文字に対する、前部、中間、および後部のマスキングを使用するワイルドカード検索
 - フリー・テキスト検索。特定のテキストを含んでいる文書用に、検索回数自然言語で表現できる。
 - 特定の文書が検索結果として選ばれた理由を示すための強調表示関数
 - シソーラス・サポート
 - 文書内のセクションへの検索の限定
 - 数値属性のサポート
 - 多数の並行ユーザーによる、大量のテキスト文書に対する高速検索
 - `db2-fn:sqlquery()` 関数を使用することによる XQuery 処理との統合
- 検索結果
 - 索引作成時に、検索結果のソート方法を指定できる
 - 大量のデータの中から検索して、結果リストが大量に及ぶものと予想される場合に、検索結果のサブセットを指定できる
 - 検索項目に対して、ヒット・カウントによる制限を設定できる
 - DB2 オプティマイザーと結合した組み込み SQL 関数が、予想される検索結果に従って最適なプランを自動的に選択する機能

制約事項: Net Search Extender では、IBM® DB2 pureScale® Feature をサポートしていません。

第 3 章 db2text コマンドの紹介

このトピックでは、このタスクについての情報を提供し、例を挙げて、db2text コマンドで次に何をすべきかを示します。

このタスクについて

Net Search Extender コマンドの例を次に示します。

```
db2text ENABLE DATABASE FOR TEXT
```

作成と索引保守のすべてのコマンドで、データベース、ユーザー、およびパスワードを指定できます。

```
db2text ... connect TO database USER userID USING password
```

注: db2text コマンドで接続オプションを省略すると、環境変数 **DB2DBDFT** によってデータベースが指定されます。

コマンドのリストを表示するには、次のコマンドを入力します。

```
db2text ?
```

各コマンドの構文を表示するには、次のコマンドを入力します。

```
db2text ? command
```

例

例えば、**CREATE INDEX** コマンドの構文を表示するには、次のコマンドを使用します。

```
db2text ? CREATE INDEX
```

db2text は、コマンドが正常に処理された場合 0 を、コマンドが処理されなかった場合は 1 を戻します。文書エラーがあるにもかかわらず、索引が更新される場合、**db2text** コマンドは警告メッセージとともに 0 を戻します。文書エラーに関する情報は、索引のイベント表にあります。

次のタスク

オペレーティング・システムおよびアクティブなコマンド・シェルに応じて、システムは ?、(、)、*、!、" などの特殊文字を解釈します。したがって、コマンドにこれらの文字が含まれている場合には、引用符またはエスケープ文字を使用してください。

以下に、特殊文字を使用する UNIX コマンドの例を示します。

```
db2 "SELECT * FROM sample WHERE CONTAINS (DESCRIPTION, '%enable%') = 1"
```

第 2 部 インストール

第 4 章 DB2 クライアント/サーバー環境での Net Search Extender のインストール

Net Search Extender 検索機能は SQL に統合されており、サーバーで実行されます。

重要: Net Search Extender は非推奨になりました。使用は推奨されません。また将来のリリースで除去される可能性があります。DB2 データベースに保管されているフルテキスト文書を SQL ステートメントおよび XQuery ステートメントを使用して検索する高速かつ用途の広い方法として、DB2 Text Search を使用してください。詳しくは、Net Search Extender から DB2 Text Search へのマイグレーションに関するトピックを参照してください。

Net Search Extender 検索機能は SQL に統合されており、サーバーで実行されます。したがって、テキスト検索照会を発行するために、クライアント上に Net Search Extender をインストールする必要はありません。

リモート・クライアントから Net Search Extender を管理する計画の場合、DB2 サーバー・エディションのいずれかと Net Search Extender 自体がクライアント・サイドにインストールされている必要があります。

第 5 章 Net Search Extender のインストールのシステム要件

DB2 バージョン 9.7 での、Net Search Extender がサポートするプラットフォームごとの最小のハードウェア要件およびソフトウェア要件。

Net Search Extender を実行するには、その前に DB2 バージョン 9.7 をインストールする必要があります。

Net Search Extender は、以下のプラットフォームでサポートされています。

- AIX® (64 ビット) プラットフォーム
- Linux x86 (32 ビット) プラットフォーム
- Linux x64 (64 ビット) プラットフォーム
- Linux on zSeries® (64 ビット) プラットフォーム
- HP-UX on Itanium-based HP Integrity Series (64 ビット) プラットフォーム
- Solaris UltraSPARC (64 ビット) プラットフォーム
- Windows on x86 (32 ビット) プラットフォーム
- Windows on x86 (64 ビット) プラットフォーム

注: Windows (32 ビット) プラットフォームの Net Search Extender では、パーティション・データベース環境はサポートされていません。

Net Search Extender の一般インストールの最小ディスク・スペースは 50 MB です。索引付けするデータの量によっては、追加のハードウェア要件が発生する可能性があります。

第 6 章 パーティション DB2 サーバーでの Net Search Extender のインストール

すべてのパーティションで、DB2 が正しくインストールおよび構成されていることを確認してください。

このタスクについて

重要: Net Search Extender は非推奨になりました。使用は推奨されません。また将来のリリースで除去される可能性があります。DB2 データベースに保管されているフルテキスト文書を SQL ステートメントおよび XQuery ステートメントを使用して検索する高速かつ用途の広い方法として、DB2 Text Search を使用してください。詳しくは、Net Search Extender から DB2 Text Search へのマイグレーションに関するトピックを参照してください。

DB2 をインストールした後、各パーティションに Net Search Extender をインストールする必要があります。

注: インスタンス所有者 ID と異なる fenced ユーザー ID では、パーティション・データベースを扱う作業ができません。

第 7 章 UNIX での Net Search Extender のインストール

UNIX 上に Net Search Extender をインストールするには、製品をインストールしてから DB2 インスタンスを更新します。

このタスクについて

重要: Net Search Extender は非推奨になりました。使用は推奨されません。また将来のリリースで除去される可能性があります。DB2 データベースに保管されているフルテキスト文書を SQL ステートメントおよび XQuery ステートメントを使用して検索する高速かつ用途の広い方法として、DB2 Text Search を使用してください。詳しくは、Net Search Extender から DB2 Text Search へのマイグレーションに関するトピックを参照してください。

手順

Net Search Extender を UNIX にインストールするには、次のステップに従ってください。

1. 製品コンポーネントをインストールします。
 - a. ターゲット・マシンに root ユーザーとしてログオンします。
 - b. ご使用のプラットフォームの適切なディレクトリーに移動します。
 - `cd /cdrom` ここで、*cdrom* は CD-ROM ドライブのパスです。
 - `cd platform`
 - c. 古いバージョンの NSE が存在する場合は、アンインストールしてください。
 - d. `./nsetup.sh` を呼び出して、画面に表示される指示に従います。

使用条件を受諾すると、選択可能なインストール・パスのリストが表示されます。インストールする Net Search Extender のバージョンおよびインストール済みの DB2 データベース製品のコピーのバージョンによって、適切なインストール・パスは異なります。インストール用のパスを選択すると、このパスに製品がインストールされます。/tmp ディレクトリーに、接頭部 `db2nsei` でインストール・ログ・ファイルが書き込まれます。

2. DB2 インスタンスを更新します。
 - a. root ユーザーとしてアクティブであることを確認します。
 - b. Net Search Extender をインストールした作業ディレクトリーのパスを変更する場合は次のコマンドを実行してください。

```
cd path/instance
```

path は、Net Search Extender がインストールされている DB2 コピーのパスです。
 - c. `db2iupdt` を実行します。 `./db2iupdt db2instance` を使用します。ここで、*db2instance* は、Net Search Extender で使用する既存の DB2 インスタンスのユーザー ID の名前です。

第 8 章 Windows でのインストール

この方式では、いくつかのコマンド行オプションを使用します。パラメーターを必要とするコマンド行オプションは、オプションとそのパラメーターの間にスペースを入れずに指定する必要があります。

このタスクについて

重要: Net Search Extender は非推奨になりました。使用は推奨されません。また将来のリリースで除去される可能性があります。DB2 データベースに保管されているフルテキスト文書を SQL ステートメントおよび XQuery ステートメントを使用して検索する高速かつ用途の広い方法として、DB2 Text Search を使用してください。詳しくは、Net Search Extender から DB2 Text Search へのマイグレーションに関するトピックを参照してください。

この方式では、いくつかのコマンド行オプションを使用します。パラメーターを必要とするコマンド行オプションは、オプションとそのパラメーターの間にスペースを入れずに指定する必要があります。

手順

Windows でインストールするには、管理権限を持つユーザー ID でログオンした後、次のステップに従う必要があります。

1. `cdrom:%windows%install%setup.exe` を使用して、パッケージからターゲット・マシンにファイルを転送します。

すべての DB2 インスタンスで、ユーザー ID およびパスワードを入力して適切な Net Search Extender サービスを作成する必要があります。

使用条件を受諾すると、選択可能な DB2 コピー名のリストが表示されます。インストールする Net Search Extender のバージョンに応じて、使用できる適切な DB2 コピー名が表示されます。DB2 コピー名を選択すると、この DB2 コピーが既にインストールされているパスに製品がインストールされます。

サイレント・インストールの場合は、`setup.exe` を 2 つの異なるモードで呼び出すことができます。

RECORDMODE

サイレント・インストール応答ファイルを作成します。

`setup.iss` という名前の定義済みの応答ファイルが、インストール・ソース・ディレクトリーにあります (これはサイレント・インストールの場合には使用されません)。新規応答ファイルを作成する場合は、`setup.exe -r` を実行してください。新規 `setup.iss` が Windows ディレクトリーに作成されます。この `setup.iss` をご使用のインストール・ソース・ディレクトリーにコピーしてください。古い応答ファイルを必ずバックアップしておいてください。

例: `setup.exe -r -f1"d:%some_directory%setup.iss"`

SILENTMODE

サイレント・インストールです。

ファイル `setup.iss` が、インストール・ソース・ディレクトリーに配置されていることを確認してください。`setup.exe -s` を実行してください。インストール・ソース・ディレクトリーにある `setup.log` ファイル内の `ResponseResult` に `0` が設定されている場合、インストールは正常に実行されています。

```
例: setup.exe -s -f1"d:¥some_directory¥setup.iss"  
-f2"d:¥another_directory¥mysetup.log"
```

インストール・パラメーターは次のとおりです。

- `/r:` 記録モード (サイレント・インストール用の応答ファイルを記録する)
 - `/s:` サイレント・インストール (サイレント・インストールを実行する)
 - `/f1:` 代替応答ファイル名を指定する (絶対パス)
 - `/f2:` 代替セットアップ・ログ・ファイル名を指定する (絶対パス)
2. インストール後にシステムをリブートします。
 3. `db2text start` を呼び出して、DB2 Net Search Extender のインスタンス・サービスを開始します。

タスクの結果

各 DB2 インスタンスが、それぞれ Windows サービスを作成します。DB2 インスタンス・サービスの実行は、システム・アカウントではなく、必ずユーザー・アカウントのもとで行ってください。

パーティション・インスタンスでは、パーティションごとに Windows サービスのセットが 1 つ作成されます。すべてのパーティションに対する DB2EXT インスタンス・サービスは、DB2 インスタンス・サービスと同じユーザー・アカウントで実行する必要があります。

Net Search Extender は Microsoft Cluster Server ではサポートされていません。

第 9 章 Net Search Extender のディレクトリー名とファイル名

Net Search Extender コマンドでは、名前を指定する必要があります。

すべての Net Search Extender コマンドで、ディレクトリー名およびファイル名を SBCS 文字で指定しなければなりません。パス名 (ファイル名を含む) の最大長は 256 バイトです。

第 10 章 Outside-In ライブラリーのインストール

Net Search Extender を Stellent の Outside-In ソフトウェアと併用する場合、プラットフォームごとにライブラリーをセットアップする必要があります。

このタスクについて

いくつかのプラットフォームでは、Stellent の Outside In ソフトウェアを使用できません。詳細は、<http://www.oracle.com>を参照してください。

手順

Net Search Extender を Stellent の Outside-In ソフトウェアと併用する場合、プラットフォームごとにライブラリーをセットアップする必要があります。

- Windows の場合、ライブラリーが置かれているディレクトリーを path 環境変数に追加してください。
- UNIX の場合は、Outside In ライブラリーを DB2 lib インストール・ディレクトリーに追加してください。

第 11 章 インストール検査

UNIX でのインストール検査

Net Search Extender が正しくインストールされているかを検査するには、次のステップを実行します。

手順

Net Search Extender が正しくインストールされているかを検査するには、次のステップを実行します。

1. 次のステップに従って管理スクリプト **nseample** を呼び出し、テキスト索引をセットアップします。
 - a. `instance_owner_home/sqllib/samples/extenders/db2ext` に変更します。
 - b. `./nseample yourdb` を呼び出します。 データベースがまだ存在していない場合、このコマンドを実行することで、そのデータベースが作成されることに注意してください。
 - c. ホスト・ディレクトリーに生成された出力ファイル `nseample.log` を確認します。
2. 同じ DB2 コマンド・ウィンドウからいくつかのサンプル照会を呼び出し、実行します。
 - a. `db2 connect to yourdb` を使用してデータベースと接続します。
 - b. `db2 -tvf search` を使用してサンプル照会を実行します。
 - c. スクリプト内に含まれる照会結果を確認します。 どの照会も 1 つ以上のヒットを戻します。

タスクの結果

`nseample_partitioned.log` ファイルにエラーがなく、すべての照会が機能していれば、Net Search Extender は正常にインストールされています。

パーティション・データベースでは、次の検査サンプルを使用してください。

```
nseample_partitioned database_name [node_number][table_space_filename]
```

Windows でのインストール検査

Net Search Extender が正しくインストールされているかを検査するには、次のステップを実行します。

手順

Net Search Extender が正しくインストールされているかを検査するには、次のステップを実行します。

- 非パーティション環境:
 1. 次のようにサンプル・テキスト索引をセットアップしてください。

- a. `db2cmd` を呼び出して、DB2 コマンド・ウィンドウを開きます。
 - b. `sqllib¥samples¥extenders¥db2ext` に移動します。
 - c. DB2 コマンド・ウィンドウから `nse-sample.bat yourdb` を呼び出します。データベースがまだ存在していない場合、このコマンドを実行することで、そのデータベースが作成されることに注意してください。
 - d. 現行ディレクトリーに生成された出力ファイル `nse-sample.log` を確認します。
2. DB2 コマンド・ウィンドウから次のサンプル照会を呼び出し、実行します。
 - a. `db2 connect to yourdb` を使用してデータベースと接続します。
 - b. `db2 -tvf search` を使用してサンプル照会を実行します。
 - c. スクリプト内に含まれる照会結果を確認します。どの照会も 1 つ以上のヒットを戻します。

`nse-sample.log` ファイルにエラーがなく、すべての照会が機能していれば、Net Search Extender は正常にインストールされています。

- パーティション・データベース環境: 次のようにサンプル・テキスト索引をセットアップしてください。
 1. `db2cmd` を呼び出して、DB2 コマンド・ウィンドウを開きます。
 2. `<sqllib>¥samples¥extenders¥db2ext` に移動します。
 3. DB2 コマンド・ウィンドウから `nse-sample_partitioned.bat <yourdb>` を呼び出します。
 4. 現行ディレクトリーに生成された出力ファイル `nse-sample_partitioned.log` を確認します。

`nse-sample_partitioned.log` ファイルにエラーがなく、すべての照会が機能していれば、Net Search Extender は正常にインストールされています。

第 12 章 Net Search Extender のアンインストール

Net Search Extender をシステムから永続的に削除して、Net Search Extender の索引をすべて削除するためには、まず Net Search Extender の索引を含んでいる各データベースを使用不可にしてから、Net Search Extender のみを削除する必要があります。

UNIX での Net Search Extender のアンインストール

UNIX オペレーティング・システムから Net Search Extender をアンインストールするには、データベースを使用不可に設定し、インスタンスを停止してから `db2nse_deinstall` コマンドを発行します。

手順

UNIX オペレーティング・システムで Net Search Extender を正常にアンインストールするには、以下のステップを実行します。

1. Net Search Extender をアンインストールする DB2 インスタンスごとに、以下を実行します。
 - a. DB2 インスタンスのユーザー ID に切り替えます。
 - b. このインスタンスで今後 Net Search Extender を使用しない予定の場合、Net Search Extender をアンインストールする前に、索引をドロップし、次のようにしてデータベースを使用不可にしてください。

```
db2text disable database for text connect to databasename
```
 - c. DB2 Net Search Extender インスタンスを停止します。
 - d. DB2 インスタンスを停止します。
2. root ユーザーとしてアクティブであることを確認します。
3. Net Search Extender を削除する DB2 のパスに作業ディレクトリーを変更します。例: `cd /opt/IBM/db2/V10.1/install`
4. `./db2nse_deinstall` コマンドを発行します。コマンド構文の詳細については、254 ページの『`db2nse_deinstall` コマンド』を参照してください。

Windows での Net Search Extender のアンインストール

Windows オペレーティング・システムで Net Search Extender をアンインストールするには、データベースを使用不可にし、インスタンスを停止し、プログラムを削除します。

手順

Windows で Net Search Extender を正しくアンインストールするには、以下のステップを実行します。

1. データベースごとに、`db2text disable database for text connect to databasename` を実行します。

2. DB2 インスタンスを停止します。
3. 「設定」 > 「コントロール パネル」 > 「プログラムの追加と削除」を選択します。インストール時に Net Search Extender に割り当てられた、DB2 *COPYNAME* に対応する Net Search Extender *COPYNAME* エントリーをリストから選択します。
4. 「削除」をクリックします。

第 3 部 高可用性 (HA) のための Net Search Extender (NSE) の構成

DB2 Net Search Extender を構成して、高可用性ノード間での索引の共有、Net Search Extender 索引のバックアップおよびリストア操作を利用するようにすることにより、高可用性をサポートすることができます。Net Search Extender のフルテキスト索引は、DB2 データベースに保管されたデータと、ファイル・システム上のいくつかの外部ファイルに保管されたデータで構成されます。高可用性のために DB2 を構成することによって、フェイルオーバー時にリカバリーされるのは、データベース内にある Net Search Extender データのみです。NSE 固有の外部ファイルは、ユーザーのシナリオおよび使用中のプラットフォームに適用可能なファイル共有技術を使用して、フェイルオーバー・ノードと共有される必要があります。外部ファイルは、何らかの索引の更新操作が中断されてもリストアされないため、索引ファイルが破損する原因になります。ファイルを手動でリストアできるように、バックアップしておく必要があります。

索引更新を中断すると、修復と予測ができない索引の破損が発生するおそれがあります。破損の重大度は、影響を受ける索引ファイルと、中断の時点での索引操作フェーズによって異なります。いくつかの索引ファイルはコピーが更新されるのではなく直接更新されるので、このことも、ロールバック・リカバリーを難しくする要因となります。そのため、索引更新中にフェイルオーバーが発生した場合は、最後に正常に実行された索引更新操作 (索引ディレクトリー・スナップショットとして保存されている) から、破損した索引ファイルをリストアする必要があります。

高可用性の構成により、フェイルオーバー時に索引更新が中断されても、共有ストレージ上にある索引ファイルが不整合の状態になるのを防ぐことができます。フェイルオーバー・システム上にあるデータベース・オブジェクトを使用して、索引ファイルを元の整合性のとれた状態に戻すことができます。

ファイル・スナップショットがどのプラットフォームによってもサポートされない場合、NSE 共有索引フォルダードライブのために、そのプラットフォームに適用できないいずれかの対応するファイル・システム共有技術を考慮する必要があります。

索引ディレクトリー・スナップショット

1. すべての Net Search Extender 索引ファイルは、最新の索引ファイルをバックアップおよびリストアするために、専用のファイル・システムに保管する必要があります。そのファイル・システムに他のデータを保管すべきではありません。
2. それぞれの索引は、独自のファイル・システムに存在する必要があります。あるいは、索引間でファイル・システムを共有することもできますが、共通のファイル・システムを共有する索引の更新スケジュールは、2 つの更新が同時に行われないようにシリアライズされます。その場合、Net Search Extender 索引のための個別のファイル・システムの数、システムが処理できる並行更新処理の数に合わせて調整されます。
3. スナップショットに使用されるスペースは、初めはとても小さいですが、ファイル・システムのコンテンツが変更されるにつれて増加する傾向にあります。索引ファイル・システムのスナップショット用に、十分なファイル・スペースが存在

することを確認してください。ファイル・スペースの使用量をモニターし、スナップショット用に十分なスペースが存在するようにします。

フェイルオーバーの準備

索引は、高可用性ノード間での共有ストレージ上にあります。それぞれの索引更新およびスケジュールされた更新が行われた直後に、その索引ディレクトリーのスナップショットを撮る必要があります。これらの指示は、次のステップに示すように、スクリプトにカプセル化して外部のスケジューラーにより実行できます。

1. 索引ファイルが高可用性ノード間の共有場所にあるかどうかを確認します。
2. `db2ext.tcommandlock` 表、および `work` ディレクトリーで `DB2 Net Search Extender` の状態をチェックしてください。
3. スナップショット・プロシージャーを実行して `Net Search Extender` 索引ファイル・システムのスナップショットを撮り、共有ストレージへ入れます。
4. `Net Search Extender` の **UPDATE INDEX** コマンドを呼び出します。
5. 索引更新の完了後に自己定義マークを削除します。

注: `Net Search Extender` ネイティブのスケジュール済み索引更新が呼び出せるのは **DB2TEXT UPDATE INDEX** コマンドのみなので、**UPDATE FREQUENCY** を `NONE` に設定してそれを無効にします。代わりに、`UNIX`、`Linux` の **CRON** コマンドや、`Windows` オペレーティング・システムの **AT** コマンドなど、オペレーティング・システム固有の索引更新スケジューリングを使用してください。それらのコマンドは、自動化された更新スケジュールを持つ索引ごとに 1 つの **crontab** エントリーを指定することで、指定された間隔でラッパー・スクリプトを呼び出します。これにより、共有ストレージ上の成功した最新の更新から、ファイル・システムにあるすべての索引の現行のスナップショットを確実に入手できます。

フェイルオーバーの間の索引特性

索引リカバリーにおいて重要なのは、フェイルオーバーによって索引が破損したかどうかを見極めることです。これには、分かっている最新の良好な状態の索引にフォールバックすることが必要であり、以下の `Net Search Extender` 索引更新処理によって判別できます。

- 各索引更新は、`db2ext.tcommandlocks` 表で挿入操作と削除操作のペアで内部的にカプセル化されます。
- その索引に対して管理コマンドが並行して行われないように、索引更新ではまず、その表に行が作成され、その行に索引に関する情報 (名前付き索引、タイム・スタンプ、操作のタイプ) が含まれます。更新が終了する前に、再度行が表から削除され、その索引に対して新しい管理コマンドを実行できるようになります。
- フェイルオーバーの間に索引更新が発生しない場合、`db2ext.tcommandlocks` 表に行は含まれず、これ以上の処置は必要ありません。高可用性サポートを介して、ログ表に格納されているすべてのデータが即時にフェイルオーバー・システムで使用可能になり、次の通常の索引更新の準備が整います。
- 索引更新中にフェイルオーバーが発生した場合、フェイルオーバー・ノード上の `db2ext.tcommandlocks` 表には、障害が発生した時に更新に関係していた索引ごとに 1 つの行が表示されます。影響を受けた索引が複数存在する場合があります、それぞれが `db2ext.tcommandlocks` の 1 つの行に対応しているため、各行ごとに操作

を繰り返す必要があります。その場合は、スナップショットをリストアするために、手動でリカバリーを開始する必要があります。影響を受けた索引はすべてその表にロック項目があるため、以降の (スケジュールもしくは手動による) 更新に対して保護されています。

- ログ表に項目がまだ保持されているかどうか調べます。索引のログ表にある最も古い項目のタイム・スタンプと、索引のイベント表にある最新の CTE0003 項目を比較します。

ログ表の最も古い項目が、最新の CTE0003 よりも新しい場合、フェイルオーバーの前にすでにログ表のクリーンアップが行われていますが、db2ext.tcommandlocks 項目はまだ削除されていません。この場合、索引は破損していないのでスナップショットをリストアしないでください。手動で db2ext.tcommandlocks 項目を削除した後、通常どおり処理を続行してください。

ログ表の最も古い項目が、最新の CTE0003 よりも古い場合、索引をスナップショットからリストアする必要があります。

スナップショットからの索引のリストア

1. 影響を受けた索引の索引ディレクトリーから、すべての索引ファイルを削除します。処理中にその索引を検索すると失敗するため、Net Search Extender を停止してください。

```
rm -rf /myWORK/NODE0000/TMP_IX300608/*
```

2. 空のディレクトリーを、スナップショットの内容に置き換えます。これはファイルを物理的にコピーする必要があるため、時間がかかります。

```
rm -rf /myINDEX
mount -o snapshot /dev/fs1v06 /mnt/
cp -pR /mnt/* /myINDEX
```

3. 索引ディレクトリーの内容をリストアした後、db2ext.tcommandlocks 表から索引に対応する行を手動で削除します。

```
db2 "delete from db2ext.tcommandlocks"
```

4. 前述の手順を、影響を受けたすべての索引に対して繰り返します。
5. 終了したら、Net Search Extender を再始動します。これで、フェイルオーバー・ノード上で通常の操作を続行できます。

ログ表にある Net Search Extender の内容はそのまま残り、**DB2TEXT INDEX UPDATE** の新規呼び出しが従来どおり処理を行います。イベント表に元の索引更新操作中に作成された項目が含まれている可能性があるため、その表で手動クリーンアップが多少必要になる場合があります。

第 4 部 DB2 Net Search Extender へのアップグレード

Net Search Extender を DB2 Net Search Extender にアップグレードする前に、DB2 サーバー、およびすべてのデータベースを含むインスタンス (Linux および UNIX の場合) が、DB2 バージョン 10.1 に正常にアップグレードされている必要があります。

始める前に

重要: Net Search Extender は非推奨になりました。使用は推奨されません。また将来のリリースで除去される可能性があります。DB2 データベースに保管されているフルテキスト文書を SQL ステートメントおよび XQuery ステートメントを使用して検索する高速かつ用途の広い方法として、DB2 Text Search を使用してください。詳しくは、Net Search Extender から DB2 Text Search へのマイグレーションに関するトピックを参照してください。

- アップグレードの前にすべてのテキスト索引ディレクトリーと索引サブディレクトリーをバックアップしてください。129 ページの『第 39 章 テキスト索引のバックアップとリストア』を参照してください。

このタスクについて

Net Search Extender のアップグレードは Net Search Extender バージョン 9.5 またはバージョン 9.7 から上位バージョンへのアップグレードのみがサポートされています。

手順

DB2 Net Search Extender にアップグレードするには、次のようにします。

1. 以下のいずれかの作業を実行して、Net Search Extender がインストールされている DB2 サーバーをアップグレードします。
 - 「DB2 バージョン 10.1 へのアップグレード」の『DB2 サーバーのアップグレード (Windows) 』
 - 「DB2 バージョン 10.1 へのアップグレード」の『DB2 32 ビット・サーバーから 64 ビット・システムへのアップグレード (Windows)』
 - 「DB2 バージョン 10.1 へのアップグレード」の『DB2 サーバーのアップグレード (Linux および UNIX)』

ご使用のデータベースはこれらの作業の一環としてアップグレードされます。Linux または UNIX 上に DB2 エンジン・ライブラリーに依存しない外部 unfenced ルーチンがある場合、**UPGRADE DATABASE** コマンドは、その外部ルーチンを **FENCED** および **NOT THREADSAFE** として再定義します。新しいマルチスレッドのデータベース・マネージャーでルーチンを安全に実行する方法について詳しくは、「DB2 バージョン 10.1 へのアップグレード」の『C、C++、および COBOL ルーチンのアップグレード』を参照してください。データベースのアップグレード中に変更された、DB2EXT というスキーマ名がついた Net

Search Extender 関数は、ステップ 6 で db2extmdb スクリプトによって NOT FENCED および THREADSAFE として再定義されます。

2. DB2 Net Search Extender をインストールします。

DB2 データベースとは異なり、DB2 Net Search Extender では「アップグレード・インストール」はサポートされていません。

インストール済みの DB2 コピーが、DB2 の「アップグレード・インストール」オプションを使用して移されている場合、インストール済みの DB2 Net Search Extender のコピーは古いバージョン・レベルのままになります。

DB2 Net Search Extender を古いバージョンの DB2 Net Search Extender の上にインストールしようとする、既存の DB2 Net Search Extender インストールをまず除去しなければならないというエラー・メッセージを受け取ります。この場合、以前のバージョンの DB2 Net Search Extender をアンインストールした後、DB2 Net Search Extender をインストールしてください。Windows オペレーティング・システムでは、DB2 Net Search Extender をアンインストールした後にマシンをリブートしてください。

3. DB2 Net Search Extender インスタンスのアップグレードは Linux および UNIX の場合にのみ該当します。Windows オペレーティング・システムの場合は、このステップは無視してください。インスタンスをアップグレードするには、root としてログオンし、次の構文を使用して **db2extimigr** スクリプトを実行します。

```
DB2DIR/instance/db2extimigr [-h|-?] InstanceName
```

ここで、*DB2DIR* は DB2 バージョン 10.1 のコピーをインストールしたディレクトリーです。

4. Linux および UNIX では、正常に DB2 Net Search Extender インスタンスがアップグレードされた後、インストールを確認してから、データベースのアップグレードを続行してください。Windows オペレーティング・システムでは、インストールをすぐに確認できます。

注: DB2 Net Search Extender サンプル・スクリプトを、DB2 Net Search Extender にアップグレードされていないデータベースには適用しないでください。安全な方法は、データベースを作成して検査を実行することです。詳細については、37 ページの『UNIX でのインストール検査』と 37 ページの『Windows でのインストール検査』を参照してください。

5. 以前のリリースの Net Search Extender で使用可能になっている、すべてのデータベースをアップグレードしてください。現在、**db2extmdb** の既知の問題 (データベースのアップグレード後、ビュー DB2EXT.DBDEFAULTS の項目が重複してしまう) を解決するには、ステップ a) および c) が必要です。データベースのアップグレード・ステップを実行するには、以下のようにします。

- a. DB2 サーバーにインスタンス所有者としてログオンします。

Net Search Extender インスタンス・サービスを正常に停止または開始できる必要があります。そのため、アップグレードするデータベースに対して DATAACCESS 権限のある DBADM が必要です。Windows オペレーティング・システムでは、インスタンス・ユーザーは Local Administrator グループのメンバーでなければなりません。

Windows オペレーティング・システムでは、完全な管理者特権で実行している DB2 コマンド・ウィンドウから行う必要があります。詳細については、ユーザー・アクセス制御フィーチャーを参照してください。

Windows 2008 および Windows Vista オペレーティング・システム以降では、標準のユーザー権限で実行しているコマンド・ウィンドウから **db2extmdb** コマンドを開始すると、動作しません。(「CTE0228 ユーザーにオペレーティング・システム・レベルでの十分なアクセス権がありません」)。

- b. 次の構文を使用して **db2extmdb** スクリプトを実行し、Net Search Extender 用に使用可能にしたデータベースをアップグレードします。

`db2extmdb database-name`

このスクリプトの実行中は、テキスト索引を持つユーザー表に何の変更も加えないようにしてください。Net Search Extender 索引を持つすべてのデータベースに対して、このコマンドを繰り返すことができます。

アップグレードの各ステップはすべて以下のいずれかのディレクトリー内にある `db2extmdatabase-name.log` という名称のファイルにログとして記録されます。

- `INSTHOME/sql1lib/db2ext/` (Linux および UNIX オペレーティング・システムの場合)
- `DB2PATH¥db2ext¥` (Windows オペレーティング・システムの場合)

ここで、`INSTHOME` はインスタンス・ホーム・ディレクトリーであり、`DB2PATH` は DB2 バージョン 10.1 のコピーをインストールした場所です。

6. バージョン 10.17 より前の DB2 32 ビット・サーバーから バージョン 10.1 の 64 ビット・サーバーにアップグレードした場合、テキスト索引をドロップしてから、テキスト索引を再作成する必要があります。125 ページの『第 37 章 テキスト索引のドロップ』および 97 ページの『第 28 章 テキスト索引の作成』を参照してください。Net Search Extender では、32 ビット・インスタンスで作成したテキスト索引を 64 ビット・インスタンスで使用することはできません。サーチ・エンジンがエラー CTE0101 (理由コード: 17) を戻します。
7. Windows オペレーティング・システム上で、DB2 サーバーのアップグレードの前に、バージョン 10.1 より前の DB2 コピーのインストール・ディレクトリーの下で作成したテキスト索引を使用したい場合は、バックアップしたテキスト索引ディレクトリーをリストアします。

129 ページの『第 39 章 テキスト索引のバックアップとリストア』を参照してください。DB2 バージョン 10.1 のインストール時に「既存の処理」ウィンドウのアップグレード操作で DB2 コピーを選択した場合、またはアップグレード後にバージョン 10.1 より前の DB2 コピーをアンインストールした場合に、テキスト索引ディレクトリーのリストアが必要です。

テキスト索引の構成には、マイグレーション前のテキスト索引ディレクトリーの場所が含まれています。テキスト索引ディレクトリーをリストアしなければ、テキスト索引を使用する照会や索引管理操作は失敗します。

第 5 部 32 ビットから 64 ビットへの DB2 Net Search Extender 索引マイグレーション・ツール

32 ビットの DB2 インスタンスで作成された Net Search Extender 索引は、64 ビットのインスタンスと互換性がありません。

32 ビット・インスタンスで作成された索引を 64 ビットの DB2 インスタンスで検索または更新しようとする、必ずエラー・メッセージが表示され、操作が完了しません。これは、索引の一部であるいくつかの小さいファイルに、ワード・サイズ固有のフォーマットがあるためです。

現在、32 ビットから 64 ビットのインスタンスにアップグレードする方法として推奨される唯一の方法は、32 ビット・インスタンスで作成した NSE 索引をドロップして再作成する方法です。索引のサイズが小さいか中程度であり、表のテキスト・コンテンツが直接アクセス可能である場合は、この推奨方法を技術的に比較的容易に行うことができます。適切にチューニングされたシステムでは、DB2 データベース・システムは毎時 12 GB まで処理できますが、それでも、索引が大きくなればなるほど、NSE 索引の再ビルドにかなりの時間がかかるようになります。

索引の完全な再作成を行う必要が生じないようにするため、ツールを使用して索引をオフラインでマイグレーションできます。このツールの現行バージョンは AIX オペレーティング・システムおよび Solaris オペレーティング・システムのみをサポートしています。

Content Manager システムの場合、Net Search Extender によって索引付けされるコンテンツは、属性以外の索引について索引が定義されている表に対してローカルではありません。属性はライブラリー・サーバーに保管されたメタデータであり、そのコンテンツはローカルです。CM システムの場合、テキスト文書はフルテキスト検索が可能な Net Search Extender データベース表に対してローカルではありません。代わりに、文書は別個の文書サーバーに保管され、索引付けのために Net Search Extender に送信されます。コンテンツはリモート・ロケーション (CM 用語ではリソース・マネージャー) から検索およびフィルタリングされるため、かなりの待ち時間がかかります。これによって、索引付け自体の速度も低下するため、NSE 索引の再ビルドのコストが極めて高くなります。

現在 32 ビット・インスタンスの DB2 を使用しており、実行している DB2 のリリースがまだバージョン 8 である場合、AIX と Solaris プラットフォームでは、バージョン 8 リリースからバージョン 9.1、バージョン 9.5 またはバージョン 9.7 にマイグレーションする前に、DB2 インスタンスを 32 ビットから 64 ビットへ更新する必要があります。これらのすべてのリリースでは、この 2 つのプラットフォーム用の 32 ビット・インスタンスはなくなりました。同様のことが Linux と Windows の場合にも当てはまります。64 ビットのシステムではメモリー使用量の制限がないため、大規模システムでは 64 ビット・インスタンスが推奨されます。

次のセクションで説明するように、索引のマイグレーションは、DB2 インスタンスおよびデータベースを 32 ビットから 64 ビットへマイグレーションする前に行う必要があります。

索引マイグレーションの手順

1. FTP サイトから索引のマイグレーション・ツールをダウンロードします。
2. 圧縮ファイルをシステム上の任意のディレクトリーで解凍します。
3. シェル・スクリプト **ctemigridx.sh** を実行することで索引のマイグレーション・ツールを実行できます。次のセクションで説明されているように、シェル・スクリプト (`./ctemigridx.sh`) を実行します。

DB2 バージョン 8 の 32 ビット AIX 環境では以下のようにします。

1. インスタンス所有者としてログインします。
2. アーカイブ `NSE_32_64_Idx_Migr_Tool_AIX_SOL.tar.gz` を解凍します。
3. `NSE_32_64_Idx_Migr_Tool_AIX_SOL` ディレクトリーまでナビゲートします。
4. DB2 データベース・マネージャーが稼働中であり、NSE が停止されていることを確認します。
5. `/usr/sbin/slibclean` を実行して、AIX ライブラリー・キャッシュをクリアします。この手順は、インストールされた NSE ライブラリーと `NSE_32_64_Idx_Migr_Tool_AIX_SOL` ディレクトリーにある同名のライブラリー間の競合を防ぐために必要です。システム構成によっては、**slibclean** を実行するのに root ユーザー権限が必要である場合があります。
6. シェル・スクリプト **ctemigridx.sh** を実行します。スクリプトは、選択可能な 2 つのモードのいずれかで実行できます。

最初のモードでは、データベース名を引数として取り、そのデータベース内にあるすべての既存の索引を自動的に特定します。それらの索引すべてを一度にマイグレーションすることも、選択した索引をマイグレーションすることもできます。

モード 1: このモードではデータベース接続を確立し、NSE データベース表から必要なデータをすべて照会します。その後、マイグレーション可能な索引のリストが表示されます。リストから 1 つまたはすべての索引を選択できます。コマンドに **-check** パラメーターを追加すると、マイグレーションを伴わずに必要なすべてのステップが実行されます。

例: `./ctemigridx.sh -dbname sample`

モード 2 は、マイグレーションする特定の索引の索引名が既に分かっており、その索引にマイグレーションのターゲットを絞る場合に使用できます。索引の数が多く、多数の索引の一覧から選択する手間を省きたい場合に役立ちます。特定の索引について繰り返しテストを実行している場合も便利です。このモードでは、ユーザーとの対話を行わずに索引情報のマイグレーションを行います。

モード 2: `./ctemigridx.sh -i index-name -p index-directory [-showmap]`

例: `./ctemigridx.sh -i IX123456 -p /home/user/sql1lib/db2ext/indexes`

索引ディレクトリーは、索引の作成時に使用するのと同じ方法で指定する必要があります。索引ディレクトリーには必ず `NODE0000` サブディレクトリーがあり、

その中に索引が入っています。 **-showmap** フラグを追加すると、さらに検査できるよう、マイグレーションされた索引の属性 (セクションおよび属性) のダンプがログ・ファイルに表示されます。

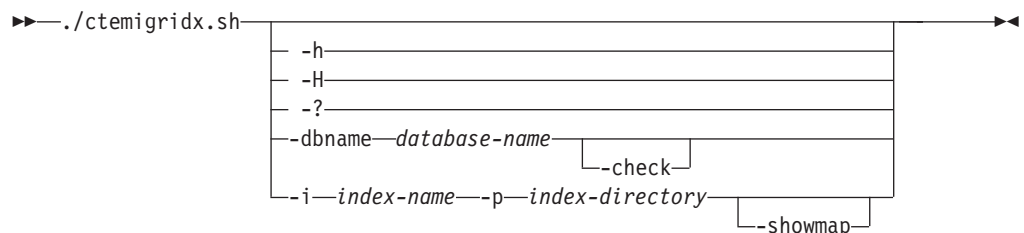
```
./ctemigridx.sh -i IX123456 -p /home/user/sql1lib/db2ext/indexes  
-showmap
```

7. この後、索引のマイグレーションが完了します。ユーザーは DB2 文書に説明されている方法を使用して、64 ビットへのインスタンス・マイグレーションに進むことができます。

DB2 バージョン 8 の 32 ビット Solaris 環境では以下のようになります。

すべてのステップは、**slibclean** のステップが不要であること以外は、AIX の場合に必要なステップと同じです。Solaris では **slibclean** がありません。

ctemigridx.sh 構文



-h, -H, -?

コマンド・ヘルプを表示して終了します。

-dbname database-name

データベースの名前。

-check

マイグレーションせずに必要なステップをすべて実行します (シミュレーション)。

-i index-name

マイグレーションする索引の名前。形式は常に IXnnnnnn となります。

-p index-directory

索引の配置先ディレクトリー。索引の作成時に使用するのと同じ方法で指定する必要があります。多くの場合、~/sql1lib/db2ext/indexes です。

-showmap

さらに検査できるように、マイグレーションされた索引の属性 (セクションおよび属性) のダンプをログ・ファイルに出力します。

注:

1. 現行のマイグレーション・ツールは、すでにマイグレーションされたバージョン 9 インスタンスに復元された索引のマイグレーションをサポートしていません。
2. 索引のマイグレーション・ツールを実行する前に DB2 インスタンスを開始し NSE を停止する必要があります。

3. ツールを実行する前に、現行ディレクトリーへの書き込み権限を持っている必要があります。
4. ツールは変更される索引ファイルのみをバックアップし、索引全体のバックアップは行いません。ツールを実行する前に索引ディレクトリー全体のバックアップを行うことをお勧めします。
5. ツールを、root ではなく DB2 のインスタンス所有者として実行する必要があります。
6. **db2_local_ps** コマンドで結果が出ない場合、DB2 データベース・マネージャーが開始されていないので、ツールは使用できません。
7. 現在、マイグレーション・ツールは単一ノード・システムのみをサポートしています。

第 6 部 32 ビットから 64 ビットの DB2 Net Search Extender 索引マイグレーション・ツール (Windows)

32 ビットの DB2 インスタンスで作成された DB2 Net Search Extender 索引は、64 ビットのインスタンスと互換性がありません。32 ビットの DB2 インスタンスで作成された索引を 64 ビットのインスタンスで検索または更新しようとする、エラー・メッセージが戻されます。

索引マイグレーション・ツールは、オフラインで索引をマイグレーションし、索引の再作成を回避します。マイグレーションは、少数のメタデータ・ファイルのみに影響を与え、大容量のデータ移動をすることなく素早く実行されます。このマイグレーションを実行するには、このパッケージで配布されるバッチ・スクリプトを実行します。

V8 32 ビットから V9 64 ビットへの索引のマイグレーション

このマイグレーション・ツールは、64 ビットにマイグレーションする 32 ビットのソース Net Search Extender インスタンスでのみ実行します。64 ビットのインスタンスで実行してはなりません。V8 システムからデータベースおよび索引ディレクトリーのバックアップを取ってください。バックアップの実行中に、拡張子 .an、.as、および .tf を持つメタファイルが適切にバックアップされているか確認してください。

マイグレーションのステップ

1. DB2 バージョン 8 データベースのユーザー定義関数 DATALINKS をドロップすることにより、データベースを準備してください。

```
db2 Drop Specific Function DB2Ext.DataLinkContent1
db2 Drop Specific Function DB2Ext.DataLinkContent1
db2 Drop Specific Function DB2Ext.DataLinkContent2
db2 Drop Specific Function DB2Ext.DataLinkContent4
db2 Drop Specific Function DB2Ext.DataLinkContent3
db2 Disconnect all
```
2. ctemigridx マイグレーション・ツールを使用して 32 ビット索引ファイルを 64 ビットにマイグレーションします。詳細は、『V9 32 ビット・マシンでツールを実行するためのステップ』を参照してください。
3. 32 ビット・インスタンスからデータベースおよび索引ファイルのバックアップを取ります。
4. バックアップしたファイルとデータベースを 64 ビットの DB2 バージョン 9.1 またはバージョン 9.5 がインストールされているターゲット・マシンに移動します。
5. Net Search Extender 索引をターゲット・マシンにマイグレーションする前に、必要な Net Search Extender データベースをターゲット・マシンにマイグレーションします。
6. Net Search Extender 索引および新しく生成された 64 ビットのメタファイルを以下の手順でターゲット・マシンにリストアします。
 - a. 次のコマンドを発行し、Net Search Extender を停止します。

- ```
db2text stop
```
- b. 索引ディレクトリーのバックアップ・コピーを、以前と同じパスにリストアします。
  - c. 次のコマンドを発行し、Net Search Extender を再始動させます。
 

```
db2text start
```
7. Net Search Extender データベースを現行リリースにマイグレーションします。
 

```
db2extmdb <database-name>
```
  8. 新しい索引と作業ディレクトリーを指定し、索引の変更を実行します。インスタンス所有者は、索引ファイルにアクセスして変更を加えるために、適切な権限を必要とします。
 

```
db2text alter index <index-name> for text index
directory <new_indexdir> work directory <new_workdir>
```
  9. これまでと同様に検索を実行し、新規にマイグレーションされた索引が検索可能かどうか検証します。

## V9 32 ビットから V9 64 ビットへの索引のマイグレーション

このマイグレーション・ツールは、64 ビットにマイグレーションする 32 ビットのソース Net Search Extender インスタンスでのみ実行します。64 ビットのインスタンスで実行してはなりません。V9 32 ビット・システムからデータベースおよび索引ディレクトリーのバックアップを取ってください。バックアップの実行中に、拡張子 .an、.as、および .tf を持つメタファイルが適切にバックアップされているか確認してください。ソースとターゲットのマシンが同じかどうかに応じて、2 つの異なる方法で DB2 バージョン 9 の 32 ビットからバージョン 9 の 64 ビットへのマイグレーションを実行できます。

- ソースとターゲットのマシンが異なる場合
  1. ctemigrdx マイグレーション・ツールを使用して V9 32 ビットのデータベースおよび Net Search Extender 索引を 64 ビットにマイグレーションします。詳細は、『V9 32 ビット・マシンでツールを実行するためのステップ』を参照してください。
  2. 32 ビット・インスタンスからデータベースおよび索引ファイルのバックアップを取ります。
  3. バックアップしたファイルとデータベースを V9 がインストールされているターゲット・マシンに移動します。
  4. Net Search Extender 索引をバージョン 9 の 64 ビットへマイグレーションする前に、必要な Net Search Extender データベースを DB2 バージョン 9 の 64 ビットへマイグレーションします。
  5. Net Search Extender 索引および新しく生成された 64 ビットのメタファイルを以下の手順で V9 64 ビットのセットアップにリストアします。
    - a. 次のコマンドを発行し、Net Search Extender を停止します。
 

```
db2text stop
```
    - b. 索引ディレクトリーのバックアップ・コピーを、以前と同じパスにリストアします。
    - c. 次のコマンドを発行し、Net Search Extender を再始動させます。
 

```
db2text start
```



6. Net Search Extender データベースを現行リリースにマイグレーションします。

```
db2extmdb database-name
```

7. 新しい索引と作業ディレクトリーを指定し、索引の変更を実行します。インスタンス所有者は、索引ファイルにアクセスして変更を加えるために、適切な権限を必要とします。

```
db2text alter index index-name for text index
directory new_indexdir work directory new_workdir
```

8. これまでと同様に検索を実行し、新規にマイグレーションされた索引が検索可能かどうか検証します。

- ソースとターゲットのマシンが同じ場合

次の手順は、使用しているバージョン 9 の 32 ビット・データベースおよび Net Search Extender 索引を 64 ビット Windows マシンへマイグレーションする方法を説明しています。

1. ctemigridx マイグレーション・ツールを使用して 32 ビット索引ファイルを 64 ビットにマイグレーションします。詳細は、『V9 32 ビット・マシンでツールを実行するためのステップ』を参照してください。
2. バージョン 9 の 32 ビットからバージョン 9 の 64 ビットへ DB2 インスタンスおよびデータベースをマイグレーションします。
3. db2extmdb コマンドを使用して、Net Search Extender データベースを現行リリースにマイグレーションします。

## V9 32 ビット・マシンでツールを実行するためのステップ

1. DB2 インスタンス所有者としてログオンします。
2. NSE\_32\_64\_Idx\_Migr\_Tool\_WINDOWS.zip アーカイブ CD を NSE\_32\_64\_Idx\_Migr\_Tool\_WINDOWS ディレクトリーへ解凍します。DB2 インスタンスが稼働中で、Net Search Extender が停止していることを確認します。索引ディレクトリーにある索引ファイルの排他ロックを試みる他のプロセスが存在しないことを確かめます。
3. バッチ・ファイル ctemigridx.bat ファイルを実行します。スクリプトは 2 つの異なるモードで実行できます。

モード 1 ではデータベース接続が確立され、Net Search Extender データベース表から必要なデータをすべて照会します。このモードでは、データベース名を引数として取り、そのデータベース内にあるすべての既存の索引を自動的に特定します。それらの索引すべてを一度にマイグレーションすることも、選択した索引をマイグレーションすることもできます。その後、マイグレーション可能な索引のリストが表示されます。リストから 1 つまたはすべての索引を選択できます。コマンドに `-check` パラメーターを追加すると、マイグレーションを伴わずに必要なステップすべてが実行されます。

```
ctemigridx -dbname sample
```

モード 2 はマイグレーションする特定の索引の索引名を既知っており、その索引にマイグレーションのターゲットを絞りたい場合に使用できます。索引の数が多く、多数の索引の一覧から選択する手間を省きたい場合に役立ちます。特定の索引について繰り返しテストを実行している場合も便利です。

```
ctemigridx -i index_identifier -p index_directory -[showmap]
```

索引ディレクトリーは、索引の作成時に使用するのと同じ方法で指定する必要があります。索引ディレクトリーには必ず NODE0000 サブディレクトリーがあり、その中に索引が入っています。現在、スクリプトは単一ノード・システムのみをサポートしています。-showmap フラグを追加すると、さらに検査できるよう、マイグレーションされた索引の属性のダンプがログ・ファイルに表示されます。

```
ctemigridx -i IX123456 -p D:%sqllib%db2ext%indexes -showmap
```

4. Net Search Extender 索引のマイグレーションが完了すると、索引ディレクトリーにいくつかの \*.32 ファイルが見つかります。マイグレーションされた索引に従って、以下の新規ファイルが見つかります。

- IX123456.as.32 (古い 32 ビット IX123456.as ファイルのバックアップ)
- IX123456.an.32 (古い 32 ビット IX123456.an ファイルのバックアップ)
- IX123456.tf.32 (古い 32 ビット IX123456.tf ファイルのバックアップ)

マイグレーションにより、一時ディレクトリーが 1 つ作成されます。これはマイグレーション中に作成されます。これが存在しない場合、TMP\_IX123456 ディレクトリーが Net Search Extender 索引ディレクトリーで検出されます。

5. 64 ビット Net Search Extender 索引の確認後、\*.32 ファイルは削除したり、バックアップ用の別の場所に転送したりできます。

これで索引のマイグレーションは完了です。64 ビットへのインスタンスのマイグレーションを実行できるようになりました。



---

## 第 7 部 計画に関する考慮事項

データベース機能を最大に引き出すには、フィーチャーを使用する前に計画を立てることが好ましいといえます。

**重要:** Net Search Extender は非推奨になりました。使用は推奨されません。また将来のリリースで除去される可能性があります。DB2 データベースに保管されているフルテキスト文書を SQL ステートメントおよび XQuery ステートメントを使用して検索する高速かつ用途の広い方法として、DB2 Text Search を使用してください。詳しくは、Net Search Extender から DB2 Text Search へのマイグレーションに関するトピックを参照してください。

Net Search Extender を最も効果的に使用するには、デプロイメントの前に計画を行うことが重要です。プラン作成には、データベース管理者、インターフェースおよびシステム的设计者、開発者など、複数のユーザー・グループが加わる場合があります。

以下のトピックでは、考慮すべき領域について説明しています。

- ディレクトリーのロケーションおよび索引ストレージ
- 表、列、および索引の名前
- 文書フォーマットおよびサポートされるコード・ページ
- Net Search Extender 索引ファイルの破損を防ぐ方法
- Outside-In フィルター・ソフトウェア
- ユーザー・ロール
- 増分更新のための拡張テキスト保持ステージング・インフラストラクチャー

Net Search Extender ベースのアプリケーションの開発について詳しくは、以下の関連トピックを参照してください。



---

## 第 13 章 Net Search Extender のディレクトリーのロケーションおよび索引ストレージ

Net Search Extender の索引に必要なディスク・スペースは、索引付けする必要のあるデータの量とタイプによって異なります。

ガイドラインとしては、1 バイト文書に索引付けする場合は、索引付けする文書のサイズの約 0.7 倍のディスク・スペースを確保する必要があります。2 バイト文書の場合は、索引付けする文書の合計サイズと同じディスク・スペースを確保してください。合計サイズには、アクティブ・データベースの外部に保管してあって、ユーザー定義関数によって取得されるデータも含める必要がある場合があります。

作業ディレクトリー内の一時ファイルに必要なスペース量は、索引ディレクトリー内の最終的な索引ファイルに必要なスペース量の 1.0 から 4.0 倍です。デフォルトの索引ディレクトリーは、DB2 インスタンス・ディレクトリー (通常、システムの /home パーティション内 (Linux および UNIX オペレーティング・システムの場合)、および C: ドライブ (Windows オペレーティング・システムの場合) ) にあるサブディレクトリーです。デフォルトの索引にはサイズの制限がある場合もあります。詳細については、283 ページの『データベース・レベル情報のビュー』を参照してください。

複数の大きい索引がある場合、特に索引の更新時または検索時に複数の索引に並行アクセスする場合は、各索引を別個のディスク装置に保管する必要があります。

各索引につき、対応する索引と作業ディレクトリーが同じファイル・システムまたはドライブに置かれる必要があります。デフォルトのロケーションを使用していない場合、索引および作業ディレクトリーが両方とも確実に同じファイル・システム上になるように、`create index` コマンドで両方のロケーションを必ず指定してください。コマンドで索引のディレクトリーのみを指定すると、作業ディレクトリーはデフォルト・パス (同じファイル・システム上ではない場合が多い) に作成されてしまいます。その逆も同じです。

パーティション・データベースの場合、テキスト索引は、索引を配置するために物理マシン上の単一ファイル・システムを引き続き使用します。大規模なデータベースの場合は、磁気ディスク制御機構 (RAID) 装置上のファイル・システムに索引と作業ディレクトリーを置いてください。これにより、テキスト索引の使用中に、入出力がボトルネックとなる可能性が最小化されます。

Net Search Extender の索引の作成、更新、および削除は、コマンド行インターフェースを使用して実行できます。



---

## 第 14 章 パーティション化された DB2 サーバーに関するリソースの考慮事項

パーティション環境で NSE を実行する場合、以下のリソースの考慮事項を念頭に置いてください。

- 複数のテキスト索引が並列で更新される場合、これが原因で、索引更新で入出力を集中的に行うフェーズにおいて、突発的な多量のディスク使用が発生することがあります。そのため、パーティションとそれに関連する NSE 索引ストレージのセットアップ、および索引更新のスケジューリングの際は、慎重な考慮が必要です。
- NSE 管理コマンドでは、索引の作成、ドロップ、および更新などの操作を行うためのプロセスが各パーティションに対して開始されます。更新処理は、リソースを消費する、実行時間の長い処理になることがあります。システムにおけるテキスト索引のスケジュール済み更新では、いくつかの索引更新が並行して実行されると、特定の時間に更新される索引の数と同じほど多くのプロセスがパーティションごとに生じることになる可能性があります。索引更新のスケジュールを適切に計画し、並行する索引更新の数を最小化することをお勧めします。
- Linux または UNIX オペレーティング・システムでは、コマンド `ulimit` と該当するオプションを使用して、処理リソース限度のサイズを表示または変更できます。DB2 Net Search Extender をパーティション・データベース環境で実行している場合は、コマンド `db2_all ulimit` と特定のオプションを使用して、すべてのデータベース・パーティションに適用するサイズを表示できます。Net Search Extender 更新索引コマンドを正常に実行するためには、ハード・オペレーティング・システムの `ulimit` を確認することが重要です。例えば、いずれかのパーティションでデータ・セグメント・サイズが不足すると、索引更新コマンドの実行中にエラーが発生する可能性があり、多くの場合テキスト索引のイベント・ビューに CTE0105 エラーがログに記録されます。



---

## 第 15 章 ストアード・プロシージャー検索のメモリー所要量

ストアード・プロシージャー検索にキャッシュを使用するときに必要なメモリーの量は、プラットフォームによって異なります。

ストアード・プロシージャー検索にキャッシュを使用すると、大量のメモリーが必要となります。また、以下のプラットフォームではメモリー所要量が異なります。

- AIX
- Windows
- Solaris
- Linux

---

### AIX (64 ビット) での Net Search Extender のメモリー所要量

Net Search Extender for AIX (64 ビット) を使用する前に、システムしきい値、共有メモリーしきい値およびスワップ・スペースを構成する必要があります。

システムしきい値の構成は以下のとおりです。

- コマンド `ulimit -a` を発行してシステムしきい値を確認します。
- 「unlimited」以外の値がある場合は、以下のステップに従います。
  - `root` でログオンします。
  - ファイル `/etc/security/limits` をバックアップしてから、ファイルを編集してハードしきい値を高くします。
  - 使用する DB2 インスタンス所有者に対して、すべての値を「unlimited」(値 -1) に設定します。

共有メモリーしきい値の構成は以下のとおりです。

- AIX では、共有メモリーしきい値を構成する必要はありません。

スワップ・スペースの構成は以下のとおりです。

- コマンド `lsattr -E -l sys0` を発行して、システム RAM サイズを入手します。
- `lspcs -a` コマンドを発行してスワップ・スペースのサイズを入手します。
- スワップ・スペース・サイズをシステム RAM 総量の 1.5 倍か 2 倍以上に設定するか、**CREATE INDEX** コマンドで指定する **MAXIMUM CACHE SIZE** パラメーターを使用します。SMIT ユーティリティを使用して、大きい方の数値を選択します。

---

### Windows (32 ビットおよび 64 ビット) での Net Search Extender ストアード・プロシージャーのメモリー所要量

仮想メモリー・ページング・ファイル・サイズは、ご使用の Windows マシンに合わせて設定する必要があります。

ページング・ファイルのサイズの調整は以下のとおりです。

- Windows 仮想メモリー・ページング・ファイル・サイズをシステム RAM の総量の 1.5 倍か 2 倍以上に設定するか、CREATE INDEX コマンドで指定する MAXIMUM CACHE SIZE パラメーターを使用します。大きい方の数値を選択してください。ページング・ファイル・サイズの変更については、Windows の資料を参照してください。

32 ビット Windows の場合は、約 1000 MB (1 GB = 1073741824 バイト) の最大キャッシュ・サイズを超えないことをお勧めします。

---

## Solaris (64 ビット) での Net Search Extender のメモリー所要量

システムしきい値、共有メモリーしきい値およびスワップ・スペースをチェックして、ご使用の Solaris マシンに合わせて構成する必要があります。

システムしきい値の構成は以下のとおりです。

- コマンド `ulimit -a` を発行してシステムしきい値を確認します。
- 次に、以下のステップに従います。
  - root でログオンします。
  - ファイル `/etc/system` をバックアップしてから、ファイルを編集してハードしきい値を高くします。
  - 以下の行を追加するか、以下の行が示された最小値以上に設定されていることを確認します。

```
rlim_fd_cur -> Default 64, recommended >= 1024
```

```
rlim_fd_cur_max -> Default 1024, recommended >= 4096
```

共有メモリーしきい値の構成は以下のとおりです。

- コマンド `sysdef -i` を発行して現在の設定を確認します。
- ファイル `/etc/system` を編集し、`set shmsys:shminfo_shmmax=0xffffffff` を使用して共有メモリー・サイズのしきい値を設定します。

場合により、以下のパラメーター値も増やす必要があります。

```
set shmsys:shminfo_shmmni=512
```

```
set shmsys:shminfo_shmseg=128、次にシステムをリブートします。
```

スワップ・スペースの構成は以下のとおりです。

- コマンド `/usr/sbin/prtconf` を発行して、システム RAM サイズを入手します。
- `swap -l` コマンドを発行してスワップ・スペース・サイズを入手します。
- スワップ・スペース・サイズをシステム RAM の総量の 1.5 倍から 2 倍以上に設定するか、CREATE INDEX コマンドで指定する MAXIMUM CACHE SIZE パラメーターを使用してください。大きい方の数値を選択してください。

スワップ・スペースを増やす方法については、Solaris システムの資料を参照してください。



約 2000 MB (2 GB = 2147483647 バイト) の最大キャッシュ・サイズを超えないことをお勧めします。

---

## Linux (32 ビットおよび 64 ビット) での Net Search Extender ストアード・プロシージャのメモリー所要量

Linux オペレーティング・システムについて、現在の共有リソース制限とシステムしきい値を検証する必要があります。

Linux 上で推奨されているカーネル・パラメーターについては、DB2 の資料を確認してください。

新規 Linux カーネルおよびディストリビューションの妥当性検査状況は頻繁に更新されます。サポートされる Linux ソフトウェア・レベルの最新情報を入手するには、<http://www.ibm.com/software/data/db2/linux/validate> を参照してください。

現在の共有リソース制限について調べるには、**ipcs -l** を使用してください。システムの制限について検査するには、**ulimit -a** コマンドを使用してください。



---

## 第 16 章 表、列、および索引の名前の考慮事項

すべての表名、列名、および索引名は、通常は大/小文字を区別しません。

Net Search Extender では、これらの名前を大/小文字混合で指定することもできます。Windows で、表、列、および索引の名前を大/小文字混合で指定する場合は、円記号 (¥) と二重引用符 (") の文字シーケンスで名前を入力する必要があります。例: ¥"DocTxt¥" など。



---

## 第 17 章 文書フォーマットおよびサポートされるコード・ページ

Net Search Extender は、検索対象のテキスト文書のフォーマット (またはタイプ) を知る必要があります。

この情報は、テキスト文書の索引作成時に必要になります。

Net Search Extender は、以下の文書フォーマットをサポートします。

**TEXT** プレーン・テキスト (例: フラット ASCII)、一般的にはマークアップなしのテキスト

### **HTML**

ハイパーテキスト・マークアップ言語

**XML** 拡張マークアップ言語

文書フォーマット XML は、XML データ・タイプの列におけるデフォルトの文書フォーマットであり、このデータ・タイプでサポートされている唯一の文書フォーマットです。

**GPP** 汎用パーサー・フォーマット (ユーザー定義タグ付きのフラット・テキスト)

### **Outside-In (INSO)**

PDF や他の一般的なテキスト・フォーマット・ツール (Microsoft Word など) からテキスト内容を抽出するためにフィルター・ソフトウェアを使用している場合にこのフォーマットを使用してください。

文書フォーマット HTML、XML、GPP、および Outside-In フィルター・フォーマットの場合、検索は文書の特定部分に制限することができます。

サポートされていない文書フォーマットを使用しているために Outside-In フィルターを使用できない場合は、独自のフィルター処理を行う、ユーザー定義関数 (UDF) を作成できます。この UDF は索引作成時に指定する必要があり、これを使用して、サポートされていないフォーマットからサポートされているフォーマットにデータを変換します。

サポートされているコード化文字セット ID (CCSID) のいずれかで保管されている文書の場合は、索引付けできます。サポートされるコード・ページのリストについては DB2 の資料を参照してください。

データベースのコード・ページをチェックするには、次の DB2 コマンドを使用してください。

```
db2 GET DB CFG for dbname
```

結果から、データベースのコード・ページに対応する値を取得してください。

整合性のために、DB2 は通常、文書のコード・ページをデータベースのコード・ページに変換します。ただし、DB2 データベース内のデータを、BLOB、FOR BIT DATA などのバイナリー・データ・タイプの列に保管する場合、DB2 はデータを変換しないため、文書は元の CCSID を保持します。

非互換のコード・ページを使用すると、テキスト索引の作成時または検索時に問題が発生する原因になる可能性があります。

---

## 第 18 章 Net Search Extender 索引ファイルの破損を防ぐ方法

索引ファイルが破損すると、そのことが索引のイベント表に記録されるエラー・メッセージによって示されます。そのエラー・メッセージには、さまざまな理由コード (破損の影響を受けたファイルによって異なる) のカーネル・エラーが報告されます。

例: 破損している索引に対して索引の更新を試みると、イベント・ビューに次に示すようなメッセージが表示されます。

101 CTE0101 検索エンジン操作が失敗しました。理由コード:  
"7"、"100001"、"0"、"カーネルの戻りコード: 17

破損した索引は修復できません。ドロップして再作成する必要があります。索引が破損しないようにするには、次に示す予防措置を講じてください。

- 実動システムでは、索引ディレクトリーと作業ディレクトリーについて、Net Search Extender 索引のデフォルト・ディレクトリー (インスタンス所有者のホーム・ディレクトリーにある) を使用するのではなく、それらのディレクトリーを CREATE INDEX ステートメントで指定してください。索引ディレクトリーと作業ディレクトリーの両方のために別個のファイル・システムを使用し、そのファイル・システムをモニターして、更新操作に必要な空きディスク・スペースが十分にあるようにしてください。更新に必要なとされるディスク・スペースは索引のサイズ (特に副次索引のサイズ)、および更新作業中に処理されるデータの量によって異なります。必要なディスク・スペースを見積もるには、更新操作中のピーク時の使用量をモニターすることが役立ちます。
- Windows のアクセス違反エラーによって生じる索引ファイル破損を防ぐには、次に示す手順に従ってください。
  - 自動バックアップ・プログラムからテキスト索引ディレクトリーおよび作業ディレクトリーを除外します。
  - アンチウィルスのスキャン・プログラムから索引ディレクトリーを除外します。
  - Windows インデックス サービスを、索引および作業ファイルを保管しているドライブについて無効にします。
- システムをシャットダウンする前に、次に示すコマンドを使用して Net Search Extender を正しく停止させます。
  - db2text control list all locks for database *database name*。保持されているロックがなくなるまでこのコマンドを繰り返します。詳細については、247 ページの『UPDATE INDEX コマンド』の説明を参照してください。
  - db2text stop





---

## 第 19 章 Outside-In フィルター・ソフトウェア

Net Search Extender は、サード・パーティーのフィルター・ソフトウェアをサポートします。

Oracle の Outside In トランスフォーメーション・テクノロジーとして知られるソフトウェアを使用すると、PDF ファイルや、一般的なテキスト・フォーマット・ツールの独自のフォーマットで作成された文書から、ネイティブ・アプリケーションを使用しないでテキスト内容を抽出できます。フォーマットの例としては、Microsoft Word や Lotus® Word Pro® などがあります。

Outside-In のライブラリーは、**UPDATE INDEX** の実行中に Net Search Extender によってプラグインとしてロードされます。ライブラリーは Net Search Extender の一部ではないので、別途インストールする必要があります。Net Search Extender が Outside-In ライブラリーを見つけられるようにする必要があります。

Outside-In ソフトウェアは、テキスト内容のみでなく、フィールドなどの構造情報も生成します。Net Search Extender は、Outside-In が生成する文書情報のうち、どの部分を索引に保管するかをカスタマイズすることができます。これを行うには、特定のタイプの文書モデルである Outside-In 文書モデルを適用する必要があります。

フィルター・フォーマットおよびサポートされるプラットフォームのリストを確認するには、Oracle の Web サイト (<http://www.oracle.com>) を参照してください。



---

## 第 20 章 ユーザー・ロール

ユーザー・ロールには DB2 インスタンス所有者、データベース管理者、およびテキスト表所有者があり、それぞれ独自の管理権限を持っています。

### DB2 インスタンス所有者

DB2 インスタンス所有者は、DB2 Net Search Extender のインスタンス・サービスの開始と停止を実行でき、ロッキング・サービスをコントロールできます。さらに、DB2 インスタンス・ユーザーは、各使用可能データベースの DBADM 権限を付与されます。これにより、Net Search Extender によって行われるすべてのデータベース変更の集中制御が可能になります。

#### 必須 DB2 権限

SECADM は、インスタンス所有者に DATAACCESS 特権を持つ DBADM 権限を付与しなければなりません。これらの権限は、DB2 Net Search Extender 管理コマンドを実行するための前提条件になります。

#### 必須ファイル・システム権限

すべてのテキスト索引ディレクトリーの読み取りおよび書き込み、モデル・ファイルへの読み取りアクセス。

#### インスタンス所有者用のコマンド

**DB2TEXT START**、**DB2TEXT STOP**、**DB2TEXT CONTROL**、および **DB2EXTHL**

これらのコマンドは、サーバーでだけ使用できます。パーティション・データベース環境の場合は、構成されているどのノードでも使用できます。各コマンドは、コマンドを実行中のユーザーが DB2 インスタンス所有者であるかどうかをチェックします。インスタンス所有者であるユーザーが fenced ユーザー ID を使用してストアード・プロシージャおよび UDF を実行する場合は、索引ディレクトリー内のすべてのファイルに対する読み取りおよび書き込みアクセス（このディレクトリー・パス全体に対する読み取りアクセスを持つ）をこの fenced ユーザーが所有する必要があります。fenced ユーザー ID によって作成されたファイルに対してインスタンス・ユーザー ID の正しいアクセスを付与し、インスタンス・ユーザー ID によって作成されたファイルに対して fenced ユーザー ID の正しいアクセスを付与するために、fenced ユーザー ID とインスタンス・ユーザー ID が同じ 1 次グループのメンバーでなければならないことに注意してください。正しいグループ・メンバーシップおよびファイル許可を割り当ててください。

インスタンス所有者に加え、インスタンス所有者と同じ 1 次グループのユーザーであれば、**DB2TEXT START**、**DB2TEXT STOP**、**DB2TEXT CONTROL** および **DB2EXTHL** を実行することができます。

### データベース管理者

データベース管理者は、Net Search Extender で使用するデータベースを使用可能または使用不可にできます。

#### 必須 DB2 権限

DBADM

## データベース管理者用のコマンド

**DB2TEXT ENABLE DATABASE、および DB2TEXT DISABLE DATABASE。**

### テキスト表所有者

テキスト表所有者は索引の作成、ドロップおよび変更を行えます。このユーザーは、(読み取りおよび書き込みアクセスを所有することによって) 索引のロケーションの制御、およびフルテキスト索引の更新の制御ができる必要があることに注意してください。

#### 必須 DB2 権限および特権

テキスト表の所有者。

### テキスト表所有者用のコマンド

**DB2TEXT CREATE INDEX、DB2TEXT DROP INDEX、DB2TEXT ALTER INDEX、DB2TEXT ACTIVATE CACHE、DB2TEXT DEACTIVATE CACHE、DB2TEXT UPDATE INDEX、DB2TEXT CLEAR EVENTS、および DB2EXTTH。**

DB2 インスタンス所有者のユーザー ID を使用した場合、コマンドの実行が不完全になることに注意してください。したがって、テキスト索引を作成または変更する前に、インスタンス所有者に必要なファイル・システム・アクセスを付与してください。必須権限に関する詳細については、各コマンドごとにリストされていますので、219 ページの『第 57 章 テキスト表所有者用の管理コマンド』を参照してください。

---

## 第 21 章 増分更新のための拡張テキスト保持ステージング・インフラストラクチャー

バージョン 9.7 では、トリガーによって認識されない変更を通常のログ表に取り込めるようにするステージング・インフラストラクチャーを追加する構成オプションが使用可能です。

バージョン 9.7 では、トリガーによって認識されない変更を通常のログ表に取り込めるようにするステージング・インフラストラクチャーを追加する構成オプションが使用可能です。このオプションを有効にすると、トリガーによって更新は通常のログ表に取り込まれ、挿入と削除はテキスト保持ステージング表に取り込まれます。

この構成オプションは、デフォルトで、範囲パーティション表で有効になっており、非パーティション表では無効になっています。テキスト保持ステージング・インフラストラクチャーを追加することは、可用性やさまざまなデータベース操作に対する基本表の状況に重要な影響を与えます。

テキスト保持ステージング・インフラストラクチャーの影響は、据え置きリフレッシュでマテリアライズ照会表 (MQT) を追加することに似ています。テキスト保持インフラストラクチャーは MQT にデータを維持しませんが、ステージング表によって、MQT ステージング表の場合に見られる動作に対応する動作が生じます。

例えば、LOAD 挿入の後には、基本表に対する後続のデータベース操作を使用可能にするために、表での保全性処理が必要になります。

表が表中のすべての行に影響するデータベース・コマンドによってのみ更新される場合 (例えば **LOAD REPLACE** の場合など)、拡張ステージング・インフラストラクチャーを追加しても効果がありません。その代わりに、索引を再作成したほうがよいでしょう。



---

## 第 8 部 Net Search Extender の管理





---

## 第 22 章 Net Search Extender インスタンス・サービス

DB2 Net Search Extender インスタンス・サービスは、ロッキング・サービスと更新サービスから構成されています。

DB2 Net Search Extender インスタンス・サービスは、次のサービスから構成されています。

- ロッキング・サービス
- 更新サービス

Windows 上の DB2 Net Search Extender インスタンス・サービスは、Windows サービスとして表されます。非パーティション DB2 インスタンスでは、DB2 インスタンスごとに次のようなサービス名を持つ 1 つのサービスが存在します。

DB2EXT - *instance\_name*

パーティション DB2 インスタンスでは、DB2 インスタンスのパーティションごとに次のようなサービス名を持つ 1 つのサービスが存在します。

DB2EXT - *<instance\_name>*[-*<nodenum>*]

次のトピックでは、DB2 Net Search Extender インスタンス・サービスの開始と停止の方法について、さらにロッキング・サービスと更新サービスについて詳しく説明します。

- NSE インスタンス・サービスの開始および停止
- ロッキング・サービス
- 更新サービス
- NSE 情報カタログ



---

## 第 23 章 Net Search Extender インスタンス・サービスの開始および停止 (コマンド行を使用)

テキスト索引を保守して、文書を検索できるようにするには、Net Search Extender インスタンス・サービスを開始する必要があります。

### このタスクについて

パーティション・データベースと共に使用される DB2 インスタンスでは、標準的な Windows での方法を使用する代わりに、**db2text start** および **db2text stop** を使用して Net Search Extender インスタンス・サービスを開始および停止することを強くお勧めします。これにより、インスタンス・サービスの開始および停止が確実に正しい順序で行われます。

### 注:

- 1 つの DB2 インスタンスあたりに 1 つの Net Search Extender インスタンス・サービスがなければなりません。ロックング・サービスは、このインスタンスで有効なすべてのデータベースのロックを保守します。
- Windows 上の DB2 Net Search Extender インスタンス・サービスは、Windows サービスとして表されます。パーティション DB2 インスタンスでは、DB2 インスタンスのパーティションごとに 1 つのサービスが存在します。

### 手順

- インスタンス・サービスを開始するには、DB2 インスタンスの所有者ユーザー ID (UNIX オペレーティング・システムのみ) にログオンして、次のコマンドを入力します。

```
db2text start
```

- インスタンス・サービスを停止するには、次のコマンドを入力します。

```
db2text stop
```



---

## 第 24 章 Net Search Extender ロッキング・サービス

Net Search Extender ロッキング・サービスは、読み取りおよび書き込みプロセスが、互いに干渉するのを防止するために使用されます。

Net Search Extender を開始すると、ロッキング・サービスは自動的に開始します。ロッキング・サービスは、Net Search Extender のテキスト索引への同時アクセスの同期をとるために必要です。

ロッキング・サービスは、2 つの処理で同時にテキスト索引を変更しようとしたり、別の処理でテキスト索引を変更中に、他の処理で同じテキスト索引データを読み取ったりすることがないようにします。したがって、大半の処理で開始前にテキスト索引に対するロックが要求され、処理が完了すると再びロックが解放されます。

Net Search Extender テキスト索引のロッキング・サービスと DB2 表へのアクセスを制御する DB2 ロックとを混同しないように注意してください。

---

### ロッキング・サービスの使用

Net Search Extender には、索引への同時アクセスを制御する、さまざまなタイプのロックがあります。

検索要求の場合のようにテキスト索引を読み取るのみなのか、索引作成の場合のようにテキスト索引への変更を計算してからファイルに書き込む必要があるのかによって、異なるロックが使用されます。

**db2text start** 時に、ロッキング・サービスは自動的に開始します。テキスト索引に対する次のタイプのロックがあります。

**S-lock** 共有読み取り専用アクセス用。例: 検索要求など。

**U-lock** 同時読み取りアクセスに対する索引への変更 (更新) を計算している間の読み取りおよび書き込みアクセス用。

**X-lock** 変更が実際に索引に書き込まれる間の短時間の排他的読み取り/書き込みアクセス用。

#### **IX-lock**

更新処理が X-lock を待機中に新規の S-locks を阻止するために意図された排他的読み取り/書き込みアクセス用。

DB2 インスタンスあたりに 1 つの Net Search Extender ロッキング・サービスが存在します。ロッキング・サービスは、複数のデータベースのロックを保持します。

ロッキング・サービス構成ファイルは、db2ext1m.cfg です。UNIX システムの場合は *instance\_owner\_home*/sql1lib/db2ext 上に保管され、Windows の場合は *sql1lib\%DB2INSTANCE%\db2ext* に保管されます。

構成ファイルの変更は、**db2text start** 時に Net Search Extender インスタンス・サービスが開始されて初めて有効となります。ユーザーは以下の値を設定できません。

- データベースの最大数
- データベースあたりの索引の最大数
- 索引あたりの許可されたロック (並行ユーザー) の最大数
- ロックを取得するための待機時間および試行回数

構成ファイルのデフォルト値は、次のとおりです。

```
<default
 maxDbs = " 8"
 maxIdxPerDb = " 50"
 maxLocksPerIdx = "100"

 sWait = " 50"
 uWait = " 500"
 xWait = " 500"

 sAttempt = "50"
 uAttempt = "10"
 xAttempt = "60"

 latchTimeout = "80"
/>
```

構文は `<default attribute=value.../>` で、属性とその意味は、次のとおりです。

#### **maxDbs**

ロックング・サービスで処理できるデータベースの数 (integer 値 >1)。

#### **maxIdxPerDb**

ロックできるデータベースあたりの索引数 (integer 値 >1)。この値は、すべてのデータベースに対して同一です。

#### **maxLocksPerIdx**

索引上に同時に存在できるロックの数 (integer 値 >1)。この値は、すべての索引に対して同一です。

共有メモリーの使用量は、前述の 3 つの「最大」値の積に比例します。共有メモリーの使用量が必要以上に大きくならないように、使用している値が DB2 インスタンスの実際の構成に適していることを確認してください。構成ファイル内で `maxDbs`、`maxIdxPerDb`、または `maxLocksPerIdx` の値を前述のデフォルト値よりも大きくする場合は、メモリーが十分にあることを確認してください。パーティション・データベース環境を使用している場合、`maxIdxPerDb` と `maxLocksPerIdx` の値については、その設定がすべてのパーティションで使用されるため、特に注意してください。これは、物理マシン上の 1 つの DB2 インスタンスに対して複数の論理パーティションが定義されている場合、メモリー所要量を決定する上で特に重要となります。

#### **sWait/sAttempt**

S-lock を要求する際に `sAttempt` は、ロックが即時に付与されない場合に試行される数です。 `sWait` はこれらの 2 つの試行の間の待ち時間 (integer 値 >1) です。これらのパラメーターは、IX-locks にも適用されます。

### **uWait/uAttempt**

U-lock を要求する際に uAttempt は、ロックが即時に付与されない場合に試行される数です。 uWait はこれらの 2 つの試行の間の待ち時間 (integer 値 >1) です。

### **xWait/xAttempt**

X-lock を要求する際に xAttempt は、ロックが即時に付与されない場合に試行される数です。 xWait はこれらの 2 つの試行の間の待ち時間 (integer 値 >1) です。

### **latchTimeout**

これは、インターバル・ロッキング・サービスの追加の待ち時間です。ロックのための待ち時間の合計を決定するには、次の計算を使用します。

```
waiting time = # attempts * (# waits + (2 * # latchTimeout))
```

待機、試行、およびタイムアウト・パラメーターのデフォルト値を変更しないことを強くお勧めします。待ち時間は、ミリ秒単位で計算されます。試行のたびに latchTimeout 値は、待ち時間の合計に追加される際に 2 倍にされることに注意してください。

---

## **ロック・スナップショットの表示**

ロック・スナップショットを表示できるコマンドはいくつかあります。初めてテキスト索引がロックされると、メモリーはデータベースおよびロッキング・サービスのテキスト索引の両方のために予約されます。

### **このタスクについて**

次のいずれかのコマンドを使用すると、ロック・スナップショットを表示できます。

- 単一のテキスト索引の場合  
`db2text CONTROL LIST ALL LOCKS FOR DATABASE mydatabase INDEX myindex`
- データベースの、ロックされたすべてのテキスト索引の場合  
`db2text CONTROL LIST ALL LOCKS FOR DATABASE mydatabase`

実際にロックされている索引だけがリストされることに注意してください。

初めてテキスト索引がロックされると、メモリーはデータベースおよびロッキング・サービスのテキスト索引の両方のために予約されます。さらにテキスト索引がロックされる場合は、メモリーはロッキング・サービスのこれらのテキスト索引にも割り振られます。このメモリーは、テキスト索引をドロップしたか、データベースを使用不可にした場合、あるいは Net Search Extender サービスを再始動した場合にだけ再び解放されます。つまり、現在、ロックが設定されていなくても、ロッキング・サービスでは、テキスト索引またはデータベースはメモリーを消費することになります。

コマンド「`db2text CONTROL CLEAR ALL LOCKS`」は、データベースまたは索引上のすべてのロックを強制的に解放します。このコマンドの使用の詳細に関しては、207 ページの『CONTROL コマンド』を参照してください。 **CLEAR ALL LOCKS** コマンドを使用する場合、索引指定子を必ず使用してください。 ロックをクリアしたい

索引にアクティブな索引更新がないことを必ず確認してから、このコマンドを使用してください。現在更新されている索引のロックをクリアすると、索引が破損し、索引の完全な再作成が必要となります。このコマンドは、データベースまたは索引に割り振られたメモリーを解放しないことに注意してください。メモリーを解放するには、索引をドロップするか、データベースを使用不可にする、または Net Search Extender サービスを再始動する必要があります。アクティブ索引の更新処理中は、ロックを解放しないでください。



---

## 第 25 章 更新サービス

表の変更と索引の更新は同期されません。索引更新処理は手動で開始するか、指定された間隔で自動的に開始するようにスケジュールに入れることができます。

更新サービスがこの機能を提供します。更新サービスは **db2text start** 中に開始されます。

索引の作成の際に、次のコマンドを使用することにより、更新サービスが、索引の更新が必要かどうかをどの程度の頻度でチェックするかを指定できます。

```
db2text create index DB2EXT.TITLE for text on DB2EXT.TEXTTAB (TITLE)
 UPDATE FREQUENCY D(1,3) H(0,12) M(0) update minimum 5
```

この例では、毎週の月曜日と水曜日の午後 12 時と午前 12 時に更新サービスが起動し、索引 `db2ext.title` に実行するべき作業が存在するかどうかをチェックします。この例では、`DB2EXT.TITLE` に対する少なくとも 5 回の変更があって初めて、自動索引更新によってテキスト索引をデータベースに同期する処理が開始されます。

パーティション・データベース環境の場合、個々の更新サービスは、各ノードに対して開始されます。複数のテキスト索引が並列で更新される場合、これが原因で、索引更新で入出力を集中的に行うフェーズにおいて、突発的な多量のディスク使用が発生することがあります。そのため、索引更新をスケジュールおよび実行する際には、慎重に考慮する必要があります。

### 注:

索引更新処理のインターバルを非常に短く設定すると、システム・パフォーマンスが低下します。各更新処理の間に処理されると予想される変更の量および処理に要する時間を考慮する必要があります。同様に、自動索引更新の間に処理する索引の数についても考慮する必要があります。各索引更新の間のインターバルが、スケジュールされている次の更新の開始までに更新を終了できるように十分長いことを確認してください。複数の索引の更新を同時に開始するようにスケジュールしていないことも確認してください。

テキスト保持ステージング・インフラストラクチャーがテキスト索引に対して構成されている場合、**RESET PENDING** コマンドを実行して、ステージング表が保留モードではない状態にしてください。



---

## 第 9 部 開発: テキスト索引の作成と保守

作業を進める前に、テキスト索引の作成と保守に関して、検討しておくべき重要ないくつかの項目があります。

**重要:** Net Search Extender は非推奨になりました。使用は推奨されません。また将来のリリースで除去される可能性があります。DB2 データベースに保管されているフルテキスト文書を SQL ステートメントおよび XQuery ステートメントを使用して検索する高速かつ用途の広い方法として、DB2 Text Search を使用してください。詳しくは、Net Search Extender から DB2 Text Search へのマイグレーションに関するトピックを参照してください。

このセクションでは、テキスト索引の作成と保守について説明します。以下の項目を扱います。

- **db2text** コマンドの紹介
- データベースをテキスト検索可能にする
- さまざまなデータ・タイプのテキスト索引の作成
- DB2 レプリケーションで、増分索引更新による、ニックネームのテキスト索引を作成する
- ストアード・プロシージャ検索で使用できるテキスト索引の作成
- ビューに基づくテキスト索引
- 索引の保守
- 範囲パーティション表でのテキスト索引の作成

この章では、発生する可能性があるコード・ページの問題の回避、およびパフォーマンス考慮事項についても説明します。

テキスト索引を作成する前に、57 ページの『第 7 部 計画に関する考慮事項』にある前提条件を満たしていることを確認してください。**db2text start** コマンドを使用して Net Search Extender インスタンス・サービスを開始してあることも確認してください。



---

## 第 26 章 データベースを使用可能にする

**ENABLE DATABASE FOR TEXT** コマンドは、データベースを Net Search Extender で使用するために準備します。

### 始める前に

DBADM 権限が必要です。

### このタスクについて

検索対象のテキストの列がある表を含んでいるデータベースごとに 1 回、この作業を実行します。

また、**ENABLE DATABASE FOR TEXT** コマンドは、265 ページの『第 60 章 SQL スカラー検索関数および SQL 表値関数』で説明されている Net Search Extender の検索関数および検索プロシージャの登録も行います。

データベースを使用可能にすると、コマンドはさらに次の表およびビューも自動作成します。

#### **db2ext.dbdefaults**

索引、テキスト、および処理特性のデータベース・デフォルト値を格納します。

#### **db2ext.textindexformats**

サポートされるフォーマットおよび使用される現在アクティブなモデル・ファイルのリストを格納します。

#### **db2ext.indexconfiguration**

索引構成パラメーターを格納します。

#### **db2ext.textindexes**

すべてのテキスト索引を追跡するカタログ・ビュー。

データベースがいったん使用可能になると、使用不可にするまでは使用可能の状態が続きます。



---

## 第 27 章 データベースを使用不可にする

データベースでテキスト検索を行う必要がなくなった場合は、**DISABLE DATABASE FOR TEXT** コマンドでデータベースを使用不可にします。

### 始める前に

データベースには DBADM 権限が必要です。

### このタスクについて

Net Search Extender がデータベースを使用できるように準備するとき、管理上の変更を行う場合があります。このセクションでは、このプロセスを元に戻す場合に役立つ関数について説明します。

接続サブシステムを使用不可にするには、次のコマンドを使用します。

```
db2text DISABLE DATABASE FOR TEXT
```

データベースを使用不可にすると、コマンドによって以下のオブジェクトが削除され、さらに既存のすべてのテキスト索引がドロップされます。

- サーバーを使用可能にした際に作成された Net Search Extender カタログ・ビューおよび表。
- Net Search Extender の SQL 関数 (UDF) の宣言。

**DISABLE DATABASE FOR TEXT** コマンドがエラーを戻した場合に、(たとえ索引がまだ使用中であっても) 使用不可にする場合は、次のコマンドを使用してください。

```
db2text DISABLE DATABASE for text force
```

**注:** データベースにテキスト索引が定義されている場合に、そのデータベースを使用不可にしようとする、失敗します。これらの索引を 1 つ 1 つ除去した上で、何か問題が生じるかどうかを検査するようお勧めします。 **DISABLE DATABASE FOR TEXT FORCE** コマンドを使用した場合に保証されるのは、データベース内の Net Search Extender カタログ表が削除されるということに過ぎません。既存のテキスト索引がテキスト保持ステージング・インフラストラクチャーを使用する場合、強制オプションは適用されません。

しかし、完全にドロップできない索引があった場合は、手作業でクリーンアップしなければならないリソースがまだあるかもしれません。以下のものが含まれます。

- 索引ディレクトリー、作業ディレクトリー、およびキャッシュ・ディレクトリー内のファイル
- ctedem.dat 内のスケジューラー項目
- レプリケーション・キャプチャー・オプションを使用して索引を作成した場合には、リモート・データベースの表にある、IBMSNAP\_SIGNAL、IBMSNAP\_PRUNE\_SET、および IBMSNAP\_PRUNCNTL の各項目を手動で削除しなければなりません。これらの項目は、`APPLY_QUAL='NSEDB2' || instance_name` および `TARGET_SERVER= database_name` 条件を使用して簡単に識別できます。

次の例では、インスタンスは DB2、データベースは SAMPLE です。

```
DELETE FROM <ccSchema>.IBMSNAP_SIGNAL
WHERE SIGNAL_INPUT_IN IN
 (SELECT MAP_ID FROM <ccSchema>.IBMSNAP_PRUNCNTL
 WHERE APPLY_QUAL= 'NSEDDB2' AND TARGET_SERVER= 'SAMPLE');
```

```
DELETE FROM <ccSchema>.IBMSNAP_PRUNCNTL
WHERE APPLY_QUAL= 'NSEDDB2' AND TARGET_SERVER= 'SAMPLE';
```

```
DELETE FROM <ccschema>.IBMSNAP_PRUNE_SET
WHERE APPLY_QUAL= 'NSEDDB2' AND TARGET_SERVER= 'SAMPLE';
```



---

## 第 28 章 テキスト索引の作成

検索対象のテキストを含んでいる列ごとに 1 回、**CREATE INDEX FOR TEXT** コマンドを発行します。

### 始める前に

次のいずれかの権限レベルが必要です。

- 索引表に対する **CONTROL** 特権
- 表に対する **INDEX** 特権、およびデータベースに対する **IMPLICIT\_SCHEMA** 権限または索引スキーマに対する **CREATEIN** 特権のいずれか
- **DBADM** 権限

### このタスクについて

すべてのデータ・タイプにテキスト索引を作成できます。ただし、以下のデータ・タイプでは要件が異なります。

- バイナリー・データ・タイプ
- サポートされないデータ・タイプ

また、ストアド・プロシージャ用のテキスト索引の作成に関しても要件は異なります。

テキスト索引を作成すると、拡張テキスト保持ステージング・インフラストラクチャーがテキスト索引に使用可能かどうかによって、以下のオブジェクトが **Net Search Extender** によって自動的に作成されます。

**注:** 索引が範囲パーティション表で作成される場合は、**ADMINISTRATION TABLES IN** 節を必ず指定してください。詳細については、**CTE0150E** を参照してください。

- 通常のログ・インフラストラクチャーを使用する場合

**ログ表** ログ表は、ユーザー表内の行に対するすべての変更を追跡します。

**RECREATE INDEX ON UPDATE** オプションを選択する場合、またはレプリケーション・キャプチャー表を使用する場合には、ログ表は作成されないで注意してください。

#### イベント表

この表は、テキスト索引の更新時におけるすべての更新および潜在的な問題に関する情報を収集します。

#### ユーザー表のトリガー (初期更新で追加された)

トリガーは、ユーザー表内の文書が追加、削除、または変更される度に、ログ表に情報を追加します。この情報は、スケジュールされている次の索引更新、または手動での索引更新時に索引を同期化するために必要です。

トリガーが作成されるのは、ログ表を作成した場合で、ビューでもニックネーム表でもなく基本表でテキスト索引を作成した場合のみであることに注意してください。

- 拡張ログおよびステージング・インフラストラクチャーを使用する場合

**ログ表** この表は、文書に対する更新を記録します。

**補助ステージング表**

この表は、挿入と削除を記録します。

**イベント表**

この表は、テキスト索引の更新時におけるすべての更新および潜在的な問題に関する情報を収集します。

**ユーザー表の更新トリガー (初期更新の間に追加された)**

索引列の文書が更新されると、更新トリガーは、影響を受けた行の主キーをログ表に追加します。

パフォーマンスおよびディスク・スペースを最適化するために、**CREATE INDEX** コマンドには表に対して異なる表スペースを指定するオプションがあります。

**注: LOAD** コマンドを使用して文書をインポートした場合は、トリガーが起動されないため、ロードされた文書の増分索引付けを通常のインフラストラクチャーでは実行できません。この場合、**DB2 IMPORT** コマンドを使用することをお勧めします。このコマンドはトリガーをアクティブ化します。

拡張テキスト保持インフラストラクチャーがテキスト索引に対して構成されている場合、ロード挿入操作によって挿入された文書は補助ステージング表に取り込まれ、増分索引付けも可能になります。

## 例

次の例では、表 `htmltab` 内のテキスト列 `HTMLFILE` のテキスト索引を作成します。

```
db2text create index DB2EXT.HTMLIDX for text on DB2EXT.HTMLTAB
 (HTMLFILE) format HTML
```

この表に主キーが存在する必要があります。

索引作成のデフォルト値は、`db2ext.dbdefaults` ビューから取得されます。

**CREATE INDEX** が行った変更を元に戻すには、**DROP INDEX** コマンドを使用します。詳細については、125 ページの『第 37 章 テキスト索引のドロップ』を参照してください。

作成された索引にテキスト列のデータを取り込むには、次のコマンドを使用します。

```
db2text update index DB2EXT.HTMLIDX for text
```

**db2text update** コマンドを使用して、テキスト索引を表と同期した後にのみ、正常に文書を検索できることに注意してください。

索引作成時にエラーが発生した場合は、`index update event` 行がイベント表に追加されます。これは、例えば、索引作成のためにキューに格納された文書が見つからない場合、または文書フォーマットが無効な場合に発生します。詳細については、290 ページの『イベント・ビュー』の説明を参照してください。

## 次のタスク

### 注: 検索のサマリー

索引作成中に選択するオプションによって、以下のようなさまざまな検索を行うことができます。

- SQL スカラー検索関数は、ビューに関して作成されたテキスト索引以外のすべてのテキスト索引を処理します。
- ストアード・プロシージャ検索関数は、キャッシュを使用して作成されたテキスト索引のみを処理します。
- SQL 表値関数は、ビューに作成されたテキスト索引を含むすべてのテキスト索引を処理します。

---

## バイナリー・データ・タイプのテキスト索引の作成

BLOB または FOR BIT DATA などのバイナリー・データ・タイプの列にデータを保管する場合、DB2 データベース・システムはデータを変換しません。

### このタスクについて

文書は元のコード・ページ (CCSID) を保持しており、これによって、テキスト索引を作成するときに問題が生じる可能性があります。これは、2 つの異なるコード・ページが存在することになるためです。よって、データベースのコード・ページを使用するか、**CREATE INDEX** コマンドで指定したコード・ページを使用するかを決定する必要があります。

この問題を回避するために、テキスト索引の作成時にコード・ページを指定してください。

```
db2text CREATE INDEX db2ext.comment FOR TEXT ON db2ext.texttab (comment)
 CCSID 1252
```

コード・ページを指定しなかった場合には、次のコマンドを実行して、索引を作成するために使用された CCSID を調べてください。

```
db2 SELECT ccsid FROM db2ext.textindexes WHERE INDSHEMA = 'DB2EXT'
 and INDNAME = 'COMMENT'
```

コード・ページが異なる複数の文書を 1 つのテキスト索引に含めることはサポートされていないことに注意してください。DB2 データベース製品が文書コード・ページの設定値を変換する方法については、「グローバル化・ガイド」を参照してください。

文字データ・タイプ上に索引を作成する場合は、この問題は発生しません。文字データ・タイプの場合は、**CCSID** パラメーターを指定しないでください。

---

## サポートされないデータ・タイプのテキスト索引の作成

索引を作成するには、テキスト列が所定のデータ・タイプでなければなりません。

## このタスクについて

索引を作成するには、テキスト列が以下のいずれかのデータ・タイプでなければなりません。

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

## 手順

ユーザー定義タイプ (UDT) など、サポートされていないデータ・タイプの列に文書がある場合は、次のようにする必要があります。

1. 入力としてこのユーザー・タイプを使用し、出力タイプとしていずれかの有効なデータ・タイプにキャストする変換関数を指定します。
2. 索引作成時にこの変換関数の名前を指定します。詳しくは、227 ページの『CREATE INDEX コマンド』を参照してください。

## 例

圧縮テキストを表に保管することにします。

1. 対話式 SQL セッションで、テキストのユーザー定義タイプ (UDT) を作成します。

```
db2 "CREATE DISTINCT TYPE COMPRESSED_TEXT AS CLOB(1M)"
```

2. 表を作成し、テキストを表に挿入します。

```
db2 "CREATE TABLE UDTTABLE (author VARCHAR(50) not null,
 text COMPRESSED_TEXT, primary key (author))"
db2 "INSERT ..."
```

3. 仮に `uncompress` という名前でユーザー定義関数 (UDF) を作成します。これは、タイプ `COMPRESSED_TEXT` の値を受け取り、対応する圧縮解除したテキストを、例えば `CLOB(10M)` 値として戻します。

4. 以下のようにテキスト索引を作成して、`uncompress` UDF を指定します。

```
db2text "CREATE INDEX UDTINDEX for text ON UDTTABLE
 (uncompress(text))
 ..."
```

---

## DB2 レプリケーションで、増分索引更新による、ニックネームのテキスト索引を作成する

レプリケーション・キャプチャー表を使用してニックネームのテキスト索引を作成するには、事前に所定のステップを実行しなければなりません。

## 手順

レプリケーション・キャプチャー表を使用してニックネームのテキスト索引を作成するには、事前に以下のステップを実行しなければなりません。

1. すべてのサーバー定義およびラッパー定義とともに、DB2 フェデレーテッド・データベースをセットアップします。
2. リモート・サーバーで、レプリケーション・コントロール表およびキャプチャー・プログラムをセットアップします。ニックネームのソース表が存在するのはここです。DB2 データベース・システムがニックネームを自動的に作成しない場合は、以下の表に関する 1 つのスキーマ名を使用して、フェデレーテッド DB2 データベースにニックネームを作成しなければなりません。

- IBMSNAP\_SIGNAL
- IBMSNAP\_PRUNE\_SET
- IBMSNAP\_PRUNCNTL
- IBMSNAP\_REGISTER
- IBMSNAP\_REG\_SYNC (DB2 以外のデータベース・サーバー・リモート・ソースのみ)

このステップを完了すると、レプリケーション・コントロール表のニックネームを、フェデレーテッド DB2 データベース上の 1 つの「キャプチャー・コントロール・スキーマ」の下のニックネームとして使用できるようになります。このスキーマ名は、**DB2TEXT CREATE INDEX** コマンドにとって重要です。

3. 表をレプリケーション・ソースとして登録します。
4. 登録ステップで DB2 データベース・システムによってニックネームが自動的に作成されない場合は、フェデレーテッド・データベースのレプリケーション・キャプチャー表のためにニックネームを作成してください。レプリケーション・キャプチャー表は、変更データ (CD) 表か整合変更データ (CCD) 表のいずれかです。このニックネームは、**DB2TEXT CREATE INDEX** コマンドのパラメーターです。

列名 IBMSNAP\_OPERATION、IBMSNAP\_COMMITSEQ、IBMSNAP\_INTENTSEQ、および主キー列の名前は変更できません。

5. DB2 レプリケーション・ソースを使用する場合は、キャプチャー・プログラムが実行中であることを確認してください。キャプチャー・プログラムでコールド・スタートを使用しないでください。使用すると、**APPLY\_QUAL LIKE 'NSE%'** に関する IBMSNAP\_SIGNAL 表内のすべての行を再び挿入しなければなりません。以下の SQL ステートメントで、これがどのように行われるかを調べることができます。

```
INSERT INTO capture_control_schema.IBMSNAP_SIGNAL
SELECT CURRENT_TIMESTAMP, 'CMD', 'CAPSTART', MAP_ID, 'P'
FROM capture_control_schema.IBMSNAP_PRUNCNTL
WHERE APPLY_QUAL LIKE 'NSE'
```

6. レプリケーションを使用してニックネームに対してテキスト索引を作成するには、以下の例を使用します。

```
DB2TEXT
CREATE INDEX indexname FOR TEXT ON nickname (text_column)
REPLICATION CAPTURE TABLE capture_nickname
CONTROL TABLE SCHEMA capture_control_schema
```

## ストアード・プロシージャー検索で使用できるテキスト索引の作成

ストアード・プロシージャー検索用のテキスト索引を作成する際には、まず所定のパラメーターを決定し、計算しなければなりません。

### このタスクについて

表から取得してユーザーに提示するデータのサブセットがあらかじめわかっている、結果リスト全体ではなく、トップ・ランクの検索結果のみを対象とする場合は、ストアード・プロシージャー検索を使用できます。ストアード・プロシージャー検索の場合は、**CREATE INDEX** コマンドの実行時にキャッシュ・オプションを指定する必要があります。キャッシュ索引を使用すると、指定したすべてのデータをメイン・メモリーに移動することによって、時間とリソースを大量に必要とする表からの物理読み取り操作を回避して、照会時のパフォーマンスを向上させることができます。

初めてキャッシュ索引を更新する前に、表に文書が格納されていることを確認して文書が設定されていない表を元に索引を更新してしまわないようにしてください。こうすることで、索引作成のパフォーマンスが向上するとともに、キャッシュ・メモリー要件を正しく予測できるようになります。

ストアード・プロシージャー検索を使用すると、文書に関連した定義済みデータを迅速に戻すことができます。これを定義するには、**CREATE INDEX** コマンドで、キャッシュ表オプションを使用します。次に、**ACTIVATE CACHE** コマンドによって、指定したデータをメモリー・キャッシュに移動します。

ストアード・プロシージャー検索用のテキスト索引を作成する際には、次のパラメーターを決定し、計算しなければなりません。

- キャッシュのタイプ (一時または永続)。
- 索引更新のタイプ (自動および増分、または更新ごとに再作成)
- Net Search Extender で使用できる最大メモリー量。MAXIMUM\_CACHE\_SIZE を使用します。
- 以後の文書更新に必要な空きメモリー量。PCTFREE を使用します。これは、増分更新の場合のみです。

以下のキャッシュのタイプを使用できます。

#### 一時キャッシュ

これは、**DB2TEXT ACTIVATE CACHE** コマンドごとに再作成され、DB2 表からメモリーへのデータの再ロードが必要となります。Net Search Extender が再始動されたりシステムがリブートされるたびにキャッシュ索引を最初から作成すると、永続キャッシュを再アクティブ化する場合に比べて時間がかかります。特に大規模な表では長時間かかります。一時キャッシュは、少量の固定データを使用している場合で、キャッシュ・データを作成するための時間を考慮する必要のない場合にのみ使用してください。

#### 永続的キャッシュ

これは、ディスク上で保守されていて、**DB2TEXT ACTIVATE CACHE** コマンドを使用することによって素早くメモリーにマップできます。増分索引更新のシナリオでは、表とキャッシュされた索引間の同期を許可するようにキャッ



シュをアクティブなままにする必要があります。そうしない場合、次の **DB2TEXT ACTIVATE CACHE** コマンドがキャッシュを最初から再作成します。

テキスト索引の更新は、次の方法で行えます。

#### **RECREATE INDEX ON UPDATE オプションを指定しない方法**

**RECREATE INDEX ON UPDATE** オプションを設定しない場合は、自動索引更新が行われます。この処理は、索引更新コマンドによってトリガーされます。更新間隔は、更新頻度オプションによって決定されます。更新処理は、増分更新とも呼ばれます。

削除された文書のスロットはキャッシュで再利用できないため、表への文書の削除および再挿入は実行しないでください。このため、アクティブにされた索引のキー列は変更しないでください。

#### **RECREATE INDEX ON UPDATE オプションを指定する方法**

これは、更新するたびに索引を再作成する方法です。可能な場合は、キャッシュ列の式で可変長データ・タイプを使用してください。これにより、キャッシュ・スペースが節約できます。また **CACHE TABLE** 節には、対応するキャスト式を使用してください。

データが非常に安定しているとはいえない場合、つまり最初の索引のアクティブ化以後に、全体の 50% よりも多い文書を挿入すると予想した場合は、このオプションを使用してください。

Net Search Extender には、**CREATE INDEX** メモリー・パラメーターを決定するのに役立つ 2 つの SQL 関数があります。それは、**MAXIMUM\_CACHE\_SIZE** 関数と **PCTFREE** 関数です。

- **MAXIMUM\_CACHE\_SIZE** は、キャッシュ索引の最大サイズを指定します。次の UDF 関数を使用すると **MAXIMUM\_CACHE\_SIZE** の値をメガバイト (MB) 単位で取得できます。

```
DB2EXT.MAXIMUM_CACHE_SIZE(maximumNumberDocs INTEGER,
 averageRowLength INTEGER, numberOfCacheColumns INTEGER)
```

以下のコマンドは、表の平均の行の長さのパラメーターを戻します。

```
SELECT AVG(LENGTH(cache column_1) + ... + LENGTH(cache column_n))
```

平均は、さらに文書を表に挿入するときに、大きく変わる場合があることに注意してください。キャッシュ列の数は、**DB2TEXT CREATE INDEX** コマンドの **CACHE TABLE** 節で使用した列式の数に関連しています。

詳細に関しては、63 ページの『第 15 章 ストアード・プロシージャ検索のメモリー所要量』を参照してください。

- **PCTFREE** は、**MAXIMUM\_CACHE\_SIZE** で指定したキャッシュの何パーセントを追加の文書のために空けておくのかを指定します。以下の UDF 関数は、実際の文書数、および最大文書数に基づく推奨 **PCTFREE** 値を戻します。

```
DB2EXT.PCTFREE(actualNumberDocs INTEGER, maximumNumberDocs INTEGER)
```

文書の実際数とは、メモリー・キャッシュを作成する最初の **ACTIVATE CACHE** コマンド実行時の、ユーザーの表の行数に相当します。

最大文書数とは、次の **DB2TEXT ACTIVATE** コマンド (一時キャッシュの場合)、または **DB2TEXT ACTIVATE CACHE RECREATE** コマンド (永続的キャッシュの場合) が実行される前の、ユーザーの表における文書の最大予想数のことです。

デフォルトは 50% に設定されます。更新のたびに索引を再作成する場合は、**PCTFREE** を 0 に設定してください。

## 例

表に 10 000 行あるとします。そして、20 000 行を超えることはないと予想しているものとします。次の呼び出しを使用して、必要な **PCTFREE** 値を計算します。

```
db2 "values DB2EXT.PCTFREE(10000,20000) "
```

最大行サイズが 20 000 で、キャッシュには 2 列あり、その平均サイズが 76 であるとして、次の呼び出しを使用して、サイズを戻します。

```
db2 " values DB2EXT.MAXIMUM_CACHE_SIZE(20000,76,2) "
```

## 次のタスク

適切なパラメーターを決定したら、次の呼び出しを使用して、キャッシュ索引を作成できます。

```
db2text CREATE INDEX db2ext.comment FOR TEXT ON db2ext.texttab (comment)
 CACHE TABLE (docid) PCTFREE 10 MAXIMUM CACHE SIZE 5
```

この例では、結果表を高速で戻すためにメイン・メモリーを使用して、`docid` 列がキャッシュされます。キャッシュ・メモリーの 10 % は、今後の文書のために予約され、キャッシュの最大は、5 MB に制限されています。



---

## 第 29 章 ビュー上でのテキスト索引の作成

ビュー上では、テキスト索引を作成して、ストアード・プロシージャ、または表値検索関数を使用できます。

### このタスクについて

しかし、スカラー関数 (CONTAINS など) はいずれも使用できません。もう 1 つの主要な制限として、ビュー上にはトリガーを作成できないため、基礎となる基本表の変更を自動認識できないという点があります。

したがって、増分索引更新の場合、ユーザーは、テキスト索引とデータベースを同期化するために、どの文書が追加、更新、または削除されたのか理解している必要があります。このためには、すべての変更をログ表に追加する必要があります。この処理を以下のサンプルで示します。

### 例

1. 次のコマンドを使用して、基本表を作成します。

```
db2 "create table DB2EXT.TLOGIX140789
(key INTEGER not null PRIMARY KEY,
name VARCHAR(50) not null, comment VARCHAR(90))"
```

2. 以下のコマンドを使用して、一部の項目を追加します。

```
db2 "insert into DB2EXT.TLOGIX140789 values
(1,'Claus','works in room 301')"
```

```
db2 "insert into DB2EXT.TLOGIX140789 values
(2,'Manja','is in the same office as Juergen')"
```

```
db2 "insert into DB2EXT.TLOGIX140789 values
(2,'Juergen','has the longest way to Raiko')"
```

```
db2 "insert into DB2EXT.TLOGIX140789 values
(3,'Raiko','is sitting in the office besides Claus ')"
```

3. 次のコマンドを使用して、ビューを作成します。

```
db2 "create view sampleview as select key, comment from DB2EXT.TLOGIX140789"
```

4. テキスト索引を作成、更新、およびアクティブ化するために、以下のコマンドを使用します。

```
db2text "create index indexview for text on sampleview(comment)
 cache table (comment) maximum cache size 1 key columns
 for index on view (key)"
```

```
db2text "update index indexview for text"
```

```
db2text "activate cache for index indexview for text"
```

**注:** ビューに基づくテキスト索引を作成するには、キャッシュ表を指定する必要があります。正しいログ表を作成するには、ビュー上の索引に関するキー列を指定しなければなりません。この方法で索引を作成すると、表値関数を使用して索引を検索することもできます。

パーティション・データベース環境でストアード・プロシージャ検索を使用する場合は、単一パーティション上の管理表の表スペースを明示的に指定し、明示的にこのパーティションで呼び出す必要があります。適切なパーティションに接続するためには、**DB2NODE** 環境変数を使用します。

5. 表を更新するために、以下のコマンドを使用します。

```
db2 "insert into DB2EXT.TLOGIX140789 values
(4,'Bernhard','is working on the same floor
 as Manja, but not as Claus')"
db2 "insert into DB2EXT.TLOGIX140789 values
(5,'Gunter','shares the office with Raiko')"
```

6. 次にログ表を更新します。ログ表の名前を取得するには、以下のコマンドを使用します。

```
db2 "select INDSHEMA,INDNAME,LOGVIEWSCHEMA,LOGVIEWNAME
 from db2ext.textindexes"
```

以下にログ表のレイアウトを示します。

| sqltype | sqllen    | sqlname.data | sqlname.length |
|---------|-----------|--------------|----------------|
| 496     | INTEGER   | 4 OPERATION  | 9              |
| 392     | TIMESTAMP | 26 TIME      | 4              |
| 497     | INTEGER   | 4 PK01       | 4              |

ログ表に項目を追加するには、以下のコマンドを使用します。

```
db2 "insert into DB2EXT.TLOGIX140789 values(0,CURRENT TIMESTAMP,4)"
db2 "insert into DB2EXT.TLOGIX140789 values(0,CURRENT TIMESTAMP,5)"
```

最初の値は、操作 (0= 挿入、1 = 更新、2 = 削除) を示します。2番目の値は、常に CURRENT TIMESTAMP である必要があります。最後の値は挿入、更新、または削除された行の主キーです。

7. 索引を再び更新するために、以下のコマンドを使用します。

```
db2text "update index indexview for text"
```

これで、ストアード・プロシージャを使用して新しい値を検索できるようになりました。

---

## 第 30 章 範囲パーティション表でのテキスト索引の作成

範囲パーティション表に対するテキスト索引の作成は、増分索引更新をサポートする拡張テキスト保持ステージング・インフラストラクチャーを使用しても、使用しなくても行うことができます。

### このタスクについて

範囲パーティション表のテキスト索引に対してインフラストラクチャーを使用不可にするには、次の例に示されているように、**AUXLOG** パラメーターを **OFF** に設定した **CREATE INDEX** コマンドを指定してください。

```
db2text create index sampleix for text on sample(comment) administration tables in
mytablespace index configuration(auxlog off) connect to mydb
```

この場合、1 次ログ表が追加され、トリガーによって文書の変更が認識されます。範囲パーティション表に索引を作成する場合は、**ADMINISTRATION TABLES IN** 節を使用する必要がある点に注意してください。これを使用しないと、エラーが発生します。

増分更新を使用して、範囲のアタッチまたはデタッチに関連した変更を処理すること、または **INSERT** パラメーターを指定した **LOAD** コマンドを使用して追加したパーティションにロードした文書を処理することはできません。基本表と同期化するために、テキスト索引を再作成する必要があります。

拡張テキスト保持ステージング・インフラストラクチャーがテキスト索引に対して使用可能になっている場合、文書の更新は更新トリガーによって 1 次ログ表に取り込まれます。一方、文書の挿入および削除は保水性処理によって補助ステージング表に取り込まれます。この処理を以下のサンプル・シナリオで示します。

### 例

**シナリオ 1:** 拡張テキスト保持ステージング・インフラストラクチャーを使用して表に対してパーティションをアタッチする

```
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)"
db2text "create index uc_007_idx for text on uc_007_customer_archive (address)
administration tables in mytablespace"
db2 "select indexname, logviewname, auxstagingname from db2ext.textindexes"
db2text "update index uc_007_idx for text"
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"
db2 "alter table uc_007_customer_archive attach
partition p2001 starting(2001) ending(2002)
exclusive from uc_007_customer_2001"
```

変更はまだ認識されておらず、健全性処理が必要です。

```
db2 "select * from sysibmts.systsauxlog_ix253720"
```

| PK                    | GLOBALTRANSID | GLOBALTRANSTIME | OPERATIONTYPE |
|-----------------------|---------------|-----------------|---------------|
| -----                 |               |                 |               |
| 0 record(s) selected. |               |                 |               |

```
db2 "set integrity for uc_007_customer_archive immediate checked"
```

健全性処理によって、従属表が保留モードになります。

```
db2 "select * from sysibmts.systsauxlog_ix253720"
```

| PK                                                                                                               | GLOBALTRANSID | GLOBALTRANSTIME | OPERATIONTYPE |
|------------------------------------------------------------------------------------------------------------------|---------------|-----------------|---------------|
| -----                                                                                                            |               |                 |               |
| SQL0668N Operation not allowed for reason code "1" on table<br>"SYSIBMTS"."SYSTSAUXLOG_IX253720". SQLSTATE=57016 |               |                 |               |

1 つ以上のテキスト保持ステージング表に対して健全性処理を実行します。コマンドは、表のすべてのテキスト索引を処理します。

```
db2text "reset pending for table uc_007_customer_archive for text"
```

```
db2 "select * from sysibmts.systsauxlog_ix253720"
```

| PK    | GLOBALTRANSID        | GLOBALTRANSTIME               | OPERATIONTYPE |
|-------|----------------------|-------------------------------|---------------|
| ----- |                      |                               |               |
| 1     | x'0000000000002215B' | x'20081020204612500381000000' | 1             |
| 2     | x'0000000000002215B' | x'20081020204612500602000000' | 1             |
| 3     | x'0000000000002215B' | x'20081020204612500734000000' | 1             |
| 5     | x'0000000000002215B' | x'20081020204612500864000000' | 1             |

増分更新は、新規にアタッチされたパーティションからのデータを処理します。

```
db2text "update index uc_007_idx for text"
```

**シナリオ 2:** 拡張テキスト保持ステージング・インフラストラクチャーを使用して表のパーティションをデタッチする

```
db2 alter table uc_007_customer_archive detach partition p2005 into t4p2005
SQL3601W
```

The statement caused one or more tables to automatically be placed in the Set Integrity Pending state. SQLSTATE=01586

```
db2text "reset pending for table uc_007_customer_archive for text"
```

```
db2text "update index uc_007_idx for text"
```

---

## 第 31 章 索引作成のためのパフォーマンスに関する考慮事項

索引作成時のパフォーマンスを向上させようとする場合は、以下の点を考慮する必要があります。

索引作成時のパフォーマンスを向上するために、次の点について考慮してください。

- `LONG VARCHAR` または `CLOB` の代わりに、`VARCHAR` データ・タイプを使用して、テキスト文書を保管します。
- テキスト索引とデータベース・ファイルの保管用に、別々の物理ディスクを使用します。
- `VARCHAR` タイプではなく、`TIMESTAMP` や `INTEGER` などの小さな主キー列を使用する。
- ユーザーのシステムに、このデータすべてに使用できる十分な実メモリーがあることを確認する。メモリーが不十分な場合、オペレーティング・システムは代わりにページング・スペースを使用します。これによって、索引付けと検索のパフォーマンスは低下します。
- 索引の自動更新時または手動更新時に更新の `commitcount` パラメーターを使用すると、増分索引作成時の索引作成パフォーマンスが低下します。このパラメーターは、初期更新プロセス時には使用されません。
- 索引更新中に多数のエラー・メッセージと警告メッセージがイベント・ログ表に書き込まれる場合は、パフォーマンスが低下することがあります。



---

## 第 10 部 テキスト索引の保守

テキスト索引を保守し、テキスト索引の状況に関する有益な情報を取得するには、多くの保守タスクを実施する必要があります。

このセクションでは、テキスト索引を保守する方法、およびテキスト索引の状況に関する有益な情報を取得する方法について説明します。保守タスクは以下のとおりです。

1. テキスト索引の更新と再編成
2. テキスト索引の変更
3. 索引更新イベント情報の消去 (削除)
4. テキスト索引のドロップ
5. 索引状況の表示

さらに、索引と使用可能データベースをバックアップおよびリストアする方法についても説明します。





## 第 32 章 テキスト索引の更新と再編成

テキスト索引を最初に作成し、更新した後は、テキスト索引を最新のものに維持する必要があります。例えば、テキスト文書を表に追加する際、または表内の既存の文書を変更する際は、索引の内容と表の内容の同期をとるために、文書に索引を付ける必要があります。同様に、表からテキスト文書を削除する場合は、文書の用語参照を索引から除去する必要があります。

**CREATE INDEX** コマンドで **RECREATE** オプションを指定すると、更新ごとに索引が完全に再作成されます。このオプションは、ログ表もトリガーも作成しません。索引を完全に再作成するには時間もリソースも大量に必要であるため、大きい表が存在する場合はこのオプションの使用に注意してください。

**RECREATE INDEX ON UPDATE** オプションなしでテキスト索引が作成された場合、新規文書、変更された、または削除された文書に関する情報がトリガーによってログ表に保管されます。テキスト索引が拡張テキスト保持ステージング・インフラストラクチャー (AUXLOG ON) によって構成されている場合、更新トリガーは変更された文書に関する情報をログ表に保管します。一方、挿入、削除に関する情報は保全性処理によって補助ステージング表に保管されます。

通常は、指定したインターバルで自動的に索引を更新します。既存の索引の更新頻度は、**ALTER INDEX** コマンドを使用して変更できます。

索引更新の頻度は、更新が作成されるタイミング、およびログ表にどれだけのテキスト変更が待機したら索引更新を開始するのかの最小数によって指定します。指定した日時に、ログ表内の変更の数が指定した数より少なければ、索引は更新されません。

定期的な索引作成は注意して計画してください。大量のテキスト文書について索引を作成するタスクは、多くの時間とリソースを必要とする場合があります。必要な時間は、多くの要因によって異なります。例えば、文書のサイズ、最後の索引更新以後に追加または変更したテキスト文書の数、およびプロセッサの能力などです。索引更新を計画するのに重要な 2 つのヒントがあります。

- パーティション・データベース環境で作業する場合、索引更新のスケジュールに特別な注意を払い、並行する索引更新の数を最小限にしてください。システムにおけるテキスト索引のスケジュール済み更新では、いくつかの索引更新が並行して実行されると、特定の時間に更新される索引の数と同じほど多くのプロセスがパーティションごとに生じることになる可能性があるためです。
- 多数の索引と非常に高頻度の自動更新とを組み合わせないでください。組み合わせると、デッドロック状態になる可能性があります。例えば、索引数が 100 個のときに、更新頻度に 5 分ごと、1 日 24 時間、1 週間で 7 日間を設定した場合は、管理する必要のあるチェックポイントとして 1 週間につき  $100 * 12 * 24 * 7 = 201600$  個分のリストが内部的に生成されます。

注: 次のような場合、DB2 表でロールバックおよびデッドロック状態が起り得ます。

- 更新頻度が高い場合

- 変更トランザクションが頻繁に起こる場合
- トランザクションが長い場合

データベース表が更新されると、Net Search Extender の索引に反映する必要のある変更がログ表に記録されます。これらのログ表項目は、項目の処理が終わるとログ表から削除されます。ログ表に対するこれらの削除操作が、記録する必要のあるデータベース表に対する更新と同時に実行された場合は、デッドロック状態になる可能性があります。

テキスト保持ステージング表インフラストラクチャーが索引に対して構成されている場合、基本表での特定のデータベース操作が補助ステージング表へのアクセスをブロックする可能性があります。テキスト索引を更新する前に、補助ステージング表が保留モードになっていないか確認してください。

---

## 第 33 章 テキスト索引の更新

**UPDATE INDEX** コマンドを使用すれば、索引を要求時に即時に更新できます。

### このタスクについて

**時期** 定期的な索引作成の実行を待たないで、索引を即時に更新する時。

### コマンド

```
UPDATE INDEX
```

**許可** ステートメントの許可 ID が保持する特権に、少なくとも次のいずれかが含まれている必要があります。

- 索引が定義されている表に対する **CONTROL** 特権
- **DATAACCESS** 権限

次のコマンドは、索引を更新します。

```
db2text UPDATE INDEX comment FOR TEXT
```

このコマンドは、複数のテキスト文書をデータベースに追加済みであり、これらの文書を直ちに検索する必要がある場合に便利です。

**CREATE INDEX** 時に **AUTOMATIC REORGANIZE** を指定した場合は、索引は必要なときに自動的に再編成されます。

上記ではなく **MANUAL REORGANIZATION** を指定したときに、手動の再編成が必要かどうかを判断するには、次のコマンドを使用して、db2ext.textindexes ビューを照会してください。

```
db2 "select reorg_suggested from db2ext.textindexes where INDNAME = 'comment'"
```

**MANUAL REORGANIZATION** を指定して列を頻繁に更新すると、更新処理は遅くなることに留意してください。手動で再編成するには、次のコマンドを使用してください。

```
db2text UPDATE INDEX comment FOR TEXT reorganize
```



---

## 第 34 章 テキスト索引の変更

更新頻度、または索引ディレクトリーと作業ディレクトリーを変更する必要がある時は、**ALTER INDEX** コマンドを発行します。

### 始める前に

ステートメントの許可 ID が保持する特権に、少なくとも次のいずれかが含まれている必要があります。

- 索引が定義されている表に対する **CONTROL** 特権
- **DBADM** 権限

### このタスクについて

**ALTER INDEX** コマンドを使用して、索引の作業ディレクトリー、索引の更新頻度、またはキャッシュ特性、特に **MAXIMUM CACHE SIZE** または **PCTFREE** を変更します。更新頻度を指定しない場合、現行設定値が未変更のまま残ります。索引の更新中、または検索が実行中の場合には、エラー・メッセージが表示されます。このエラー・メッセージは、索引が現在ロックされていて、変更できないことを示します。

### 例

以下の例は、索引の更新頻度を変更します。

```
db2text ALTER INDEX comment FOR TEXT
 UPDATE FREQUENCY d(1,2,3,4,5) h(12,15) m(00) UPDATE MINIMUM 100
```

この例では、月曜日から金曜日の 12:00 または 15:00 に、100 個以上のテキスト文書がキューに入れられている場合に、索引が更新されます。

索引の定期的な更新を停止するには、次のコマンドを入力します。

```
db2text ALTER INDEX comment FOR TEXT
 UPDATE FREQUENCY NONE
```

**ALTER INDEX** コマンドを使用して索引ディレクトリーを変更すると、索引ファイルは元の索引ディレクトリーから新しいロケーションに移動されます。この処理の間、索引はロックされます。大規模な索引の場合、およびファイル・システムをまたぐ変更の場合は、かなりの時間がかかることがあります。コピー処理が終了すると、索引がアンロックされて、再び使用できるようになります。



---

## 第 35 章 索引イベントのクリア

索引のイベント表内のメッセージが不要になった場合は、**CLEAR EVENTS FOR INDEX** コマンドを発行します。

### 始める前に

ステートメントの許可 ID には、以下の特権が少なくとも 1 つ含まれていなければなりません。

- 索引が定義されている表に対する CONTROL 特権
- DBADM 権限

### このタスクについて

更新の開始時刻と終了時刻、索引付き文書の数、または更新時に発生した文書エラーなどの索引作成イベントに関する情報が、索引のイベント表に保管されます。この情報によって、問題の原因を判別できます。これらのメッセージが不要になった場合には、削除できます。

### 例

以下の例は、指定したテキスト索引からメッセージを削除します。

```
db2text CLEAR EVENTS FOR INDEX comment FOR TEXT
```





---

## 第 36 章 テキスト索引の妥当性検査 (Windows、AIX)

DB2 バージョン 10 フィックスパック 1 以降では、Net Search Extender 索引妥当性検査ユーティリティ (**checknseindex** コマンド) を使用して、テキスト索引が正しく機能するようにしてください。

### 始める前に

妥当性検査の対象であるテキスト索引が、更新操作中、または削除操作中でないことを確認してください。

### このタスクについて

次のいずれかのシナリオが発生した場合、テキスト索引に破損の可能性がないか、チェックが必要です。

- 検索照会がエラーを返したが、具体的なエラー・コード、または説明がない。
- 検索結果には、新しく追加、または更新されたテキスト文書が反映されていない。
- 索引の更新に失敗。
- カーネル・エラー・メッセージがイベント・ログにある。

### 手順

テキスト索引の妥当性検査を実行するには、少なくとも **-i** と **-p** パラメーターを指定して **checknseindex** コマンドを実行してください。このユーティリティでは、大量のシステム・リソースを使用するため、テキスト索引のサイズによっては、実行に長時間かかるおそれがあります。

### タスクの結果

例 1: **checknseindex** コマンドを使用して、有効なテキスト索引の状況を検査します。

```
C:\SQLLIB\bin\checknseindex.exe -p "C:\litu_ict\Corrupted Index\NODE0000"
-i IX335811
```

```
CTE5265I The NSE index validation utility found the specified index,
"IX335811", to be valid.
```

これ以上のアクションは不要です。

例 2 **checknseindex** コマンドを使用して、間違ったパスが指定された有効なテキスト索引の状況を検査します。

```
C:\SQLLIB\bin\checknseindex.exe -p "C:\litu_ict\Corrupted Index\NODE00001"
-i IX335811
```

```
CTE5254E The NSE index validation utility failed to validate the specified
index because the utility could not access the
path "C:\litu_ict\Corrupted Index\NODE00001".
```

索引パスを修正し、コマンドを再発行してください。

例 3: **checknseindex** コマンドを **-deepCheck**、および **-v** パラメーターと共に使用して、無効なテキスト索引の状況を検査します。

```
C:\SQLLIB\bin\checknseindex.exe -p "C:\%litu_ict\Corrupted Index\%NODE0000"
-i IX335812 -deepCheck -v
```

```
=====
checknseindex

Net Search Extender index validation utility

=====

Validating NSE mapping indexes...

Forward mapping data control record : Number of blocks = "1"
Reverse mapping data control record : Number of blocks = "1"
Forward mapping index control record : Number of documents = "8"
Reverse mapping index control record : Number of documents = "8"

Validating NSE internal indexes...

Fri Mar 30 13:39:40 2012 Verify primary index...
100%
Fri Mar 30 13:39:40 2012 Verify secondary index...
100%
Fri Mar 30 13:39:40 2012 End

Validating consistency of NSE mapping and NSE internal indexes...

CTE5263E The NSE index validation utility found the NSE mapping index
and the NSE internal index to be inconsistent. Reason code: "1".
Diagnostic data: "".
```

壊れた索引の修正を試みます。

### 次のタスク

- 索引が壊れており、ハード・ディスクのスペースを数ギガバイト占有していない場合は、索引を再作成してください。詳しくは、テキスト索引の作成についてのトピックを参照してください。
- 無効な索引によってハード・ディスクのスペースを数ギガバイト占有している場合は、IBM お客様サポートにお問い合わせください。
- ドキュメント管理システムを使用している環境で Net Search Extender テキスト索引をバックアップする場合、対応する Net Search Extender テキスト索引のリカバリー手順に従ってください。

---

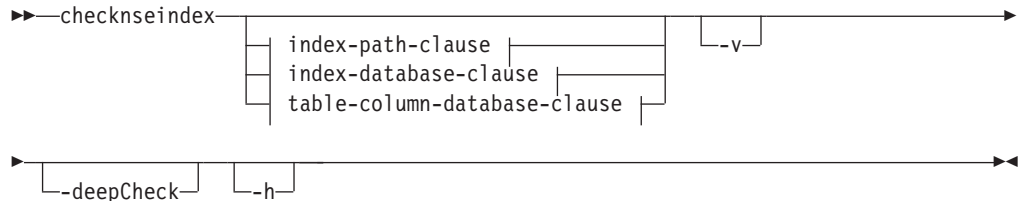
## checknseindex コマンド (Windows、AIX)

DB2 バージョン 10.1 フィックスパック 1 以降では、Net Search Extender テキスト索引の破損および他の異常を検出することができます。このコマンドによって、マッピング索引および内部索引の間で整合性をチェックして、データが正しくカタログされるようにすることができます。

## 許可

- 許可 ID には、テキスト索引ディレクトリーの読み取りアクセス権が最低限必要です。
- データベース・インスタンス・レベルで操作が可能なユーザーは、このユーティリティーに必要な実行権を持ちます。

## 構文



### index-path-clause:

```
--p absolute_index_path --i index_id
```

### index-database-clause:

```
--indschema index_schema_name --indname index_name --dbname database_name
```

### table-column-database-clause:

```
--tabschema table_schema_name --tablename table_name --colname column_name
--dbname database_name
```

## パラメーター

**-p** *absolute\_index\_path*

索引ディレクトリーの絶対パスを指定します。

**-i** *index\_id*

検査する索引 ID を指定します。

**-indschema** *index\_schema\_name*

NSE 索引の妥当性検査をするための索引スキーマを指定します。

**-indname** *index\_name*

NSE 索引の妥当性検査をするための索引名を指定します。

**-dbname** *database\_name*

NSE 索引の作成時に使用するデータベース名を指定します。

**-tabschema** *table\_schema\_name*

NSE 索引が属し、妥当性検査を必要とする表スキーマを指定します。

**-tablename** *table\_name*

NSE 索引が属し、妥当性検査を必要とする表名を指定します。

**-colname** *column\_name*

NSE 索引作成の対象となり、妥当性検査を必要とする表の列名を指定します。

**-v** 詳細ツール出力を表示します。

**-deepCheck**

Net Search Extender のマッピング索引および内部索引の両方をチェックして、同期が取れていることを検査します。**-deepCheck** パラメーターを指定せずにこのユーティリティを実行して、索引が有効であることをユーティリティがレポートしたにもかかわらず次の条件のいずれかに当てはまる場合、**-deepCheck** パラメーターを指定してユーティリティを再実行してください。

- 検索照会がエラーを返したが、具体的なエラー・コード、または説明がない。
- 検索結果には、新しく追加、または更新されたテキスト文書が反映されていない。
- 索引の更新に失敗。
- カーネル・エラー・メッセージがイベント・ログにある。

**-deepCheck** パラメーターを指定せずにこのユーティリティを実行して、索引が無効であることをユーティリティがレポートした場合、**-deepCheck** パラメーターを指定してユーティリティを再実行する必要はありません。

**-h** このユーティリティのヘルプを表示します。

このユーティリティでは、大量のシステム・リソースを使用するため、テキスト索引のサイズによっては、実行に長時間かかるおそれがあります。

## 例

例 1: **checknseindex** コマンドを実行して、有効なテキスト索引の状況をチェックします。

```
C:\%SQLLIB%\bin%\checknseindex.exe -i IX335811 -p "C:\myTextIndexes\%NODE0000"
CTE5265I The NSE index validation utility found the specified index, "IX335811",
to be valid.
```

## 使用上の注意

- Net Search Extender 索引妥当性検査ユーティリティは、SQLLIB/bin ディレクトリにあります。Windows オペレーティング・システム上でこのユーティリティを実行するには、**checknseindex.exe** を使用してください。AIX オペレーティング・システム上では、**checknseindex** を使用してください。

- 次のサンプル SQL ステートメントを変更し、ご使用のシステム上で索引 ID を取得してください。

```
select a.indexidentifier, indexdirectory
from db2ext.ttextcolumns a, db2ext.ttextindexes b
where a.indexidentifier = b.indexidentifier
and a.tablename = 'MYTAB'
and a.schemaname = 'MYSHEMA' and a.columnname = 'MYDATA''
```

- パーティション化された環境では、対応するすべてのノード上でこのユーティリティを実行してください。

---

## 第 37 章 テキスト索引のドロップ

テキスト列でテキスト検索を行う必要がなくなった場合は、**DROP INDEX FOR TEXT** コマンドを発行します。

### 始める前に

ステートメントの許可 ID には、以下の特権が少なくとも 1 つ含まれていなければなりません。

- 索引が定義されている表に対する CONTROL 特権
- DBADM 権限

### 例

```
db2text DROP INDEX comment FOR TEXT
```

テキスト索引をドロップすると、以下の表とビューもドロップされます。

- 索引のログ表とビュー
- 索引のイベント表とビュー
- ログ表のトリガー (存在する場合)
- テキスト保持ステージング表とビュー (存在する場合)

**注:** 表をドロップする前に、表の索引を必ずドロップしてください。先に表をドロップした場合、テキスト保持ステージング表 (存在する場合) も同じようにドロップされますが、ドロップした表の管理表およびビューの索引は残ってしまいます。



---

## 第 38 章 テキスト索引状況の表示

データベース内の現行テキスト索引に関する情報を得るには、Net Search Extender カタログ・ビューを使用します。

### 例

例えば、現行データベースのデフォルトを知りたい場合は、以下のコマンドを使用します。

```
db2 "select * from db2ext.dbdefaults"
```

現在使用できる索引、それに対応する表、および索引作成の行われた文書に関する情報を得るには、以下のコマンドを使用します。

```
db2 "select indschema, indname, tabschema, tablename, number_docs
from db2ext.textindexes"
```

特定の索引のフォーマットに関する情報を得るには、次のようにします。

```
db2 "select format, modelname from db2ext.textindexformats where
indschema = 'DB2EXT' and indname = 'TITLE'"
```

COMMITCOUNT を設定しない場合、db2ext.textindexes からの NUMBER\_DOCS パラメータは更新処理の実行中は更新されません。更新処理中に、現在の更新済み文書数を表示するには、次のコマンドを使用してください。

```
db2text CONTROL LIST ALL LOCKS FOR DATABASE sample INDEX db2ext.title
```





---

## 第 39 章 テキスト索引のバックアップとリストア

使用可能なデータベースおよびテキスト索引をバックアップまたはリストアする前に、Net Search Extender サービスを停止する必要があります。

### 手順

- 使用可能なデータベースと、Net Search Extender によって作成されたテキスト索引をバックアップするには、以下のようにします。
  1. Net Search Extender が作成した索引とその保管場所を探します。  
db2ext.textindexes ビューで select ステートメントを呼び出します。  
db2 "select indschema, indname, indexdirectory from db2ext.textindexes"
  2. 索引更新が実行中でないことを確認してから、Net Search Extender サービスを停止します。 次のコマンドを発行します。  
db2text stop
  3. データベースをバックアップした後に、索引ディレクトリーとサブディレクトリーをバックアップします。
  4. Net Search Extender サービスを再始動します。 次のコマンドを発行します。  
db2text start
- 使用可能なデータベースと、Net Search Extender によって作成されたテキスト索引をリストアするには、以下のようにします。
  1. Net Search Extender を停止します。 次のコマンドを発行します。  
db2text stop
  2. 索引ディレクトリーのバックアップ・コピーを、以前と同じパスにリストアします。
  3. Net Search Extender を再始動します。 次のコマンドを発行します。  
db2text start



---

## 第 40 章 /tmp ディレクトリーからのファイルの除去

Net Search Extender サービスの実行中は、/tmp ディレクトリーから削除してはならないファイルがあります。

Net Search Extender サービスの実行中は、/tmp 内のファイルを削除するスケジュール済みジョブによってこれらのファイルが誤って除去されないように注意してください。Net Search Extender サービスの実行中は、以下のファイルが /tmp ディレクトリーに存在する必要があります。これらのファイルを削除しないでください。

- セマフォ・ファイルおよび共用メモリー・ファイル:

```
instance_owner.TEXT.0000.LATCH
instance_owner.TEXT.0000
instance_owner.CACHE.0000
instance_owner.SCHEDULER.LATCH
instance_owner.DEMON.SEM
instance_owner.DEMON.MEM
```

注: パーティション・データベース環境の場合、

```
instance_owner.TEXT.0001.LATCH、instance_owner.TEXT.0001、
instance_owner.CACHE.0001 など、他にも各ノードに対応する類似した名前のファイルが存在する可能性があります。
```

- 索引の作成時に、一時キャッシュを使用している場合は、/tmp に次のようなファイルが存在する場合があります。

```
database_name.IX123456
database_name.IX123456.data0
```



---

## 第 11 部 テキストを検索する方法

Net Search Extender では、SQL スカラー検索関数、ストアド・プロシージャ検索関数、および SQL 表値関数によりテキストの検索ができます。

Net Search Extender には、テキストを検索するための以下の方式があります。

### SQL スカラー検索関数

テキスト検索副照会は SQL 照会に組み込むことができます。Net Search Extender には使用可能な SQL 関数の拡張機能として SQL スカラー検索関数が用意されています。テキスト検索副照会を SQL 照会に組み込むことによって、Net Search Extender の検索機能を DB2 の XQuery 処理と組み合わせることができます。XML 文書に対するテキスト検索照会は `db2-fn:sqlquery()` XQuery 入力関数で使用でき、これによって結果の XML 文書を XQuery で直接処理できるようになります。

### ストアド・プロシージャ検索関数

これを使用すると、定義済みのキャッシュ結果表を戻すことができます。

### SQL 表値関数

この検索は、ストアド・プロシージャ検索と同じような方法で使用できます。

SQL スカラー検索関数について、このセクションでは以下の項目を説明します。

- CONTAINS、NUMBEROFMATCHES、および SCORE 関数を使用する、テキストの検索。

構文の詳細については 265 ページの『第 60 章 SQL スカラー検索関数および SQL 表値関数』を参照してください。

- CONTAINS 関数の例を使用した検索インデックスの指定。

構文の詳細な説明については 257 ページの『第 59 章 検索インデックスの構文』を参照してください。

ストアド・プロシージャ検索関数について、このセクションでは以下の項目を説明します。

- ストアド・プロシージャ検索を使用してテキストを検索する。
- 検索インデックスの指定については、257 ページの『第 59 章 検索インデックスの構文』のパラメーターの説明を参照してください。

SQL 表値関数について、このセクションでは以下の項目を説明します。

- SQL 表値関数および HIGHLIGHT 関数を使用するテキストの検索。

構文の説明は 265 ページの『第 60 章 SQL スカラー検索関数および SQL 表値関数』を参照してください。

- 検索インデックスの指定については、257 ページの『第 59 章 検索インデックスの構文』のパラメーターの説明を参照してください。

この章では、考慮する必要があるかもしれない検索パフォーマンスについても説明します。

検索を行う前に、さまざまなデータ・タイプを含めて 91 ページの『第 9 部 開発: テキスト索引の作成と保守』で説明したすべての該当する索引作成のステップを実行してあることを確認してください。

---

## 第 41 章 SQL スカラー検索関数を使用するテキストの検索

CONTAINS 関数、NUMBEROFMATCHES 関数、および SCORE 関数を使用して SQL スカラー検索関数を使用するには、いくつかの方法があります。

このセクションでは、以下の方法で SQL スカラー検索関数を使用する方法を、例を挙げて説明します。

- 関数 CONTAINS を使用して照会を発行する。
- 関数 NUMBEROFMATCHES を使用してテキスト文書内で見つかった検索語の一致数を判断する。
- 関数 SCORE を使用して検出されたテキスト文書の適合性を取得する。

構文の説明は 265 ページの『第 60 章 SQL スカラー検索関数および SQL 表値関数』を参照してください。

---

### 照会の発行

CONTAINS 関数の照会を発行すると、表の中で検出された列のテキストが検索されます。

#### 例

この例は、CONTAINS 関数が、表 texttab の列 comment 内のテキストを検索する方法を示しています。この関数は、テキストが検索指数を満足する場合には 1 を返し、それ以外の場合には 0 を返します。

```
SELECT AUTHOR,TITLE
 FROM DB2EXT.TEXTTAB
 WHERE CONTAINS(COMMENT, 'book') = 1
```

この例では、列 COMMENT で用語 book を検索します。

"" を検索することはサポートされていません。検索語で 2 つの連続する引用符を使用すると、構文エラー・メッセージが発生します。また、検索ストリング内で改行文字が使用されていると、照会構文エラーが発生します。

#### 注:

テキスト検索のみで非常に大きい結果セットが返ることがわかっている場合は、限定的な検索基準を追加するのが有益です。例を挙げます。

```
SELECT AUTHOR,TITLE
 FROM db2ext.texttab
 WHERE CONTAINS(COMMENT, 'book') = 1 AND PRICE < 20
```

---

### 検索を実行し、検出された一致の数を戻す

NUMBEROFMATCHES 関数を使用して、検索を実行し、検出された一致の数を戻すことができます。

## このタスクについて

NUMBEROFMATCHES 関数を使用して、各テキスト文書で検索語が検出された数を判断します。

```
SELECT AUTHOR,TITLE,NUMBEROFMATCHES(COMMENT,'"book"')
FROM DB2EXT.TEXTTAB WHERE
NUMBEROFMATCHES(COMMENT, '"book"') > 0
```

NUMBEROFMATCHES は各行の整数値を返します。

---

## 検索を実行し、検出されたテキスト文書のスコアを戻す

SCORE は、同じ索引で検出された他の文書と比較して、この文書が検索語にどの程度一致しているかを示す正数を返します。この値は、文書内で検出された一致の数と文書のサイズの関連を基に計算されます。

### 例

以下の例では、SCORE 関数を使用して、検出された文書のスコアを入手しています。

```
WITH TEMPTABLE(docid,score)
AS (SELECT docid,
SCORE(COMMENT,'"book"')
FROM DB2EXT.TEXTTAB)
SELECT *
FROM TEMPTABLE
WHERE score > 0
ORDER BY score ASC
```

SCORE は、0 と 1 の間の DOUBLE 値を返します。

SCORE の戻す値は同じ索引を使用した他の SCORE 値と比較する場合にのみ意味があります。他の索引を使用して戻されたスコアとこの値を比較することはできません。

注: ビューに作成される索引については、CONTAINS、SCORE、および NUMBEROFMATCHES 検索関数は使用できません。

SCORE 値は、DB2 データベース環境によって異なります。

- 非パーティション・データベース環境では、すべての文書が 1 つの表の中にあります。SCORE 値は、単一の表、および表内の他のすべての文書との文書関係に基づいて計算されます。
- パーティション・データベース環境では、すべての文書が異なるパーティションにあります。索引付けの際には、各パーティション上でローカルに存在する文書のみが、テキスト索引の作成に使用されます。この場合、SCORE 値は、複数のパーティションのうちただ 1 つのパーティションにあるすべての文書との文書関係に基づきます。



---

## 第 42 章 SQL 検索指数の指定

CONTAINS、NUMBEROFMATCHES、および SCORE の各関数はすべて、検索指数を使用します。本資料のこのセクションでは、CONTAINS 関数を使用して、Net Search Extender 関数での検索指数のさまざまな例を示します。

構文の詳細な説明については 257 ページの『検索指数の構文』を参照してください。

---

### 任意の順序での用語検索

1 つの検索指数に複数の用語を指定できます。複数の検索語を結合する 1 つの方法では、コンマを使用して結合します。

#### 例

例えば、以下のように指定します。

```
SELECT AUTHOR,TITLE
 FROM DB2EXT.TEXTTAB
 WHERE CONTAINS(COMMENT,
 '("kid", "dinosaur")') = 1
```

この形式の検索指数では、いずれかの検索語を任意の順序で含んでいるテキストが検索されます。論理項としては、暗黙の OR 演算子で検索語を接続します。

---

### ブール演算子 AND と OR を使用する検索

検索語は、ブール演算子「&」(AND) および「|」(OR) を使用して、他の検索語と結合できます。

#### このタスクについて

検索語は、ブール演算子「&」(AND) および「|」(OR) を使用して、他の検索語と結合できます。

```
SELECT AUTHOR, TITLE
 FROM DB2EXT.TEXTTAB
 WHERE CONTAINS(COMMENT,
 '"author" | "pulitzer"') = 1
```

ブール演算子を使用して複数の検索語を結合することもできます。

```
SELECT AUTHOR, TITLE
 FROM DB2EXT.TEXTTAB
 WHERE CONTAINS(COMMENT,
 '"author" | "pulitzer" & "book"') = 1
```

複数のブール演算子を使用する場合は、左から右に評価されます。ただし、通常のブール論理同様、論理 AND 演算子 (&) は、論理 OR 演算子 (|) よりも強く結合します。以下の例ではこの評価が示されています。この例では、括弧は含まれていません。

```
"book" & "pulitzer"| "year" & "author"
```

Net Search Extender は、ブール演算子を次のように評価します。

```
("book" & "pulitzer") | ("year" & "author")
```

他の順序でブール演算子が評価されるようにするには、次のように括弧を使用する必要があります。

```
"book" & ("pulitzer" | "year") & "author"
```

次のように、コンマ区切り記号を使用してチェーニングした検索語でブール演算子を結合することもできます。

```
("author", "pulitzer") & "book"
```

この場合、コンマは次のようにブール OR 演算子として解釈されます。

```
("author" | "pulitzer") & "book"
```

---

## ブール演算子 NOT を使用する検索

ブール演算子 NOT を使用して、特定のテキスト文書を検索から除外できます。

### 例

例えば、以下のように指定します。

```
SELECT AUTHOR, TITLE
FROM DB2EXT.TEXTTAB
WHERE CONTAINS(COMMENT,
 '("author", "pulitzer") & NOT "book"') = 1
```

この例では、用語「book」を含むすべてのテキスト文書を、「author」または「pulitzer」の検索結果から除外しています。

---

## ファジー検索

「ファジー」検索は、検索語と類似したスペルのワードを検索します。

### 例

```
SELECT AUTHOR, TITLE
FROM DB2EXT.TEXTTAB
WHERE CONTAINS(COMMENT,
 'fuzzy form of 80 "pullitzer"') =1
```

この例では、検索によって、つづりを誤ったワード pulitzer の出現を検出しています。

突き合わせレベル (例では「80」) は、必要な正確度を指定します。文書内につづりの誤りがある可能性が高い場合は、ファジー検索を使用してください。光学式文字認識装置 (OCR) または音声入力で作成した文書の場合には、つづりの誤りがよく発生します。ファジーの程度を示すには 1 以上 100 以下の値を使用します。100 は、完全一致を示し、80 未満の各値では、次第に「ファジー」となります。

**注:** ファジー検索では妥当な正確度が得られない場合は、文字のマスキングを使用して、用語の一部を検索してください。

---

## 用語の一部の検索 (文字のマスキング)

「ワイルドカード」文字とも呼ばれるマスク文字を使用すると、より広範囲な検索を実行できます。これにより、検索によって検出されるテキスト文書の数が増加します。Net Search Extender は、パーセント (%) と下線 ( \_ ) という 2 つのマスク文字を使用します。

### このタスクについて

Net Search Extender では、DB2 の述部 LIKE と同じ方法でこれらのマスク文字を使用します。

- % は、不定数の任意の文字を表します。

検索語の真ん中のマスク文字として % を使用した例を次に示します。

```
SELECT AUTHOR, TITLE
 FROM DB2EXT.TEXTTAB
 WHERE CONTAINS(COMMENT, 'th%er') = 1
```

この検索語は、「thriller」、「throttle」、および「thread-splitter」というワードを含むテキスト文書を検索します。

- \_ は、検索語内の 1 文字を表します。

次の例も、「thriller」というワードを含んでいるテキスト文書を検出します。

```
SELECT AUTHOR, TITLE
 FROM DB2EXT.TEXTTAB
 WHERE CONTAINS(COMMENT, 'th_iller') = 1
```

1 つの句に複数のワイルドカード文字を使用できます (句の複数の語にワイルドカードを含められます)。ただし、ワイルドカード拡張の結果の用語は、複数ワード用語ではなく、単一用語のみになります。例えば、ワイルドカード表現「th%er」は、句「the caller」とは一致しません。

ワイルドカード文字は、結果リストのサイズを大幅に増加して、パフォーマンスを低下させ、望ましくない検索結果を戻す可能性があるため、慎重に使用してください。

ファジー検索またはシソーラス検索とワイルドカード文字検索を組み合わせることはできないことに注意してください。また、ワイルドカード文字は、暗黙的または明示的な EXPANSION LIMIT *number* 検索パラメーターの範囲内で拡張可能であることを確認してください。EXPANSION LIMIT *number* 検索パラメーターの詳細については、260 ページの『検索パラメーター』を参照してください。

---

## マスク文字を含んでいる用語の検索

「%」文字または「\_」文字を含んでいる用語を検索する場合、これらの文字の前にエスケープ文字と呼ばれる文字を置いてください。照会内のエスケープ文字は、ESCAPE キーワードを使用して指定する必要があります。

### 例

次の例では、「!」がエスケープ文字です。

```
SELECT AUTHOR, TITLE
 FROM DB2EXT.TEXTTAB
 WHERE CONTAINS(COMMENT,
 "'100!%' ESCAPE '!') = 1
```

---

## 固定順序での用語検索

「primary key」を検索する場合、用語を固定順序で検索できます。

### 例

「primary key」を検索する場合、2つの用語が隣接し、指定したシーケンスで存在する場合にのみ、これらの用語が検出されます。

```
SELECT AUTHOR, TITLE
 FROM DB2EXT.TEXTTAB
 WHERE CONTAINS(COMMENT, "primary key") = 1
```

---

## 同じ文または同じ段落内での用語検索

Net Search Extender には、同じ文または同じ段落内での用語検索をする限定機能があります。

### 例

検索語 web が語 disk と同じ文にあるテキスト文書を検出する検索指数の例を、次に示します。

```
SELECT AUTHOR, TITLE
 FROM DB2EXT.TEXTTAB
 WHERE CONTAINS(COMMENT,
 "'web' IN SAME SENTENCE AS 'disk'") = 1
```

一緒に存在する複数のワードを検索することもできます。次の例では、同じ段落に存在する2つの句が検索されます。

```
SELECT AUTHOR, TITLE
 FROM DB2EXT.TEXTTAB
 WHERE CONTAINS(COMMENT,
 "'linguistic analysis processing' IN SAME PARAGRAPH AS
 'search algorithms'") = 1
```

---

## 構造化文書のセクション内での用語検索

このトピックでは、構造化文書のサブセクション H2 内で、検索語「IBM」が現れるテキスト文書を検出する検索指数の例を示します。

### 例

構造化文書のサブセクション H2 内で、検索語「IBM」が現れるテキスト文書を検出する検索指数の例を、次に示します。

```
SELECT CATEGORY, DATE
 FROM DB2EXT.HTMLTAB
 WHERE CONTAINS(HTMLFILE,
 'SECTIONS ("H2") "IBM"') = 1
```

セクション名には大文字小文字の区別があることに注意してください。モデル・ファイルと照会でのセクション名が同じになるようにしてください。

---

## シソーラス検索

シソーラス検索は、Net Search Extender での強力な検索語拡張関数です。検索対象となる追加の用語が、ユーザー自身が作成したシソーラスから取られるので、これらの用語を直接制御できます。

### このタスクについて

例えば、「データベース」のシソーラス検索では、「リポジトリ」や「DB2」のような用語も、これらの用語が関連すると指定してあれば、検出できます。

頻繁に検索し、より効果的な検索結果を生成する特定の関連エリアでこのタイプの検索を使用します。

### 例

以下の例は、シソーラス拡張を使用するための構文を示しています。

この例は、「product」という用語を指定して、シソーラス「nseamplethes」で検出される、この用語のすべての関連用語を追加することによって拡張します。

```
SELECT CATEGORY, DATE
 FROM DB2EXT.HTMLTAB
 WHERE CONTAINS(HTMLFILE,
 'THESAURUS "nseamplethes"
 EXPAND RELATED
 TERM OF "product"') = 1
```

次の例は、「product」という検索語を指定します。すると、検索語のすべての同義語に拡張して検索が実行されます。

```
SELECT CATEGORY, DATE
 FROM DB2EXT.HTMLTAB
 WHERE CONTAINS(HTMLFILE,
 'THESAURUS "nseamplethes"
 EXPAND SYNONYM
 TERM OF "product"') = 1
```

---

## 数値属性検索

テキスト索引に保管されている数値属性は、検索できます。

### このタスクについて

テキスト索引に保管されている数値属性は、次の構文を使用して検索できます。

```
SELECT AUTHOR, TITLE
 FROM DB2EXT.TEXTTAB
 WHERE CONTAINS(COMMENT,
 'ATTRIBUTE "PRICE" between 9 and 20') = 1
```

---

## フリー・テキスト検索

「フリー・テキスト検索」は、検索語がフリー・フォーム・テキストとして表される検索です。検索対象のサブジェクトを、自然言語の句または文によって示します。

### このタスクについて

フリー・テキスト照会でのワードの順序は問題とはなりません。ただし、フリー・テキスト照会のうちの 1 つ以上の照会条件が、検索対象の文書に存在する必要があります。

フリー・テキスト引数の検索ストリングでは、文字またはワードのマスキングがサポートされていないことに注意してください。

### 例

例えば、以下のように指定します。

```
SELECT AUTHOR, TITLE, SCORE(COMMENT,
 'IS ABOUT EN_US "something related to dinosaur"')
FROM DB2EXT.TEXTTAB
WHERE CONTAINS(COMMENT,
 'IS ABOUT EN_US "something related to dinosaur") = 1
```

---

## 第 43 章 その他の検索構文の例

追加の検索構文の具体例については、`sqllib/samples/extenders/db2ext/` にある `search` スクリプトを実行してください。

### このタスクについて

このファイルには、サンプル表に対して実行する Net Search Extender の検索関数の例が入っています。

次のコマンドを入力します。

```
db2 -tvf search
```

XML データの照会方法を示すサンプルもあります。データベースに接続したら、`db2 -tvf xmlsearch` を発行することによってデータを検索できます。

表と索引が作成されていない場合は、次の操作のいずれかを実行します。

- UNIX オペレーティング・システムの場合: `instance_owner_home/sqllib/samples/extenders/db2ext` ディレクトリーで `nseample` を実行します。
- Windows オペレーティング・システムの場合: `sqllib%samples%extenders%db2ext` ディレクトリーで `nseample (.bat)` を実行します。
- XML 検索では、データベースにデータを設定して、索引の作成および更新を行うには `xmlsample (.bat)database` を呼び出してください。





---

## 第 44 章 ストアード・プロシージャ検索を使用するテキスト検索

テキスト検索結果のランク・サブセットおよびハイパフォーマンス照会のみを必要とする場合はストアード・プロシージャ検索インターフェースを使用します。

すべての検索結果が必要な場合、または大量の文書の索引付けが必要な場合は、ストアード・プロシージャを使用しないでください。主な理由は、ユーザー表の一部がメモリーにコピーされる結果、大量の実メモリーを使用可能にする必要があるためです。

ストアード・プロシージャでは、まず 0 から 20 の結果を要求し、次に 21 から 40 というように、カーソル・ナビゲーションと類似の方法をとることができます。このカーソル機能をキャッシュ (索引作成中に計算される) の使用と組み合わせると、検索は極めて高速になります。これは、特にユーザー表との結合が不要であるためです。

ストアード・プロシージャを使用する場合は、以下の点を考慮に入れていることを確認してください。

- **CREATE INDEX** の実行時に、`Cache-search-result` オプションが指定されている。
- 増分更新実施の可能性を含めて、現在および将来の共有メモリー所要量が、完全に考慮に入れられている。
- **db2text activate** コマンドを使用して、索引のキャッシュがアクティブ化されている。
- パーティション・データベース環境の場合、ストアード・プロシージャ検索を使用できるようにするには、表は単一パーティション上の表スペースを使用し、同じパーティションでプロシージャを呼び出す必要がある。そうしない場合、検索は許可されず、エラーが戻されます。

以下は、ストアード・プロシージャ検索の例です。

```
db2 "call db2ext.textSearch('%book%', 'DB2EXT', 'COMMENT', 0, 2, 1, 1, '?', '?")"
```

最初のパラメーターは、検索語です。検索語の構文は、SQL スカラー関数における構文と同じです。後続のパラメーターは、索引スキーマおよび索引名です。名前を引用符で囲まなかった場合は、英大文字に変換されます。さらに後続の 2 つの数値引数には、結果スライスの開始点およびスライス内の結果数が戻されます。次の 2 つの整数値は、スコアとヒット情報を要求するかどうかを指定します。終わりの 2 つの値は関数の戻り値です。

**注:** より大きい結果セットを要求する場合は、ユーザー表スペースが必要です。使用できるものがない場合は、表スペースを作成してください。以下の例は、UNIX プラットフォーム上で表スペースを作成しています。

```
db2 "create user temporary tablespace tempts managed by system
using ('/work/tempts.ts')"
```



---

## 第 45 章 SQL 表値関数を使用するテキスト検索

すべての検索結果を必要としているわけではなく、ストアド・プロシージャ検索で使った場合のようにキャッシュ索引を使用するために十分なメモリがない場合は、SQL 表値関数を使用してください。

使用可能な SQL 表値関数は 2 つありますが、いずれも `db2ext.textsearch` と呼ばれます。一方には、`db2ext.highlight` 関数と併用するための 2 つの追加パラメーターがあります。

SQL 表値関数には、結果の一部のみにアクセスする、ストアド・プロシージャと同じカーソル・インターフェースがあります。ただし、依然として、結果をユーザー表に結合する必要があります。これは、以下の例に示されています。

```
db2 "select docid , author, score from TABLE(db2ext.textsearch('%book%' ,
'DB2EXT','COMMENT',3,2,cast(NULL as integer))) as t, db2ext.texttab u
where u.docid = t.primkey"
```

SQL 表値関数から戻すことのできる値を以下に示します。

```
--> primKey <single primary key type>
the primary key

--> score DOUBLE
the score value of the found document

--> NbResults INTEGER
the total number of found results (same value for all rows)

--> numberOfMatches INTEGER
the number of hits in the document
```

注:

- 単一の主キー列のみが許可されています。
- パーティション・データベース環境の場合、SQL 表値関数を使用して検索できるようにするには、表は表スペースを単一パーティションで使用しなければならず、同じパーティションで関数を呼び出す必要があります。そうしない場合、検索は許可されず、エラーが戻されます。

---

### HIGHLIGHT 関数の使用

SQL 表値関数 `db2ext.highlight` を使用するには、追加の `numberOfHits` および `hitInformation` パラメーターを指定して `db2ext.textsearch` 関数を使用する必要があります。

#### このタスクについて

インプリメンテーションが変更される可能性があるトランスフォーメーション関数を使用して索引を作成した場合、強調表示関数は使用すべきではありません。索引付け時点の位置情報と検索時点の位置情報の不一致のために、戻される強調表示位置情報が正しくない可能性があります。

## 例

この例では db2ext.highlight 関数を呼び出し、db2ext.textsearch 関数によって検出されたヒットをどれも強調表示することなく文書全体を表示します。

```
select p.docid,
 db2ext.highlight(p.comment, t.hitinformation, 'WINDOW_NUMBER = 0')
 as highlight
from DB2EXT.TEXTTAB p,
 table (db2ext.textsearch('"bestseller" | "peacekeeping" | "soldiers"
 | "attention"', 'DB2EXT', 'COMMENT', 0, 20,
 cast(NULL as INTEGER), 10)) t
where p.docid = t.primkey and p.docid = 2
```

照会によって、次の結果が戻されます。

DOCID HIGHLIGHT

```
2 A New York Times bestseller about peacekeeping soldiers called
 "Keepers" who devise a shocking scheme to get the worlds
 attention after their tour of duty ends.
```

1 record(s) selected.

この例では db2ext.highlight 関数を呼び出し、db2ext.textsearch 関数によって検出されたヒットをすべて強調表示して、文書全体を表示します。

```
select p.docid,
 db2ext.highlight(p.comment, t.hitinformation, 'WINDOW_NUMBER = 0,
 TAGS = ("<bf>", "</bf>") ') as highlight
from DB2EXT.TEXTTAB p,
 table (db2ext.textsearch('"bestseller" | "peacekeeping" | "soldiers"
 | "attention"', 'DB2EXT', 'COMMENT', 0, 20,
 cast(NULL as INTEGER), 10)) t
where p.docid = t.primkey and p.docid = 2
```

検索引数は以下の結果を戻します。

DOCID HIGHLIGHT

```
2 A New York Times <bf>bestseller</bf> about <bf>peacekeeping</bf>
 <bf>soldiers</bf> called "Keepers" who devise a shocking scheme to
 get the worlds <bf>attention</bf> after their tour of duty ends.
```

1 record(s) selected.

この例では、db2ext.highlight 関数を呼び出して、最大で文書の 10 の部分 (ウィンドウ) を表示します。各ウィンドウ・サイズは 24 文字です。この場合、ヒットの両側のデータは約 12 バイトです。さらに、表関数 db2ext.textsearch によって検出されたヒットが強調表示されます。

```
select p.docid,
 db2ext.highlight(p.comment, t.hitinformation, 'WINDOW_NUMBER = 10,
 WINDOW_SIZE = 24, TAGS = ("<bf>", "</bf>") ') as highlight
from DB2EXT.TEXTTAB p,
 table (db2ext.textsearch('"bestseller" | "peacekeeping" | "soldiers"
 | "attention"', 'DB2EXT', 'COMMENT', 0, 20,
 cast(NULL as INTEGER), 10)) t
where p.docid = t.primkey and p.docid = 2
```

検索引数は以下の結果を戻します。

DOCID HIGHLIGHT

```
2 York Times <bf>bestseller</bf> about <bf>peacekeeping</bf> ...
```

```
<bf>peacekeeping</bf> <bf>soldiers</bf> called "Keepers" ... the
worlds <bf>attention</bf> after their
```

1 record(s) selected.

検出される最初のヒットは **bestseller** で、このヒットは最初のウィンドウを決定します。2 番目のヒットは **peacekeeping** で、最初のヒットから 8 バイトしか離れていないため、最初のウィンドウに完全に入れられます。3 番目のヒット **soldiers** は、最初のウィンドウの外側にあるので、新規ウィンドウを決定します。2 番目のヒット **peacekeeping** は **soldiers** ヒットの左側から 2 バイトしか離れていないので、これも 2 番目のウィンドウに入れられ、強調表示されます。4 番目のヒット **attention** は、2 番目のウィンドウの外側にあるので、新規ウィンドウを決定します。このウィンドウのサイズに以前のヒットまたは追加のヒットは入らないので、ヒットの周りのデータがウィンドウに入ります。

また、WINDOW\_SEPARATOR が指定されていないため、デフォルトのウィンドウ区切り文字「...」を使用して 3 つのヒット・ウィンドウが分離されます。

注: db2ext.highlight 関数を使用する際のハイパフォーマンスを確保するには、ユーザーは db2ext.textsearch 表値関数の検索結果を制限する必要があります。



---

## 第 46 章 複数列の検索

複数の列のテキスト索引を作成する必要がある場合、最も簡単なのは、SQL スカラー関数を使用し、それらの列の検索を結合する方法です。

### 例

これを以下の例に示します。

```
SELECT AUTHOR,TITLE
 FROM DB2EXT.TEXTTAB
 WHERE CONTAINS(COMMENT,
 '"book"')=1 and CONTAINS(AUTHOR,'"Mike"')=1
```

表値関数の場合は、パフォーマンス上の理由で戻される表の和集合を使用する必要がある場合があるためさらに困難です。表値関数の場合の別の方法は、ビューを使用し、ビューの列の中で表の列を組み合わせて、単一のテキスト索引をこのビュー列上に作成するというものです。これによって、2 つの別々のテキスト検索呼び出しを行う必要がなくなります。

テキスト列を組み合わせると、パフォーマンスが向上する可能性があります。とは言え、これは個々の検索上の要件に大きく依存します。





---

## 第 47 章 外部結合でのテキスト検索の使用

CONTAINS() 検索関数を使用できます。これは、外部結合照会の中で使用されるものです。

CONTAINS() 検索関数を使用する外部結合照会を使用する場合は、CONTAINS() 述部で、外部結合のタプル保護側にある表の列を参照していない限りは、理由コード「CTE0129 NULL 値をパラメーターとして渡すことができません」によって照会が失敗する可能性があります。

例えば、T1 は「T1 left outer join T2」のタプル保護側であり、T2 は「T1 right outer join T2」のタプル保護側です。



---

## 第 48 章 検索時のパフォーマンスに関する考慮事項

検索時のパフォーマンスを向上させるには、SQL 内で検索を行う際の課題、ストアド・プロシージャに関する課題、CONTAINS 関数を併用せずに NUMBEROFMATCHES 関数や SCORE 関数を使用するかどうかを考慮する必要があります。

検索時のパフォーマンスを向上するために、次の点について考慮してください。

- SQL 内で検索する場合には、次の点について考慮してください。
  - パフォーマンスの低下に気付いた場合には、explain ステートメントを使用して、DB2 オプティマイザーの処理プランをチェックしてください。
  - パラメトリック検索を使用すると、検索が高速になることがあります。特に、結果のサイズを削減するために他の検索述部を使用している場合が当てはまります。
  - すべての結果を必要とするのでない場合、結果の制限キーワードを使用してください。
- ストアド・プロシージャを使用して検索する場合は、次の点について考慮してください。
  - 指定したキャッシュ表式は、データベースからメモリーにコピーされるため、ワークステーションに、このデータに使用できる十分なメモリーがあることを確認してください。十分メモリーがない場合は、ページング・スペースが使用され、検索パフォーマンスを低下させます。
- NUMBEROFMATCHES または SCORE 関数を使用する場合、CONTAINS 関数を併用しないと、照会パフォーマンスが低下することがあります。また、処理の重複を防ぐため、CONTAINS 関数のストリングが、NUMBEROFMATCHES または SCORE 関数に使用されているストリングと必ず正確に一致するようにしてください。



---

## 第 49 章 ユーザー・シナリオ

SQL スカラー検索、ストアド・プロシージャ検索または SQL 表値関数検索を実行するときに Net Search Extender を使用できます。

この章では、以下の順を追った例を使用して、Net Search Extender について学習します。

### SQL スカラー検索関数の例

このコマンド行の例では、索引作成および使用可能な検索関数を示しています。

### ストアド・プロシージャの例

このコマンド行の例では、前述の例の索引コマンドを使用しています。ただし、この例では、キャッシュを追加することによって、ストアド・プロシージャ検索に使用可能な、異なる索引作成関数と検索関数を示しています。

### SQL 表値関数の例

SQL 表値関数の例は、ストアド・プロシージャ検索の例の変形です。

注: 例を使用する前に、インストール検査手順を使用して、Net Search Extender を正常にインストールしてください。

---

## SQL スカラー検索関数 の場合の簡単な例

DB2 Net Search Extender の例に示すステップを SQL スカラー検索関数に使用できます。

### 例

DB2 Net Search Extender の例では、以下のステップを使用します。

1. データベースを作成する
2. データベースをテキスト検索可能にする
3. 表を作成する
4. フルテキスト索引を作成する
5. サンプル・データをロードする
6. テキスト索引を同期させる
7. テキスト索引で検索する

既存データベースを使用して、オペレーティング・システムのコマンド行でサンプル・コマンドを発行することができます。以下の例では、データベース名を sample とします。

### データベースを作成する

以下のコマンドを使用して、DB2 のデータベースを作成できます。

```
db2 create database sample
```

## データベースをテキスト検索可能にする

オペレーティング・システムのコマンド行で、DB2 コマンドと同じように DB2 Net Search Extender コマンドを発行できます。例えば、Net Search Extender インスタンス・サービスを開始するには、以下コマンドを使用します。

```
db2text START
```

次に、DB2 Net Search Extender で使用するデータベースを準備します。

```
db2text ENABLE DATABASE FOR TEXT CONNECT TO sample
```

このステップは、それぞれのデータベースで 1 回だけ実行します。

## 表を作成する

```
db2 "CREATE TABLE books (isbn VARCHAR(18) not null PRIMARY KEY,
author VARCHAR(30), story CLOB(100k), year INTEGER)"
```

この DB2 コマンドによって、books という名前の表が作成されます。この表には、author、story、isbn 番号、および本が出版された year に関する列が含まれます。表には、主キーが必要です。

## フルテキスト索引を作成する

```
db2text "CREATE INDEX db2ext.myTextIndex FOR TEXT ON books (story)
CONNECT TO sample"
```

このコマンドで、列 story のフルテキスト索引が作成されます。テキスト索引の名前は db2ext.myTextIndex です。

## サンプル・データをロードする

```
db2 "INSERT INTO books VALUES ('0-13-086755-1','John', 'A man was
running down the street.',2001)"
db2 "INSERT INTO books VALUES ('0-13-086755-2','Mike', 'The cat hunts
some mice.', 2000)"
db2 "INSERT INTO books VALUES ('0-13-086755-3','Peter', 'Some men
were standing beside the table.',1999)"
```

これらのコマンドで、3 冊の本の isbn、author、story、および publishing year が表にロードされます。

## テキスト索引を同期させる

sample 表のデータでテキスト索引を更新するには、以下のコマンドを使用します。

```
db2text "UPDATE INDEX db2ext.myTextIndex FOR TEXT CONNECT TO sample"
```

## テキスト索引で検索する

テキスト索引を検索するには、以下の CONTAINS スカラー検索関数を使用します。

```
db2 "SELECT author, story FROM books WHERE CONTAINS
(story, '¥"cat¥') = 1 AND YEAR >= 2000"
```

**注:** 使用するオペレーティング・システム・シェルによっては、テキスト検索句を囲む二重引用符の前に、別のエスケープ文字が必要な場合があります。前述の例ではエスケープ文字として「¥」を使用しています。

この照会では、本の year の値が 2000 以上で、cat という語を含むすべての本が検索されます。照会によって、次の結果表が戻されます。

AUTHOR Mike  
STORY The cat hunts some mice.

他に、SCORE および NUMBEROFMATCHES 関数がサポートされています。  
SCORE は、見つかった文書を検索指数がどの程度正確に表現しているかを示す標識を戻します。NUMBEROFMATCHES は、それぞれの結果の文書内にある照会語のマッチ回数を戻します。

---

## キャッシュを使用したストアード・プロシージャ検索 の場合の簡単な例

DB2 Net Search Extender の例に示すステップをキャッシュの使用に利用できます。

### 例

DB2 Net Search Extender ストアード・プロシージャ検索の例では、以下のステップを使用します。

1. キャッシュ・オプションを指定してテキスト索引を作成する。
2. 索引を同期させ、キャッシュをアクティブ化する。
3. TEXTSEARCH ストアード・プロシージャで検索する。

注: ストアード・プロシージャの例は、前の例が完了して、データベースがまだ使用可能であることを前提としています。

#### キャッシュ・オプションを指定してテキスト索引を作成する

データベースが既に使用可能になっているため、以下のコマンドを使用してフルテキスト索引を作成します。

```
db2text "CREATE INDEX db2ext.mySTPTextIndex FOR TEXT ON books (story)
 CACHE TABLE (author, story) MAXIMUM CACHE SIZE 1
 CONNECT TO sample"
```

この例では、フルテキスト索引は、列 story に関するものであり、列 author および story を含んだキャッシュ表が指定されています。テキスト索引の名前は mySTPTextIndex です。

#### 索引を同期させ、キャッシュをアクティブ化する

表に挿入されたデータで索引を更新するには、以下のコマンドを使用します。

```
db2text "UPDATE INDEX db2ext.mySTPTextIndex FOR TEXT CONNECT TO sample"
```

キャッシュをアクティブ化するには、以下のコマンドを使用します。

```
db2text "ACTIVATE CACHE FOR INDEX db2ext.mySTPTextIndex FOR TEXT
 CONNECT TO sample"
```

これによって、列 author および列 story の内容がキャッシュにロードされます。

#### TEXTSEARCH ストアード・プロシージャで検索する

DB2 Net Search Extender ストアード・プロシージャは、特定のケースでのみ使用できます。

```
db2 "call db2ext.textSearch
 ('¥"cat¥"', 'DB2EXT', 'MYSTPTTEXTINDEX', 0, 2, 0, 0, '?', '?")"
```

この照会では、cat に関するすべての本が検索されますが、最初の 2 つの結果のみを戻します。本についての結果表の例を次に示します。

出力パラメーターの値

```

Parameter Name : SEARCHTERM COUNTS
Parameter Value : 1
Parameter Name : TOTALNUMBEROFRESULTS
Parameter Value : 1
```

```
AUTHOR STORY
Mike The cat hunts some mice.
```

リターン状況 = 0

検索構文のその他のサンプルについては、DB2 インスタンス・ディレクトリーにある `sql1lib/samples/extenders/db2ext/search` ファイルを参照してください。

---

## SQL 表値関数の場合の簡単な例

SQL 表値関数は、直前の例で作成されたテキスト索引に対して使用できます。

### このタスクについて

SQL 表値関数照会は、前に使用した `CONTAINS` 照会に対応しています。詳しくは、トピック 157 ページの『SQL スカラー検索関数 の場合の簡単な例』に含まれる『テキスト索引を同期させる』を参照してください。

```
db2 "SELECT author, story FROM books b, table (db2ext.textsearch
('¥"cat'¥"', 'DB2EXT', 'MYTEXTINDEX', 0, 2, CAST
(NULL AS VARCHAR(18)))) T where T.primKey = b.isbn
```

この例では、`NULL` が主キーのデータ・タイプにキャストされます。



---

## 第 50 章 検索項目を拡張するためのシソーラスの使用

特定の検索項目のみでなく、それに関連する項目も検索することによって、照会を拡張することができます。シソーラスからの関連検索項目の検索と抽出を行う Net Search Extender の関数を使用することによって、この処理を自動化できます。

シソーラスとは、通常特定のサブジェクト・エリアに関する意味的に関連する用語の統制された語いです。

Net Search Extender を使用すれば、以前作成したシソーラスから追加の用語を追加することによって、検索項目を拡張することができます。照会でシソーラス拡張を使用する方法に関しては、257 ページの『第 59 章 検索指数の構文』を参照してください。

シソーラスを作成して検索アプリケーションで使用するには、シソーラス定義ファイルが必要であり、これを内部フォーマット、シソーラス・ディクショナリーにコンパイルする必要があります。

このセクションでは、以下のことについて説明します。

- 『シソーラスの構造』

シソーラスは、関係によって互いにリンクされたノードのネットワークのような構造になっています。このセクションでは、Net Search Extender の定義された関係およびユーザー独自の関係を定義する方法について説明します。

- 163 ページの『シソーラスの作成およびコンパイル』

ここでは、シソーラス定義ファイルの構文、およびシソーラス定義ファイルをシソーラス・ディクショナリーにコンパイルするために使用するツールについて説明します。

---

### シソーラスの構造

シソーラスは、関係によって互いにリンクされたノードのネットワークのような構造になっています。

Net Search Extender は、シソーラスで特定の用語の検索から始め、次に、用語間の関係のパスに従い、そのプロセスで見つかった用語を出力します。

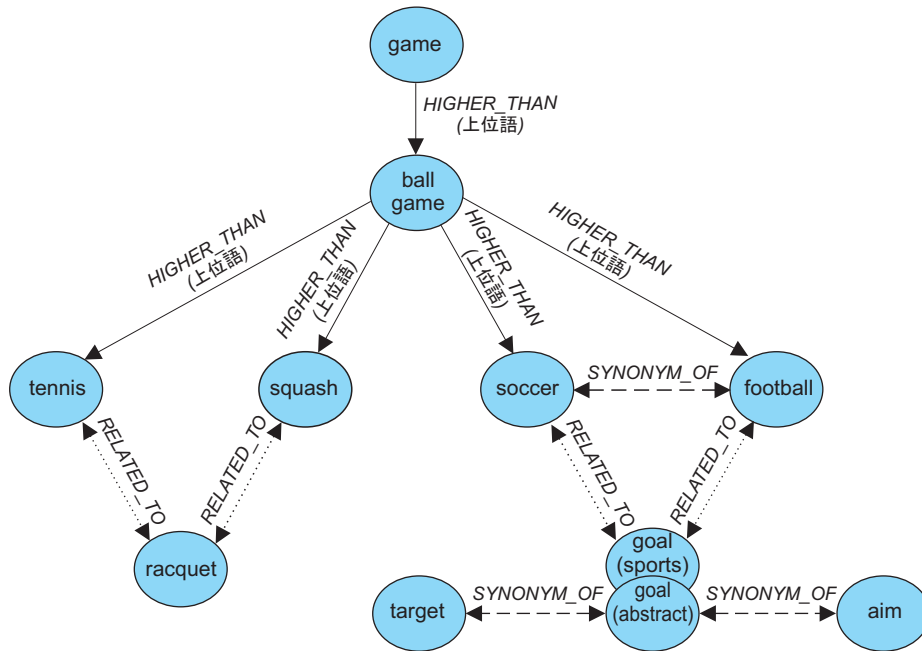


図7. シソーラスの構造の例

シソーラスの項目は関係によって結び付けられています。BROADER などの関係名を使用すると、関係階層での名前の付いた特定の線に拡張を制限できます。関係には双方向のものと単方向のものがあります。例えば、BROADER は単方向関係の名前です。

## 定義済みシソーラス関係

Net Search Extender は、連想関係、同義語関係、および階層関係などの定義済み関係を備えています。

Net Search Extender で事前に定義された関係を以下に示します。

- 連想関係

連想関係は、同じ概念を表現しているわけではないものの、相互に関係する 2 つの用語の間の双方向の関係です。

定義済み連想関係: RELATED\_TO

例:

```
tennis RELATED_TO racket
football RELATED_TO goal (sports)
```

- 同義語関係

同義語関係は、同じ、または類似した意味を持ち、相互に代替語として使用できる 2 つの用語の間の双方向の関係です。この関係は、例えば、用語とその省略形などに使用できます。

定義済み同義語関係: SYNONYM\_OF

例:

spot SYNONYM\_OF stain  
US SYNONYM\_OF United States

162 ページの図 7 の図では、同じシソーラスで、2 つの goal という用語が示されています。1 つはコメント (sports) が付いており、もう 1 つはコメント (abstract) が付いています。用語が同じスペルを持っていても、同義語関係で異なるワード・グループを接続できます。シソーラスを定義する際に、異なる関係を使用することによってこれをモデル化することができます。

#### • 階層関係

階層関係は 2 つの用語間の単一方向の関係です。ある用語は別の用語より広い (よりグローバルな) 意味をもっています。この関係を使用すると、方向により、より限定的な用語またはよりグローバルな用語を検索できます。

定義済み階層関係:

- LOWER\_THAN 幅が狭くなる関係をモデル化します。

LOWER\_THAN 関係は、より限定的用語のシーケンスをモデル化するためのものです。幅が狭くなる関係の先へいくほど、用語がより具体的なものになっていきます。例えば、LOWER\_THAN 関係を指定して ball game という用語を検索すると、squash tennis などが結果として戻り、より一層限定された具体的な用語のリストが示されます。

- HIGHER\_THAN 幅の広くなる関係をモデル化します。

HIGHER\_THAN 関係は、だんだんグローバルになっていく用語のシーケンスをモデル化するためのものです。このような関係の先へいくほど、用語がより具体的でなくなっていくます。例えば、HIGHER\_THAN 関係を指定して ball game という用語を検索すると、game などが結果として戻り、より一層広範囲な抽象的な用語のリストが示されます。

## ユーザー独自の関係の定義

Net Search Extender を使用すると、ユーザー独自の RELATED\_TO、LOWER\_THAN、および HIGHER\_THAN のシソーラス関係を定義できます。

各関係名はユニークなものでなければならぬため、RELATED\_TO(42) などのようにユニーク番号を追加することによって、そのような関係名を限定する必要があります。

LOWER\_THAN(42) などのように、同じリレーションシップ番号を使用して異なるタイプのリレーションシップを定義できます。番号 0 は、Net Search Extender の定義済み関係を示すために使用されます。

---

## シソーラスの作成およびコンパイル

Net Search Extender 関数で使用できるシソーラスを作成する際は、以下のステップに従ってください。

Net Search Extender 関数で使用できるシソーラスを作成するには、以下のステップを使用します。

1. シソーラス定義ファイルを作成します。

2. 定義ファイルをシソーラス・ディクショナリーにコンパイルします。

## シソーラス定義ファイルの作成

シソーラス定義ファイルを作成するには、一定の制約事項があります。

### このタスクについて

ユーザー独自のシソーラスを作成するには、まず、テキスト・エディターを使用して、定義ファイルで内容を定義します。

**制約事項** 拡張子を含むファイル名の長さは、256 文字を超えてはなりません。同じディレクトリーに複数のシソーラスをもつことができますが、各シソーラスごとに別々のディレクトリーをもつことをお勧めします。

サンプルの英語のシソーラス定義ファイル `nsesamplethes.def` が提供されています。Windows システムのシソーラス・ディレクトリーは次のとおりです。

```
sqllib\db2ext\thes
```

UNIX システムでの、シソーラス・ディレクトリーは次のとおりです。

```
instance_owner_home/sqllib/db2ext/thes
```

定義ファイルの最初のいくつかの定義グループを以下に示します。

```
:WORDS
 accounting
 .RELATED_TO account checking
 .RELATED_TO sale management
 .SYNONYM_OF account
 .SYNONYM_OF accountant

:WORDS
 acoustics
 .RELATED_TO signal processing

:WORDS
 aeronautical equipment
 .SYNONYM_OF turbocharger
 .SYNONYM_OF undercarriage

:WORDS
 advertising
 .RELATED_TO sale promotion
 .SYNONYM_OF advertisement
:
:
:
```

図8. サンプル・シソーラス定義ファイルの一部

各定義グループの構文については、165 ページの『シソーラス・サポート』を参照してください。

各メンバーは、単一行に書き込む必要があります。各関連用語は、関係名の前に置く必要があります。メンバー用語が互いに関係している場合、メンバー関係を指定します。

メンバー用語と関連用語の長さは 64 文字に制限されています。同一文字の 1 バイト文字と 2 バイト文字は、同じと見なされます。大文字と小文字は区別されません。用語には空白文字を含めることができ、1 バイト文字のピリオド "." またはコロン ":" を使用できます。

ユーザー定義の関係は、すべて連想タイプに基づいています。これらは、1 から 128 の間のユニークな番号によって識別されます。

## 定義ファイルのシソーラス・ディクショナリーへのコンパイル

シソーラス定義ファイルをコンパイルするには、`db2extth` を実行します。

### このタスクについて

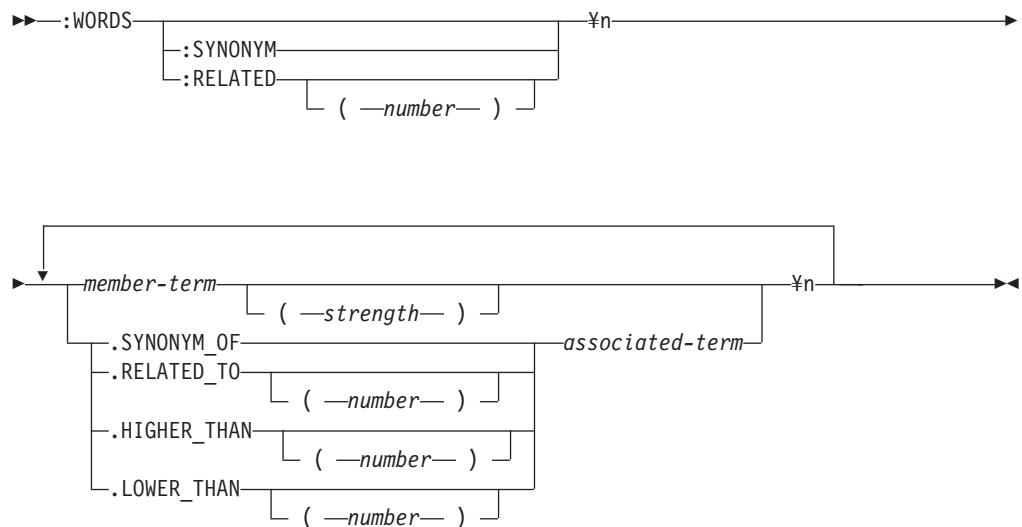
パーティション化された環境でシソーラス・ディクショナリーを使用するには、作成されたファイルにすべての物理ノードがアクセスできることを確認してください。

## シソーラス・サポート

独自のシソーラスを作成するときは、固有の構文を使用する必要があります。

ユーザーが独自のシソーラスを作成する場合の各定義グループの構文を以下に示します。

### シソーラス定義の構文



¥n は構文の一部ではありませんが、シソーラス定義ファイルでの行の終了を示します。

シソーラス定義ファイルでは、以下のようにしてコメント行を挿入できます。

```
my comment text
```

### :WORDS

関連するワード・グループを開始するキーワード。

## **:SYNONYM、:RELATED [(number)]**

関係名。

関係名は関係タイプおよび番号から構成されます。番号を省略すると、システム提供の関係名であるゼロが想定されます。:SYNONYM は常にシステム提供の関係名です。

:SYNONYM などのコロンで始まる関係名は、同じ関係で互いに関連しているワード・リストの前に置きます。例えば、以下のように指定します。

```
:WORDS
:SYNONYM
 air steward
 cabin staff member
 flight attendant
```

### *member-term*

シソーラス・ディクショナリーに組み込む用語。

- 最大長は 64 バイトです (コード・ページ UTF-8 の場合は 42 バイト)。
- 同一文字の 1 バイト文字と 2 バイト文字は、同じと見なされます。
- 英大文字および小文字は区別されません。
- 用語にはブランク文字を含めることができます。
- 1 バイト文字のピリオド "." またはコロン ":" は使用できません。

このパラメーターは、検索した用語と弱い関係をもつワードをシソーラス検索に含ませたくない場合に役に立ちます。強さは、1 から 100 までの数値です。デフォルト値は 100 です。

## **.SYNONYM\_OF、.RELATED\_TO [(number)], .HIGHER\_THAN [(number)], .LOWER\_THAN [(number)]**

関係名。関係名 .HIGHER\_THAN は BROADER 照会関係に対応します。また、関係名 .LOWER\_THAN は NARROWER 照会関係に対応します。関係名は関係タイプおよび番号から構成されます。番号を省略すると、システム提供の関係名であるゼロが想定されます。関係名 .SYNONYM は常にシステム提供の関係名です。

.SYNONYM\_OF などのピリオドで始まる関係名は、あるワードと別のワードとの関係を定義します。例えば、以下のように指定します。

```
:WORDS
 air steward
.SYNONYM_OF cabin staff member
.SYNONYM_OF flight attendant
```

オプションの *number* はユーザー定義の関係を識別します。この番号は、シソーラス定義ファイル全体から取られた重複しない数 (現在は 1 から 128) でなければなりません。例: RELATED\_TO(42)。

アプリケーションで関係名および番号の代わりにシソーラス関係のシンボル名を使用する場合は、アプリケーションで名前と番号のマッピングを処理する必要があります。例: 関係 *opposite\_of* を RELATED\_TO(1) として定義する場合は、アプリケーションでこの名前を内部関係名 RELATED\_TO(1) にマップする必要があります。

### *associated-term*

各関連用語は、関係名の前に置く必要があります。関連用語は、指定した関係について各メンバー用語と関連しています。すべてのメンバー用語がお互いに関連している場合は、メンバー関係を使用してそれを指定できます。

- 最大長は 64 バイトです (コード・ページ UTF-8 の場合は 42 バイト)。
- 同一文字の 1 バイト文字と 2 バイト文字は、同じと見なされます。
- 英大文字および小文字は区別されません。
- 用語にはブランク文字を含めることができます。
- 1 バイト文字のピリオド "." またはコロン ":" は使用できません。

以下に関連用語の例を示します。

```
:WORDS:SYNONYM
 reject
 decline
 RELATED_T0(1) accept
```

---

## シソーラスでサポートされている CCSID

複数の CCSID がシソーラスでサポートされています。

以下の CCSID がシソーラスでサポートされています。

|             |                           |
|-------------|---------------------------|
| <b>819</b>  | Latin 1                   |
| <b>850</b>  | PC データ Latin 1            |
| <b>874</b>  | タイ語                       |
| <b>932</b>  | 結合された日本語                  |
| <b>943</b>  | 結合された日本語                  |
| <b>949</b>  | 結合された韓国語                  |
| <b>950</b>  | 結合された中国語 (繁体字)            |
| <b>954</b>  | 日本語                       |
| <b>970</b>  | 結合された韓国語                  |
| <b>1208</b> | UTF 8                     |
| <b>1250</b> | Latin 2                   |
| <b>1252</b> | Latin 1                   |
| <b>1253</b> | チェコ語                      |
| <b>1254</b> | トルコ語                      |
| <b>1255</b> | ヘブライ語                     |
| <b>1256</b> | アラビア語                     |
| <b>1258</b> | ベトナム語                     |
| <b>1363</b> | 結合された韓国語                  |
| <b>1381</b> | 結合された中国語 (簡体字)            |
| <b>1383</b> | 中国語 (簡体字)、結合された SBCS/DBCS |
| <b>1386</b> | 中国語 (簡体字)、結合された SBCS/DBCS |
| <b>5039</b> | 日本語 (結合された SBCS/DBCS)     |

---

## シソーラス・ツールが戻すメッセージ

シソーラス・ツールを使用しているときに戻される可能性のあるエラー・メッセージがいくつかあります。

### ADM\_MSG\_INVALID\_CCSID

無効な CCSID が指定されました。

要求されたコード・ページはサポートされていません。

### ITL\_THES\_MSG\_BUFFER\_OVERFLOW

バッファオーバーフロー。

### ITL\_THES\_MSG\_DICT\_EXIST

シソーラス・ディクショナリー *dictionary name* が既に存在しています。

上書きできません。

### ITL\_THES\_MSG\_DICT\_INTEGRITY\_ERROR

ディクショナリー *dictionary name* の整合性が失われています。

シソーラス・ディクショナリー・ファイルが壊れています。

### ITL\_THES\_MSG\_DICT\_NOT\_EXIST

シソーラス・ディクショナリー *dictionary name* が存在しません。

### ITL\_THES\_MSG\_DICT\_VERSION\_ERROR

ディクショナリー *dictionary name* バージョン・エラー。

シソーラス・ディクショナリーが互換性のない古いバージョンによって作成されています。

### ITL\_THES\_MSG\_ERROR\_IN\_FILE

ファイル *file name* のエラー。

### ITL\_THES\_MSG\_FILE\_ACCESS\_ERROR

ファイル *file name* にアクセスできませんでした。

### ITL\_THES\_MSG\_FILE\_CLOSE\_ERROR

ファイル *file name* をクローズできませんでした。

### ITL\_THES\_MSG\_FILE\_EOF\_ERROR

*file name* に予期しないファイル終わりがあります。

定義ファイルにエラーがあります。

### ITL\_THES\_MSG\_FILE\_OPEN\_ERROR

ファイル *file name* をオープンできませんでした。

### ITL\_THES\_MSG\_FILE\_REACHED\_END

シソーラス定義ファイルに予期しないファイル終わりがあります。

定義ファイルにエラーがあります。

### ITL\_THES\_MSG\_FILE\_READ\_ERROR

ファイル *file name* を読み取れませんでした。



**ITL\_THES\_MSG\_FILE\_REMOVE\_ERROR**

ファイル *file name* を除去できませんでした。

**ITL\_THES\_MSG\_FILE\_RENAME\_ERROR**

ファイル *file name 1* の名前を *file name 2* に変更できませんでした。

**ITL\_THES\_MSG\_FILE\_WRITE\_ERROR**

ファイル *file name* に書き込むことができませんでした。

**ITL\_THES\_MSG\_IE\_BLOCK\_START**

ファイル *file name* の行 *line number* で、ブロック開始行が見つかりませんでした。

**ITL\_THES\_MSG\_IE\_EMPTY**

シソーラス定義ファイル *file name* が空です。

**ITL\_THES\_MSG\_IE\_NO\_TERM**

*file name* の行 *line number* で用語が定義されていません。

**ITL\_THES\_MSG\_IE\_REL\_SYNTAX**

*file name* の行 *line number* でリレーションシップが誤って指定されています。

**ITL\_THES\_MSG\_IE\_STRENGTH\_DOMAIN**

強さが範囲外です。

有効値は 1 から 100 です; デフォルトは 100 です。

**ITL\_THES\_MSG\_IE\_STRENGTH\_SYNTAX**

強さの値が誤って指定されています。

構文: 用語の後に、強さ 20 の場合は [ :20 ] と入力してください。

**ITL\_THES\_MSG\_IE\_TERM\_LEN**

シソーラス用語が 64 文字より長くなっています。

**ITL\_THES\_MSG\_IE\_USER\_DEF**

*file name* の行 *line number* でリレーションシップが誤って指定されています。

**ITL\_THES\_MSG\_IE\_USER\_DEF\_DOMAIN**

*file name* の行 *line number* でリレーションシップ番号が範囲外になっています。

**ITL\_THES\_MSG\_INPUT\_ERROR**

シソーラス定義ファイル *file name* の行 *line number* にエラーがあります。

**ITL\_THES\_MSG\_INTERNAL\_ERROR**

内部エラー

**ITL\_THES\_MSG\_LOCKED**

シソーラス・ディクショナリー *dictionary name* は使用中です。

**ITL\_THES\_MSG\_LOCKING\_ERROR**

ディクショナリー *file name* をロックできませんでした。

**ITL\_THES\_MSG\_MEMORY\_ERROR**

メモリー・エラー。

**ITL\_THES\_MSG\_NAMELEN\_ERROR**

パラメーター・エラー *file name*。シソーラス定義ファイル名が長過ぎます。

**ITL\_THES\_MSG\_NO\_TARGET\_DIR\_ERROR**

パラメーター・エラー。ターゲット・ディレクトリーが指定されていません。

**ITL\_THES\_MSG\_NONAME\_ERROR**

パラメーター・エラー。シソーラス定義ファイル名が指定されていません。

**ITL\_THES\_MSG\_NORMALIZE\_ERROR**

用語を正規化する際のエラー。

シソーラス定義ファイルにエラーがあります。

**ITL\_THES\_MSG\_OUTFILE\_EXIST**

出力ファイル *file name* は既に存在しています。

**ITL\_THES\_MSG\_PARAMETER\_ERROR**

内部パラメーター・エラー。

**ITL\_THES\_MSG\_PATHLEN\_ERROR**

パラメーター・エラー *file name*。シソーラス定義ファイル・パスが長過ぎます。パスの長さは、オペレーティング・システムのディレクトリー名でサポートされている最大長を超えてはなりません。

**ITL\_THES\_MSG\_UNEXPECTED\_ERROR**

予期しない内部エラー。

---

## 第 51 章 Net Search Extender 索引付け構成

Net Search Extender の索引付けや検索の動作を変更するためのいくつかの構成オプションがあります。

この章では、Net Search Extender の索引付けや検索の動作を変更するためのいくつかの構成オプションについて説明します。

- トークン化
- ストップワード
- 構成

---

### トークン化

索引作成時に、Net Search Extender は次の方法でテキストをトークンに分割して、文書テキストを処理します。

#### ワード

すべての英数字 ("a".."z","A".."Z", "0".."9") が、フルテキスト索引を作成するために使用されます。区切り文字は、ブランク文字および後述の文認識に関するセクションで説明する文字です。改行およびブランク文字などの制御文字は、次のように解釈されます。行の中間にある制御文字 (0x20 より小) は、ブランク文字と見なされます。改行 (0x0A) の前後にあるブランク文字と制御文字は無視されます。1 バイト文字の前後の改行はブランク文字と見なされ、同じ文字の 2 バイト文字は常にその 1 バイト文字と同じものと見なされます。同じ文字の大文字と小文字、例えば、"A" と "a" は、検索時に特に指定がない場合には同じ文字と見なされ、検索時に正確なマッチングが要求された場合には異なる文字と見なされます。

#### 文

Net Search Extender は、以下の条件が満たされる場合に「.」、「!」、「?」を認識します。

- 特殊文字「.」が文の終わりと見なされるためには、その後にブランク・スペースまたは改行が続く必要があります。
- テキストの任意の場所に文字「!」または「?」がある場合は (スペースまたは改行がない場合でも)、文の終わりを意味します。
- 「!」、「?」、または「.」が引用符で囲まれている場合は、無視されます。文の終わりであるとは見なされません。

#### 段落

段落の認識は、文書フォーマットによって異なります。プレーン・テキスト・フォーマットでは、2 つの連続する改行文字 (復帰文字が介在する可能性があります) は、段落の境界として認識されます。HTML では、段落タグ <p> が段落の境界として解釈されます。他の文書フォーマットでは、段落の認識をサポートしません。

---

## ストップワード

ストップワードとは、頻繁に現れ、テキスト検索処理に関連する内容をもたないワードのことです。

通常、"and"、"or"、および "in" など、すべての (言語的意味での) 機能語がストップワードと見なされます。ストップワードを索引検索した場合、テキスト検索システムの精度が大幅に低下する可能性があります。

Net Search Extender には、言語リストに対するストップワード処理が用意されています。索引の作成時に構成パラメーター **IndexStopWords** を設定でき、ストップワードに索引を付けるかどうかを指定できます。デフォルトは、ストップワードに索引を付けることを意味する 1 です。

ストップワードに索引を付けない場合は、索引の作成時に、**IndexStopWords** に 0 を設定するとともに、言語パラメーターを使用して入力文書の言語を指定する必要があります。ストップワードに索引を付けなかった場合は、索引が小さくなり、高速になります。索引を作成した後で構成 .ini ファイル・テンプレート内のこの値を変更しないでください。変更すると、文書の処理方法が索引付けされたタイミングによって異なってしまい、その結果、ストップワード処理の一貫性がなくなります。

索引作成時のストップワードの無視は、コレクション内のすべての文書が同じ言語で記述されている場合にのみ効果があります。

## ストップワードをサポートする言語

ストップワード処理を行うことができる言語がいくつかあります。

以下の言語でストップワード処理を行うことができます。

### **AR\_AA**

アラビア語諸国で話されるアラビア語

### **CA\_ES**

スペインで話されるカタロニア語

### **DA\_DK**

デンマークで話されるデンマーク語

### **DE\_CH**

スイスで話されるドイツ語

### **DE\_DE**

ドイツで話されるドイツ語

### **EL\_GR**

ギリシャで話されるギリシャ語

### **EN\_GB**

英国で話される英語

### **EN\_US**

米国で話される英語

### **ES\_ES** スペインで話されるスペイン語

|              |                        |
|--------------|------------------------|
| <b>FI_FI</b> | フィンランドで話されるフィンランド語     |
| <b>FR_CA</b> | カナダで話されるフランス語          |
| <b>FR_FR</b> | フランスで話されるフランス語         |
| <b>HE_IL</b> | イスラエルで話されるヘブライ語        |
| <b>IS_IS</b> | アイスランドで話されるアイスランド語     |
| <b>IT_IT</b> | イタリアで話されるイタリア語         |
| <b>IW_IL</b> | イスラエルで話されるヘブライ語        |
| <b>NB_NO</b> | ノルウェーで話されるノルウェー語ブークモール |
| <b>NL_BE</b> | ベルギーで話されるオランダ語         |
| <b>NN_NO</b> | ノルウェーで話されるノルウェー語ニーノシュク |
| <b>PT_BR</b> | ブラジルで話されるポルトガル語        |
| <b>PT_PT</b> | ポルトガルで話されるポルトガル語       |
| <b>RU_RU</b> | ロシアで話されるロシア語           |
| <b>SV_SE</b> | スウェーデンで話されるスウェーデン語     |

---

## 構成

Net Search Extender は、さまざまな組み合わせで使用される文字 (例えば英数字、数字、および特殊文字) を含むワードを検索できます。

そのために、Net Search Extender には以下の構成が用意されています。

### 文字正規化

文字正規化は、2 とおりの書き方ができるワードについて、どちらも検索できるようにします。例えば、ドイツ語のワード 'Überbau' は 'Ueberbau' と書くこともできます。正規化により、これら双方のワードを 'Überbau' と 'Ueberbau' のどちらからでも検索できます。この機能では、'accès' などのアクセント付き文字を 'acces' のような単純文字に一致させるための正規化も行います。このオプションを使用すると、例えば文字 Ü が Ue に相当する標準正規化を持たないような言語においては望ましくない結果が生じることに注意してください。

### ワードの一部としての特殊文字の使用

ワードの一部として特殊文字を使用すると、英数字、特殊文字、および数字の連続が含まれることのある製品名を 1 つのワードとして検索対象にできます。例えば、英数字の組み合わせ 'DT9' を 1 つのワードとして扱った

り、特殊文字' を使用可能にしたりします。これによって、OS/390<sup>®</sup> が 'OS' と '390' としてではなく 1 つのワードとして検索されます。

これらの構成設定では、スイッチを使用できます。スイッチをカスタマイズするには、索引を作成する前に .ini ファイル・テンプレートを変更します。

.ini ファイル・テンプレートは sqllib/db2ext/cteixcfg.ini に保管されています。CREATE INDEX コマンドを使用してこのテンプレート・ファイル内の値の大部分を変更することも可能ですが、以下の値のみを変更することをお勧めします。

AccentRemoval (for character normalization)  
UmlautNormalization (for character normalization)  
TreatNumberAsWords (for treating numeric characters as part of the word)  
AdditionalAlphanumCharacters (for using specific characters as part of a word)

#### **AccentRemoval**

このパラメーターは、アクセント付き文字を一致 (対応) する単純文字に正規化するかどうかを指定します。例えば、événement は evenement としても索引付けされます。デフォルトは True です。

#### **UmlautNormalization**

このパラメーターは、ウムラウト文字を、同じ意味の 2 文字としても索引付けするかどうかを指定します。例えば、'Übersee' は 'Uebersee' としても索引付けされます。デフォルトは True です。

#### **TreatNumbersAsWords**

このパラメーターは、ワードに隣接する数字をワードの一部とするかどうかを指定します。例えば、'DT9' は 'DT' という 1 のワードと数字 '9' ではなく 1 つのワードとして扱われます。

#### **AdditionalAlphanumCharacters**

このパラメーターのストリング値で、どの文字をワードの一部として扱うかを定義します。この特殊文字のストリングは、UTF-8 で記述した 1 つ以上の文字のシーケンスである必要があります。デフォルトのストリングには、"-@\_" の文字が含まれています。

ワードの一部として処理される文字のリスト内では、ワイルドカード文字である % および \_ は使用できません。使用した場合、照会実行時に問題が発生します。

これらの構成値のいずれかを変更する場合は、.ini ファイルを編集してから索引を作成してください。非アクティブなスイッチをアクティブにするには、行頭のコメント・マーカ「;」を削除します。詳細については cteixcfg.ini ファイルを参照してください。

.ini ファイル内の他の値は変更しないことをお勧めします。

---

## 第 12 部 構造化文書の使用

Net Search Extender を使用すると、構造化文書内の表題、作成者、価格などのテキスト・フィールドや数値フィールドに索引付けをしたり、検索したりできるようになります。

文書は XML、Outside-In、または HTML フォーマットで記述したり、ユーザー定義のタグ (GPP) を含めたりすることができます。

文書モデル のマークアップ・タグおよびフィールド名を使用して、文書内のどのフィールドに索引が付けられるか定義します。その結果、検索が使用可能になります。フィールドの名前 (セクション名とも呼ばれる) は、そのフィールドに対する照会で使用できます。

これらのフィールドを検索できるようにするためには、文書を含むテキスト索引の作成時に FORMAT および MODEL ファイルを指定する必要があります。





---

## 第 52 章 ネイティブな状態で格納された XML 文書の検索

XML データ列に索引を作成する場合、通常は、FORMAT を指定する必要はありません。XML タイプの列にテキスト索引を作成する場合、Net Search Extender はデフォルトで XML フォーマットを選択します。

以下のセクションでは、ネイティブな状態で格納された XML 文書の検索について説明します。ネイティブな状態で格納された XML 文書に対してセクション検索の概念を適用するための方法およびこの機能を XQuery 処理と統合する方法を説明します。

XML データ列に索引を作成する場合、通常は、FORMAT を指定する必要はありません。XML タイプの列にテキスト索引を作成する場合、Net Search Extender はデフォルトで XML フォーマットを選択します。XML データ列にフォーマット指定子 TEXT および HTML を使用することはできません。

XML 列におけるテキスト索引の作成および使用を示す以降のサンプルでは、次の XML 文書が使用されます。この XML 文書は、表 t1 にある XML タイプの列 c2 に格納されています。

```
<?xml version="1.0">
<purchaseOrder orderDate="2001-01-20">
 <shipAddress countryCode="US">
 <name>Alice Smith</name>
 <street>123 Maple Street</street>
 <city>Mill Hill</city>
 <zip>90999</zip>
 </shipAddress>
 <item partNo="123" quantity="1">
 <name>S&B Lawnmower Type ABC-x</name>
 <price>239.90</price>
 <shipDate>2001-01-25</shipdate>
 </item>
 <item partNo="987" quantity="1">
 <name>Multifunction Rake ZYX</name>
 <price>69.90</price>
 <shipDate>2001-01-24</shipdate>
 </item>
</purchaseOrder>
```

---

### デフォルト文書モデルの使用

CREATE INDEX ステートメントに文書モデルを指定しなかった場合、Net Search Extender は、デフォルトの文書モデルを使用します。

#### このタスクについて

デフォルト文書モデルの特性の 1 つは、セクション名が、各要素および属性の絶対パスを指定する XPath 表記で記述されていることです。検索照会内のセクション名は、照会の実行時に評価される XPath 式ではないことに注意してください。セクション名は、構造化文書内の特定の部分 (要素および属性) を指す名前です。

モデル・ファイルを使用していない場合は、次のようにして XML 文書のテキスト索引を定義してください。

```
db2text CREATE INDEX i1 FOR TEXT ON t1(c2) CONNECT TO mydbname
```

列 c2 はデータ・タイプが XML であるため FORMAT 指定を省略できます。この場合 FORMAT 指定にはデフォルトで XML が設定されます。

文書モデルを指定しなかった場合、各 XML 要素には、文書内の各要素の絶対 XPath に応じた名前が自動で割り当てられます。例えば、要素 price には、検索照会のセクション名 /purchaseOrder/item/price によってアクセスできます。属性 countryCode にはセクション名 /purchaseOrder/shipAddress/@countryCode を使用してアクセスできます。

**db2text update** コマンドを使用して索引を更新した後の、SECTION 検索を使用したスカラー検索関数による SQL 式の例を次に示します。

```
SELECT c2 FROM t1
WHERE CONTAINS(c2, SECTIONS("/purchaseOrder/item/name") "Rake") = 1
```

この照会は前述のサンプル XML 文書を戻します。

---

## カスタマイズした文書モデルの使用

カスタマイズしたセクション名を定義するためには、文書の特定の部分にユーザー定義の名前を割り当てる、モデル・ファイルを指定する必要があります。文書モデルを使用する利点の 1 つは、XML 文書のどの部分に索引付けするのかを指定でき、XPath 式でそれらの部分を指定できることです。

### このタスクについて

前述の XML 文書のモデル・ファイルの例を次に示します。

```
<?xml version="1.0"?>
<XMLModel>
 <XMLFieldDefinition
 name="itemName"
 locator="/purchaseOrder/item/name" />
 <XMLFieldDefinition
 name="customerName"
 locator="//shipAddress/name" />
 <XMLAttributeDefinition
 name="partNumber"
 type="NUMBER"
 locator="/purchaseOrder//item/partNo" />
 <XMLFieldDefinition
 name="none"
 locator="/purchaseOrder/orderDate"
 exclude="yes" />
</XMLModel>
```

この文書モデルでは、前述の検索照会で参照されている要素 /purchaseOrder/item/name に名前 itemName を割り当てています。

このモデル・ファイルを使用した索引定義を次に示します。

```
CREATE INDEX i1 FOR TEXT ON t1(c2) DOCUMENTMODEL XMLModel IN
/mydir/myfilename/xmlmodel.xml CONNECT TO mydbname
```

文書モデル名 (パラメーター DOCUMENTMODEL を使用する) は、モデル・ファイル内のルート要素を指定します。これは、XML 文書モデルにおける XMLModel です。パス /mydir/ ... は、モデルを定義しているファイルを指しています。

文書モデルの構文は、要素を容易に識別できる W3C XPath 構文のサブセットをサポートしています。

前述のモデル・ファイルを使用してテキスト索引を作成し、**db2text update** コマンドを使用して索引を更新し終わったら、次の方法で要素 /purchaseOrder/item/name を検索できます。

```
SELECT c2 FROM t1
WHERE CONTAINS(c2, SECTIONS("itemName") "Rake") = 1
```

文書モデルを指定しなかった場合の検索照会との違いに注意してください。どちらの照会も前述の同じサンプル XML 文書を戻します。

この XML 文書モデルでは、要素 item の XML 属性 partNo に、属性 partNumber も定義しています。Net Search Extender 属性定義のデータ・タイプは、NUMBER である必要があります。

前述のサンプル・モデル・ファイル内の属性定義を使用すると次のような値範囲の検索を実行できます。

```
SELECT c2 FROM t1 WHERE CONTAINS
(c2, ATTRIBUTE "partNumber" BETWEEN 300 AND 500) = 1
```

---

## XQuery サポート

データベース内の XML 文書の検索時には、XQuery を使用して検索結果を処理することもできます。DB2 データベース・サーバーのハイブリッド・データベース・エンジンを利用すると、SQL テキスト検索照会を XQuery 処理と組み合わせることができます。

### このタスクについて

これは、db2-fn:sqlquery() 入力関数を XQuery のコンテキストで使用することによって実現されます。XQuery 入力関数を使用するには、**set language XQuery** コマンドを使用して SQL から XQuery に切り替えるか、またはキーワード XQuery を照会の前に置く必要があります。これは、XQuery 式を扱うパーサーに対する重要な標識であり、XQuery 言語に適用される大/小文字の区別の規則と構文規則に従う必要があります。

db2-fn:sqlquery() 関数は、全選択を表すストリング・リテラルを受け取ります。db2-fn:sqlquery() 関数は、全選択によって選択された XML 列値の連結を表す XML シーケンスを戻します。

次の式を使用すると、ネイティブな状態で格納された XML 文書に対するテキスト検索と XQuery 処理を組み合わせることができます。

```
XQUERY db2-fn:sqlquery('SELECT c2 FROM t1
WHERE CONTAINS(c2,
''SECTIONS ("/purchaseOrder/item/name") "Rake" '')
= 1 ')//shipAddress/name
```

この照会は、XML 文書内の shipAddress 要素の下にあり、「Rake」という名前の購買注文品目を含むすべての name 要素を戻します。SELECT ステートメントで XML 列 (この例では c2) を明示的に選択する必要があります。

前述のサンプルは、次のように FLWOR 構成によって拡張して、ご使用のアプリケーションに組み込むことができます。

```
XQUERY FOR $item in db2-fn:sqlquery('SELECT c2 FROM t1
 WHERE CONTAINS(c2, '' SECTIONS ("/purchaseOrder/item/name") "Rake" '')
 = 1 ')
 WHERE $item[@partNo > "800"]
RETURN $item/price
```

db2-fn:sqlquery() 入力関数での全選択は、常に、ヒットした XML 文書全体を戻すことに注意してください。

ネイティブな状態でデータベースに格納されている次の XML 文書について考えます。

```
<?xml version="1.0" ?>
<dept bldg="101">
 <employee id="901">
 <name>Sabine</name>
 <resume>DB2 programmer</resume>
 </employee>
 <employee id="902">
 <name>Holger</name>
 <resume>XML expert</resume>
 </employee>
</dept>
```

同じ部門 (dept) に所属していて「XML」という用語が履歴書 (resume) に含まれる従業員 (employee) に対する検索は、次のようになります。

```
SELECT c2 FROM t1 WHERE CONTAINS(c2, SECTIONS("/dept/employee/resume") "XML")=1
```

この SELECT ステートメントは XML 文書全体を戻します。次のように XQuery に検索照会を埋め込みます。

```
XQUERY db2-fn:sqlquery('SELECT c2 FROM t1
 WHERE CONTAINS(c2,
 ''SECTIONS ("/dept/employee/resume") "XML" '') =1') //employee/name
```

この場合、以下の 2 つの結果が戻ります。

```
<name>Sabine</name>
<name>Holger</name>
```

従業員 Sabine の履歴書には「XML」という用語が存在しませんが、この XQuery の結果シーケンスに名前が表示されていることがわかります。この現象は、全選択では文書全体が戻される (つまり 1 人以上の従業員の履歴書に用語「XML」が存在する XML 文書全体が戻される) ために発生します。

照会で結果 <name>Holger</name> のみを戻すようにするには、次の XQuery ステートメントを発行します。

```
XQUERY for $d in db2-fn:sqlquery('SELECT c2 FROM t1
 WHERE CONTAINS(c2,
 ''SECTIONS ("/dept/employee/resume") "XML" '') =1')
 return $d/dept/employee/name[contains(parent::employee/resume,"XML")];
```

Net Search Extender は、XML 列上の構造依存のフルテキスト索引を使用することによって、セクション /dept/employee/resume に用語 XML を持つすべての XML 文書をフィルタリングして除外します。戻された XML 文書のサブセットに基づいて return ステートメント `return $d/dept/employee/name[contains(parent::employee/resume,"XML")]` は、XPath 軸を使用して XML 文書をナビゲートすることによって <resume> という名前の兄弟要素に XML という用語を含んだ <name> 要素のみを戻します。



---

## 第 53 章 構造化文書のサポート

---

### 文書モデルが構造化文書を記述する方法

HTML や XML 形式の文書は構造化文書の例です。これらの文書には、テキスト・フィールドや文書属性を識別するタグが含まれています。テキスト・フィールドには、文書の表題、作成者、または文書の説明などの情報を含めることができます。

以下に示すのは、構造化されたプレーン・テキスト文書の一部です。HTML のようなタグで区切られたテキストがあります。

```
[head]Handling structured documents
[/head]
```

```
[abstract]This document describes the concept of structured documents
and the use of document models to...
[/abstract]
```

```
:
:
```

構造化文書の索引を作成する場合、テキスト・フィールドおよび属性に索引を付けることができ、固有名を指定して保管できるように、Net Search Extender は構造を認識している必要があります。こうすることによって、Net Search Extender は SECTION 節または ATTRIBUTE 節を使用して、特定のテキスト・フィールドを選択的に検索したり、特定の属性を持つ文書を検索したりできるようになります。

Net Search Extender が特定の文書フォーマットを理解できるようにするには、文書モデル内の構造の定義を Net Search Extender に渡す必要があります。または、Net Search Extender に用意されているデフォルトの文書モデルを使用できます。

**CREATE INDEX** コマンドを呼び出して文書に索引を付けるときに、文書のモデルの名前を引数として指定します。例えば、次のようにします。CREATE INDEX i1 FOR TEXT ON t1(c2) DOCUMENT MODEL GPPModel IN mymodel.x1d CONNECT TO db

パラメーター **GPPModel** は、ユーザーが使用している文書モデルのタイプを示します。

文書モデルを使用して文書に索引付けする前に、まず文書モデルを定義してから、索引に文書モデルが認識されるようにする必要があります。

**注:** XML 文書が整形形式でない索引を使用する場合、索引付け処理は文書内で問題が検出された場所で停止します。つまり、文書の一部のみが索引付けされます。文書を訂正しない場合、文書の中で索引付けが行われた部分でしか検索を行えません。これは、表の列タイプが XML でない場合に限り発生します。

---

### 文書モデルの例

文書モデルはテキスト・フィールド定義と属性定義からなっています。

索引を付ける文書フォーマットごとに 1 つの文書モデルを定義することが必要です。以下は、プレーン・テキスト構造化文書の簡単な文書モデルです。この例の中の GPP は General Purpose Parser (汎用パーサー) の略です。

```
<?xml version="1.0"?>
<GPPModel> - the GPP document model begin here

 <GPPFieldDefinition - a field definition begins here
 name="Head" - the name you assign to this field
 start="[head]" - the boundary string at the beginning of the field
 end="[/head]" - the boundary string at the end of the field
 exclude="YES" />

 <GPPFieldDefinition - the next field definition begins here
 name="Abstract"
 start="[abstract]"
 end="[/abstract]"
 exclude="NO" />
 :
 :
</GPPModel>
```

文書モデルは、199 ページの『第 54 章 文書モデルの参照情報』に定義されているタグを使用して XML 言語で指定します。文書モデルはテキスト・フィールド定義と属性定義からなっています。前述の例に示されているのは、GPPFieldDefinition 要素に定義されているテキスト・フィールド定義のみです。同じようにして、GPPAttributeDefinition を使用して文書属性を定義できます。

例において 1 行目の <?xml version="1.0"?> は、文書モデルが XML タグを使用して書かれていることを示しています。テキスト・フィールド定義ごとに、フィールド定義の開始 (start) と終了 (end) を表す境界ストリングをソース・ドキュメントに指定します。したがって、ある文書に [head]、何かのテキスト、[/head] という順序で続く文字シーケンスが含まれる場合は、それらの境界ストリングの間のテキストが head という名前で識別されたテキスト・フィールドの内容であると解釈されます。

各フィールド定義にはフィールド名を割り当てます。このフィールド名を、CONTAINS 関数の SECTION 節で使用することにより、照会はテキスト・フィールドの内容の検索を制限できます。フィールド名は、固定の名前を指定することも、またはある規則によって構造単位の内容から導出することもできます。例えば、XML エンティティのタグ名や XML 属性の名前などをそのようにして決めることができます。

---

## 文書モデル

文書モデルは、主として文書の構造のどの部分に索引を付ける必要があるか、どのように索引を付けるかを制御します。

その目的は次のとおりです。

- ソース・ドキュメントで区別する必要があるテキスト・フィールドを識別する
- 上記のようなテキスト・フィールドのタイプを判別する
- フィールド名をテキスト・フィールドに割り当てる



文書モデルがテキストをテキスト・フィールドに属していると識別すると、テキストは文書のテキストの内容の一部であると見なされ、用語が抽出されて索引に保管されます。

文書モデルの要素は、その文書フォーマットに使用されたパーサーによって異なります。

- HTML フォーマットの場合、文書モデルは、HTML タグ名を使用して、どのタグに索引を付けるべきか、メタタグ情報をどのように処理するかを定義します。
- XML フォーマットの場合、定義済みの一連のタグは存在しないので、文書モデルはまず、どのタグが目的のタグであるかを定義する必要があります。同じ名前前の XML 要素も、他のどの要素に組み込まれているかに基づいて区別できます。
- GPP (汎用パーサー) フォーマットの場合、文書モデルはテキスト・フィールドの境界を判別する必要があるために、より深くパーサーと相互作用します。ここではフィールド定義が、フィールドの境界を検出するためにストリングを指定する必要があります。
- Outside-In フォーマットの場合、文書モデルは、HTML タグ名に類似したタグを使用して、どのタグに索引を付けるべきか、メタタグ情報をどのように処理するかを定義します。Outside-In トランスフォーメーション・テクノロジーは INSO としても知られています。

## テキスト・フィールド

文書モデルを使用することにより、文書の特定の部分またはセクションを、特定のテキスト・フィールドに属するもの、または文書属性、あるいはその両方として識別できます。

文書のテキストは、テキスト・フィールドの一部であるかどうかに関係なく、完全に索引付けされます。意味のある用語が抽出されて索引に保管されます。したがって、テキスト検索に制約がない場合、そのテキストは検索の対象となります。

ただし、テキスト・フィールドを定義すれば、特定のフィールドのテキストを選択的に検索できます。例えば、`structure` というワードをテキスト・フィールド `Abstract` に持つ文書を検索できます。例えば、次のようにします。

```
SELECT doc
from my_docs WHERE CONTAINS (doc, SECTIONS(Abstract) "structure" = 1
```

1 つの文書に同じテキスト・フィールドが複数回出現してもかまいません。例えば、すべての図のキャプションに適用されるテキスト・フィールドを定義することができます。テキスト・フィールドは、他のテキスト・フィールドにオーバーラップすることもできます。

特定のテキスト・フィールドの内容に索引付けをしないようにする場合は、`exclude="YES"` を含むフィールド定義を指定できます。テキスト・フィールドおよび文書属性の制限のリストが、202 ページの『テキスト・フィールドおよび文書属性の制限』に示されています。

## 文書属性

文書属性には、`number` というタイプの短い定様式の情報が組み込まれています。

テキスト・フィールドとは対照的に、このような属性を含む文書の検索には、値の範囲を使用できます。

属性は索引テキストとともに保管されず、別個の項目索引に保管されます。そのため、属性の内容で文書を検索するには、属性を明示的に指定した属性検索を行う必要があります。例えば、次のようにします。SELECT doc FROM my\_docs WHERE CONTAINS (doc,ATTRIBUTE "year" BETWEEN 2001 AND 2005) = 1

## 数値属性

Net Search Extender には浮動小数点数を認識するパーサーが備わっています。

属性値のフォーマットの正しい例と誤った例を以下に示します。

表 1. 属性値でサポートされるフォーマット

正しいフォーマット	誤ったフォーマット
1000 1 000 1.000 - ピリオドは小数点文字	1,000
100 000 100 000.00123	1 000 000 - 1 と 0 の間にスペースが 2 つ

数値の小数部にはスペース文字を使用することができないことに注意してください。例えば、1 000.000 100 は 2 つの数 1000.000 と 100 であるものとして扱われます。

言語固有の区切り記号および言語固有の通貨フォーマットはサポートされていません。

## デフォルトの文書モデル

デフォルトの文書モデルのいずれかを使用するときは、すべてのフィールドが索引付けされており、特殊な情報は抽出されておらず、数値属性に索引付けされていないようにしなければなりません。

HTML、XML、および Outside-In フィルター文書の場合は、索引作成時に文書モデルを指定しない場合に使用されるデフォルトの文書モデルが Net Search Extender に用意されています。構造化プレーン・テキスト文書の場合、文書モデルを指定する必要があります。

デフォルトの文書モデルの 1 つを使用する場合は、

- すべてのフィールドに索引が付けられ、メタ情報などの特殊な情報は抽出されません。
  - HTML および INSO フォーマットの場合、各フィールドには対応するタグの名前が割り当てられます。
  - XML の場合、XML 文書のすべての XML ノードは、各ノードの完全修飾要素パスによって識別される重複フィールドにマップされます。例えば、パス /play/role/name です。
- 数値属性の索引は作成されません (数値属性がデフォルトの文書モデルでは定義されていないためです)。

表 2. サポートされる文書フォーマットに対するデフォルトの文書モデルの動作

文書タイプ	デフォルトの文書モデルの動作
HTML	次のタグをテキスト・フィールドとして受け入れる: <a> <address> <au> <author> <h1> <h2> <h3> <h4> <h5> <h6> <title>。フィールド名は、タグ名。例: "address"。
XML	すべてのタグをテキスト・フィールドとして受け入れる。フィールド名は、完全修飾要素パス名 (例: "/play/title")。
構造化プレーン・テキスト (GPP)	デフォルトの文書モデルなし。
Outside-In (INSO)	Outside-In フィルターによって戻されるものとして 195 ページの『Outside-In フィルター文書の文書モデルの定義』に示されている文書プロパティをテキスト・フィールドとして受け入れる。フィールド名は Outside-In が使用する文書プロパティの名前です。例: "SCCCA_TITLE"。

文書のタイプごとにデフォルトの文書モデルが定義されます。各モデルは異なっているため、以下のセクションでは、モデルごとに例および説明を記載してあります。

#### 注:

デフォルトの文書モデルでも正しく文書进行处理しますが、索引付けと検索の効率を上げるには、独自の文書モデルを定義する必要があります。

デフォルトの文書モデルを使用すると、文書のテキストは、テキスト・フィールドの一部であるかどうかに関係なく、完全に索引付けされます。したがって、テキスト検索に制約がない場合、そのテキストは検索の対象となります。

## 構造化プレーン・テキスト文書の文書モデルの定義

このトピックでは、文書モデル要素のパラメーターと制約事項を示します。

文書モデル要素のパラメーターは下記のとおりです。

**name** 定義ごとに名前をテキスト・フィールドまたは文書属性に割り当てます。この名前を使用すれば、検索照会を特定のテキスト・フィールドまたは文書属性の内容に制限できます。前述の例を使用すると、Abstract という名前の付いたテキスト・フィールド内で、structure という単語を含んだ文書を検索できます。

**start** テキスト・フィールドまたは文書属性の開始をマークするコード・ページ UTF-8 の境界ストリング。境界ストリングの指定に規則はありません。任意の UTF-8 ストリングとすることができます。例えば、次のような例が挙げられます。start="introduction:"、start="note!"、start="\$\$\_..."

印刷できない文字および特殊な XML 文字 "<" と "&" は、デフォルト XML 文字エンタリー ("<" の場合 "&lt;"、"&" の場合 "&amp;") を使用して指定する必要があります。

**end** オプション。テキスト・フィールドまたは文書属性の終了をマークするコード・ページ UTF-8 の境界ストリング。終了タグを指定しない場合、次に検

出された開始タグは、フィールドの終了と見なされます。これ以降、開始タグが検出されない場合は、フィールドは文書の終わりまで拡張され、これ以上のフィールドは識別されません。

**type** 文書属性のタイプは常に「NUMBER」でなければなりません。このパラメーターは、フィールド定義には適用されません。

**exclude**

YES または NO。フィールド定義内のテキストを除外して索引を付けないかどうかを決定するパラメーター。このパラメーターは、属性定義には適用されません。

この例では、フィールド定義「head」は除外されますが、定義「abstract」は含まれます。

**制約事項:**

- 同じ開始タグを持つ 2 つのフィールド定義または属性定義がないようにしてください。ただし、フィールド定義および属性定義は、同じ開始タグと終了タグをもつことはできます。
- 開始タグは別のタグに固有の接頭部であってはなりません。例えば、開始タグ "author" と開始タグ "authority" を指定することはできません。
- 開始タグおよび終了タグは空ストリングであってはなりません。

## GPP 文書を索引付けするとどうなるか

汎用パーサーは、開始境界ストリングの 1 つを検索して文書をスキャンします。開始境界ストリングを検出すると、対応する終了境界ストリングを検出するまで、後続のフィールドを解析します。

次に、定義用語に従って、つまりテキスト・フィールドまたは文書属性として、フィールドの内容に索引が付けられます。テキスト・フィールドと文書属性が同じ開始境界ストリングと終了境界ストリングをもつ場合、フィールドの内容には、テキスト・フィールドおよび文書属性の両方として索引が付けられます。

フィールドのネスティングは許可されていません。終了境界ストリングに到達する前に新たな開始境界ストリングが検出されると、新たな開始境界ストリングは通常のテキストとして解釈されます。

対応する終了境界ストリングが検出されない場合、フィールドは文書の終わりまで拡張していると想定されます。その場合は適切な理由コードが報告されます。

文書モデルで終了境界ストリングが指定されない場合、新規の開始境界ストリングが直前のフィールドの終了のシグナルを出します。

---

## HTML 文書の文書モデルの定義

HTML パーサーはテキストを UTF-8 コード・ページに変換します。その結果、HTML タグを認識し、タグ・クラスに分類します。

HTML パーサーはテキストを UTF-8 コード・ページに変換します。その結果、HTML タグを認識し、タグ・クラスに分類します。

- 無視されるタグ付き情報。例: フォント情報など。

- 定位置情報を提供するタグ。例: 新規の段落用の <p>; など。
- 構造上の情報を提供するタグ。例: <Title> など。

このパーサーは、HTML 4 で定義されたすべての文字エンティティ参照 (例: "&auml;" (ä) など) を認識し、UTF-8 での対応するコード・ポイントに分解します。

メタ・タグを認識し、メタ・タグ・テキストを解析します。

以下に HTML 文書の例を示します。

```
<HTML>
<HEAD>
<META NAME="year" CONTENT="2002">
<TITLE> The Firm </TITLE>
</HEAD>
<BODY>
<H1>Synopsis</H1>;
```

```
<H1>Prologue</H1>;:
:
</BODY>
```

以下に HTML 文書モデルの例を示します。

```
<?xml version="1.0"?>
<HTMLModel>

 <HTMLFieldDefinition
 name="subtitle"
 tag="title"
 exclude="NO" />

 <HTMLFieldDefinition
 name="header1"
 tag="h1"
 exclude="YES" /> - This is the start of text field
 - This is the end of the text field

 <HTMLAttributeDefinition
 name="year"
 tag="meta"
 meta-qualifier="year"
 type="NUMBER" /> - This is the start of the document
 attribute
 - This is the end of the document
 attribute

</HTMLModel>
```

1 行目の <?xml version="1.0"?> は、文書モデルが XML タグを使用して書かれていることを示しています。このモデルは XML フォーマット文書用には書かれていないことに注意してください。

各フィールドは、エレメント・パラメーターを含む HTMLFieldDefinition タグまたは HTMLAttributeDefinition タグ内で定義されます。

すべてのテキスト・フィールド定義は、<HTMLModel> タグ内に含まれる必要があります。タグ名は次のように、索引作成時にパラメーターとして渡されます。CREATE INDEX iA FOR TEXT ON T1(C2) DOCUMENTMODEL HTMLModel IN myModel.xml CONNECT TO db

文書モデル要素のパラメーターは下記のとおりです。

**name** 定義ごとに名前をテキスト・フィールドまたは文書属性に割り当てます。この名前を使用すれば、検索照会を特定のテキスト・フィールドまたは文書属性の内容に制限できます。前述の例を使用すると、`subtitle` という名前の付いたテキスト・フィールド内で、`firm` という単語を含んだ文書を検索できます。

**tag** 開始タグおよび (暗黙の) 終了タグがテキスト・フィールドまたは文書属性をマークする要素を識別します。その名前の要素内にあるテキストは、定義されたフィールドの内容を構成しています。

タグの大文字小文字は無視されます。

前述の例を使用すると、`H1` タグの後に続くテキストには、フィールド「`header1`」の一部としての索引が付けられます。このサンプル文書を基にした場合は、「`synopsis`」および「`prologue`」の索引が作成されます。

### meta-qualifier

このタグは、**タグ**要素と一緒に使用する必要があります。「`meta`」タグを指定することによって、`meta-qualifier` に一致する内容の値が抽出されます。

HTML 文書の例では、メタ・タグは次の要素を持ちます。

```
<META NAME="year" CONTENT="2002">
```

文書モデルの例では、`meta-qualifier` は「`year`」です。したがって、属性「`year`」の値として、内容「`2002`」の索引が作成されます。

**type** 文書属性タイプは「`NUMBER`」でなければなりません。このパラメーターは、フィールド定義には適用されません。

### exclude

YES または NO。フィールド定義内のテキストを除外して索引を付けないかどうかを決定するパラメーター。このパラメーターは、属性定義には適用されません。

この例では、フィールド定義「`header1`」は除外されますが、定義「`subtitle`」は含まれます。

文書のこれ以外のテキストはすべて索引付けされますが、フィールドの一部としてではありません。

## 要素パラメーター

文書モデル・エレメントのパラメーターは下記のとおりです。

**name** 定義ごとに名前をテキスト・フィールドまたは文書属性に割り当てます。この名前を使用すれば、検索照会を特定のテキスト・フィールドまたは文書属性の内容に制限できます。前述の例を使用すると、`subtitle` という名前の付いたテキスト・フィールド内で、`firm` という単語を含んだ文書を検索できます。

**tag** 開始タグおよび (暗黙の) 終了タグがテキスト・フィールドまたは文書属性をマークする要素を識別します。その名前の要素内にあるテキストは、定義されたフィールドの内容を構成しています。

タグの大文字小文字は無視されます。



前述の例を使用すると、H1 タグの後に続くテキストには、フィールド「header1」の一部としての索引が付けられます。このサンプル文書を基にした場合は、「synopsis」および「prologue」の索引が作成されます。

#### meta-qualifier

このタグは、**タグ**要素と一緒に使用する必要があります。「meta」タグを指定することによって、meta-qualifier に一致する内容の値が抽出されます。

HTML 文書の例では、メタ・タグは次の要素を持ちます。

```
<META NAME="year" CONTENT="2002">
```

文書モデルの例では、meta-qualifier は「year」です。したがって、属性「year」の値として、内容「2002」の索引が作成されます。

**type** 文書属性タイプは「NUMBER」でなければなりません。このパラメーターは、フィールド定義には適用されません。

#### exclude

YES または NO。フィールド定義内のテキストを除外して索引を付けないかどうかを決定するパラメーター。このパラメーターは、属性定義には適用されません。

この例では、フィールド定義「header1」は除外されますが、定義「subtitle」は含まれます。

文書のこれ以外のテキストはすべて索引付けされますが、フィールドの一部としてではありません。

---

## XML 文書の文書モデルの定義

XML 文書の文書モデルを使用することにより、XML 文書内で検索された要素をフィールド、文書属性、または両方にマップする方法を定義できます。

以下に XML 文書の例を示します。

```
<?xml version="1.0"?>
<purchaseOrder orderDate="2001-01-20"> [4]
 <shipAddress countryCode="US"> [1]
 <name>Alice Smith</name> [2]
 <street>123 Maple Street</street>
 <city>Mill Hill</city>
 <state>CA</state>
 <zip>90999</zip>
 </shipAddress>
 <item partNo="123" quantity="1"> [3]
 <name>S&B Lawnmower Type ABC-x</name>
 <price>239.90</price>
 <shipDate>2001-01-25</shipDate>
 </item>
 <item partNo="987" quantity="1"> [3]
 <name>Multifunction Rake ZYX</name>
 <price>69.90</price>
 <shipDate>2001-01-24</shipDate>
 </item>
</purchaseOrder>
```

前述のサンプル文書に一致する XML 文書モデルの例を以下に示します。

```

<?xml version="1.0"?>
<XMLModel>

<XMLFieldDefinition [1]
name="addresses"
locator="/purchaseOrder/shipAddress" />

<XMLFieldDefinition [2]
name="customerName"
locator="//shipAddress/name"
exclude="yes" />

<XMLAttributeDefinition [3]
name="partNumber"
type="NUMBER"
locator="/purchaseOrder//item/@partNo" />

<XMLFieldDefinition [4]
name="none"
locator="/purchaseOrder/@orderDate" />

</XMLModel>

```

1 行目の `<?xml version="1.0"?>` は、このモデルが XML を使用して書かれていることを示しています。各フィールドは、要素パラメーターを含む `XMLFieldDefinition` または `XMLAttributeDefinition` タグ内で定義されます。

すべてのテキスト・フィールド定義は、`<XMLModel>` タグ内に含まれる必要があることに注意してください。タグ名は次のように、索引作成時にパラメーターとして渡されます。 `CREATE INDEX i1 FOR TEXT ON T1(C2) DOCUMENTMODEL XMLModel in myModel.xml CONNECT TO db`

サンプル内のフィールドおよび属性は、この例のモデル・ファイル内の定義に対応する番号でマークされます。

フィールドのネストは許可されています。例えば、ある指定の XPath ロケーションで選択されているノードが、別の属性定義によって選択されている XML 要素内にある場合です。ネストされたフィールドは、前述のサンプル XML 文書に示されています。フィールド `addresses` は、XML 文書内でフィールド `customerName` が選択するノードの上位ノードを選択します。したがって、論理上、組み込みノードの内容は両方のフィールドに属します。テキスト・フィールドが重複することはあっても、それらのフィールド内のテキストが索引付けされるのは一回だけです。この例では、フィールド制限を指定して検索を実行すると、Alice Smith が `addresses` だけでなく `customerName` にも検出されます。しかし、ロケータ式のマッチング・セマンティクス上、同じ 1 つの XML ノードを複数のフィールドにマップすることはできません。

Net Search Extender は、XML 文書のコード・ページを探すことはしません。DB2 のコード・ページが取得されます。

フィールドの内容は、次の規則に従って決定されます。

- ロケーターがコメント、処理命令、または XML 属性に一致するフィールドの場合、フィールドの内容は実際のコメント・テキスト、処理命令テキスト、または属性値テキストです。



- XML 要素またはルート・ノードに一致するフィールドの場合、`exclude="YES"` が指定されたフィールドによって一致する要素の場合を除き、フィールドの内容は任意の組み込まれた要素からのテキストで構成されています。

文書は適格な XML で構成される必要がありますが、必ずしも DTD を XML 文書で指定する必要はありません。DTD の妥当性検査または外部エンティティー解決は実行されません。Net Search Extender は XML 文書を文書モデルに一致させるだけです。内部エンティティーは XML の要求どおりに置換されます。

## 要素パラメーター

文書モデル要素のパラメーターは下記のとおりです。

**name** 定義ごとに名前をテキスト・フィールドまたは文書属性に割り当てます。これらの名前を使用すれば、検索照会を特定のテキスト・フィールドまたは文書属性の内容に制限できます。

名前の中で次の変数の 1 つを使用できます。変数は、ソース・ドキュメントの一致する要素から生成されたストリングによって置き換えられます。

**変数 値**

**\$(NAME)**

XPath に一致する XML 要素の実際の修飾名 (QName)。

**\$(LOCALNAME)**

XPath に一致する XML 要素の実際のローカル名 (接頭部なし)。

**\$(PATH)**

XPath に一致する XML 要素のスラッシュおよびタグのシーケンスとしての実際の絶対パス (デフォルト文書モデル内で名前として使用)。

**type** 文書属性タイプは「NUMBER」でなければなりません。このパラメーターは、フィールド定義には適用されません。

**locator** 検索フィールドとして使用されるソース・ドキュメントの一部を選択する XPath 言語の式。

XML 文書モデル・ファイルの作成時には、ロケーター内の修飾名 (QNames と呼ばれる) が XML 文書内の特定のタグと同一でなければなりません。同一でないと、フィールドが全く認識されないため、フィールドを照会しても結果が戻されません。

ロケーターを以下に示します。

**purchaseOrder | salesOrder**

すべての purchaseOrder 要素および salesOrder 要素

**shipAddress**

すべての shipAddress 要素

\* すべてのエレメント (これは `child::*` の省略形 - 詳細については構文を参照してください)

**name/item**

name の親を持つすべての item 要素

**purchaseOrder//item**

purchaseOrder を発生源とするすべての item 要素

/ ルート・ノード

**comment()**

すべてのコメント・ノード

**processing-instruction()**

すべての処理命令

**attribute::\* (または @\*)**

すべての属性ノード

リテラルは、単一または二重引用符で囲まれるストリングです。端末トークンの正確な定義に関しては、XML 推奨を参照してください。

Net Search Extender 文書モデルでサポートされる XPath ロケーターは、XML Stylesheet Language Transformation (XSLT) パターンに類似しています。これらは、述部、関数 'id' および 'key'、あるいはノード・テスト 'text()' および 'node()' を含まない XSLT パターンのサブセットのみから構成されています。

**ignore** YES または NO。パラメーターを使用して、ロケーターに例外を指定します。

時には \* などの汎用ロケーターを指定して、索引付けしようとするノードを一致させたい場合があります。しかし、より具体的なロケーターに一致する一部のノードを索引付けさせないように指定することもできます。

これを公式化するには、索引付け時に無視するノードに、より具体的なロケーターを指定してフィールド定義を組み込みます。続いて、汎用ロケーター(下記のセクションを参照)による優先順位よりも高い優先順位をこのロケーターに与えて、ignore="yes" を指定します。これは、一致するノードに対してフィールド情報を生成してはならないことをインデクサーに指示します。

このような無視されたノードが索引作成者生成ノードに組み込まれた場合でも、無視されたノードの内容は、フィールド生成ノードの内容にも属するので、索引付けされることに注意してください。

**priority**

特定のロケーターによって検出された定義に与えられる優先順位を指定する -1 から +1 の間の浮動小数点数。

優先順位を指定しない場合は、デフォルトの優先順位が使用されます。

- | によって区切られた複数の選択肢は、一連の定義 (各選択肢ごとに 1 つの定義) として処理されます。
- 単一の名前によって一致するロケーター。つまり、次のいずれかの書式を持つロケーターは、デフォルトの優先順位が 0 になります。
  - ChildOrAttributeAxisSpecifier QName
  - ChildOrAttributeAxisSpecifier processing-instruction(Literal))
- ChildOrAttributeAxisSpecifier NCName:\* の書式を持つロケーターは、デフォルトの優先順位が -0.25 になります。

- ChildOrAttributeAxisSpecifier NodeTest の書式を持つロケータは、デフォルトの優先順位が -0.5 になります。
- これ以外のロケータは、デフォルトの優先順位が 0.5 になります。

ロケータがより具体的であるほど、デフォルトの優先順位も高くなることに注意してください。例えば、具体的でないロケータ \* が、定義内で低い優先順位を与えられるのに対して、名前はより具体的なロケータであり、より高い優先順位が与えられます。

また、あるノードが複数のロケータと一致した場合、優先順位を割り当てることによって、どの定義が選択されたかを判別できることにも注意してください。優先順位が最も高い定義が選択されます。2 つの定義が同じ優先順位であった場合、最新のものが選択されます。

この競合解決は、XML Stylesheet Language Transformation (XSLT) で使用されるものと同一です。

### exclude

YES または NO。フィールド定義内のテキストを除外して索引を付けないかどうかを決定するパラメータ。このパラメータは、属性定義には適用されません。

この例では、フィールド定義「customerName」は除外されますが、定義「addresses」は含まれます。

**注:** ロケータ値が同じで名前が異なるフィールド定義が複数ある場合は、最後のフィールド定義のみが有効です。その前のフィールド定義はすべて無視されます。

以下のモデル・ファイルには、1 つのフィールド定義が含まれています。

```
<XMLFieldDefinition
name="from"
exclude="NO"
locator="/document/email/from" />
```

索引作成後に、セクション from 内で検索すると文書が検出されます。

次に、モデル・ファイルの末尾に別のフィールド定義を追加します。

```
<XMLFieldDefinition
name="from_last"
exclude="NO"
locator="/document/email/from" />
```

索引を再作成して更新した後は、セクション from\_last 内で検索すると文書が検出されますが、セクション from 内で検索すると文書は検出されません。

---

## Outside-In フィルター文書の文書モデルの定義

Outside-In フォーマットの文書モデルは、特定のタグ・セットによって識別される構造要素を Net Search Extender テキスト・フィールドおよび文書属性にマップできるという点で、HTML 文書モデルに非常によく似ています。

例えば、Microsoft Word 文書のセットを持っていて、文書プロパティの "title"、"subject"、および "keyword" をフィールドとして、文書プロパティ

"author" および "category" を文書属性として索引付けするとします。次の例の Outside-In 文書モデルを使えば、このマッピングが行えます。

```
<?xml version="1.0"?>
<INSOModel>

<INSOFieldDefinition
name="title"
tag="SCCCA_TITLE"/>

<INSOFieldDefinition
name="title"
tag="SCCCA_SUBJECT"/>

<INSOFieldDefinition
name="title"
tag="SCCCA_KEYWORDS"/>

<INSOAttributeDefinition
name="author"
tag="SCCCA_AUTHOR"
type="STRING"/>

<INSOAttributeDefinition
name="category"
tag="SCCCA_CATEGORY"
type="STRING"/>

</INSOModel>
```

## 要素パラメーター

文書モデル要素のパラメーターは下記のとおりです。

- name** テキスト・フィールドまたは文書属性に割り当てる名前。各フィールド定義にはフィールド名を、各属性定義には属性名を割り当てます。これらの名前を使うことにより、照会は特定のテキスト・フィールドの内容に検索を制限したり、特定の属性を持つ文書を検索したりできます。
- tag** タグを識別します。このタグの開始および終了 (または暗黙終了) 要素によって、テキスト・フィールドや文書属性が明示されます。指定された名前の要素内にあるテキストは、定義されるフィールドまたは属性の内容を構成します。タグの大文字小文字は無視されます。可能な値を以下の段落で説明します。
- type** 文書属性のタイプは "NUMBER"、"DATE"、または "STRING" のいずれかにすることができます。このパラメーターは、フィールド定義には適用されません。
- exclude** YES または NO。フィールド定義内のテキストを除外して索引を付けないかどうかを決定するパラメーター。このパラメーターは、属性定義には適用されません。

Outside-In 文書モデルは、それぞれが名前とタグを定義するフィールド定義および属性定義から構成されています。属性定義の場合はタイプも必須ですが、フィールド定義にはオプションの "exclude" フラグがあります。HTML モデルの場合と同じように、このような定義の名前属性は文書部分のマップ先となる Net Search Extender フィールドまたは属性を定義します。これは、任意の UTF-8 テキスト・ストリング

にすることもできます。追加情報については、*Outside-In Content Access Specification, Version 7.5* を参照してください。

Outside-In 開始、終了および文書プロパティのタグに関連するタグ属性が取り得る値のリストについての詳細は、『*Outside-In タグ属性値*』を参照してください。

## Outside-In 文書を索引付けするとどうなるか

デフォルトでは、すべてのテキストはどのフィールドにも属さないものとして索引付けされます。

テキストのストリームの中で現れた開始タグが、現在アクティブである文書モデルのある定義項目と一致すると、開始タグとそれに対応する終了タグの間にあるテキストは、その定義項目に従って処理されます。例えば、索引付けされたフィールドとして、除外されたフィールドとして、属性として、またはそれらの複合として処理されます。

一致する定義がない場合は、開始タグとそれに対応する終了タグが無視されます。

Outside-In フィルターは文書のフォーマットとコード・ページを自動的に認識するので、CCSID 仕様には影響しません。Outside-In フィルターがフォーマットとコード・ページを正確に判別できない場合、文書は ASCII ファイルとして処理されます。



---

## 第 54 章 文書モデルの参照情報

Net Search Extender は、文書モデルに関するさまざまな情報を提供します。

Net Search Extender は、文書モデルの以下の参照情報を提供します。

- 文書モデル用の DTD
- ロケーター (XPath) 式のセマンティクス
- テキスト・フィールドおよび文書属性の制限
- Outside-In タグ属性値

---

### 文書モデル用の DTD

このトピックでは、文書タイプ定義 (DTD) の形式の文書モデルの構文を正式に記述したものを示します。

文書タイプ定義 (DTD) の形式の文書モデルの構文を正式に記述したものを次に示します。

```
<!ELEMENT GPPModel (GPPFieldDefinition|GPPAttributeDefinition)+>
<!ELEMENT HTMLModel (HTMLFieldDefinition|HTMLAttributeDefinition)+>
<!ELEMENT XMLModel (XMLFieldDefinition|XMLAttributeDefinition)+>

<!ELEMENT GPPFieldDefinition EMPTY>
<!ATTLIST GPPFieldDefinition name CDATA #REQUIRED>
<!ATTLIST GPPFieldDefinition start CDATA #REQUIRED>
<!ATTLIST GPPFieldDefinition end CDATA #IMPLIED>
<!ATTLIST GPPFieldDefinition exclude (YES|NO) NO>

<!ELEMENT GPPAttributeDefinition EMPTY>
<!ATTLIST GPPAttributeDefinition name CDATA #REQUIRED>
<!ATTLIST GPPAttributeDefinition start CDATA #REQUIRED>
<!ATTLIST GPPAttributeDefinition end CDATA #REQUIRED>
<!ATTLIST GPPAttributeDefinition type NUMBER #REQUIRED>

<!ELEMENT HTMLFieldDefinition EMPTY>
<!ATTLIST HTMLFieldDefinition name CDATA #REQUIRED>
<!ATTLIST HTMLFieldDefinition tag CDATA #REQUIRED>
<!ATTLIST HTMLFieldDefinition meta-qualifier CDATA #IMPLIED>
<!ATTLIST HTMLFieldDefinition exclude (YES|NO) NO>

<!ELEMENT HTMLAttributeDefinition EMPTY>
<!ATTLIST HTMLAttributeDefinition name CDATA #REQUIRED>
<!ATTLIST HTMLAttributeDefinition tag CDATA #REQUIRED>
<!ATTLIST HTMLAttributeDefinition meta-qualifier CDATA #IMPLIED>
<!ATTLIST HTMLAttributeDefinition type NUMBER #REQUIRED>

<!ELEMENT XMLFieldDefinition EMPTY>
<!ATTLIST XMLFieldDefinition name CDATA #REQUIRED>
<!ATTLIST XMLFieldDefinition locator CDATA #REQUIRED>
<!ATTLIST XMLFieldDefinition ignore (YES|NO) NO>
<!ATTLIST XMLFieldDefinition priority CDATA #IMPLIED>
<!ATTLIST XMLFieldDefinition exclude (YES|NO) NO>

<!ELEMENT XMLAttributeDefinition EMPTY>
<!ATTLIST XMLAttributeDefinition name CDATA #REQUIRED>
<!ATTLIST XMLAttributeDefinition locator CDATA #REQUIRED>
```

```
<!ATTLIST XMLAttributeDefinition ignore (YES|NO) NO>
<!ATTLIST XMLAttributeDefinition priority CDATA #IMPLIED>
<!ATTLIST XMLAttributeDefinition type NUMBER #REQUIRED>
```

---

## ロケーター (XPath) 式のセマンティクス

XPath 式はコンテキスト・ノードを基準にして解釈する必要があり、一連のノードを表します。Net Search Extender セレクター・パターンとして使用する場合、コンテキスト・ノードは自由です。つまり、相対パス・パターン `p` は `//p` として解釈されます。

XML データ・モデルに従えば、XML 文書は、以下の種類のノードを含むツリーのように見えます。

- ルート・ノード
- 要素ノード
- テキスト・ノード
- 属性ノード
- 名前空間ノード
- 処理命令ノード
- コメント・ノード

これらのノード間のリンク、つまりツリー構成リレーションシップは、XML 文書の直接の包含リレーションシップを反映しています。

ルート・ノードはルートにのみ存在することができ、ツリーの他の場所には存在できません。このノードには、子として、文書要素、オプションのコメント、および処理命令が含まれます。

要素ノードには、ルート・ノード以外のあらゆる種類のノードが含まれます。その他の種類のノードは、ツリーのリーフ・ノードとしてのみ許可されます。

「子」、「属性」、および「名前空間」の、3 つの種類の包含リンクがあります。「属性」および「名前空間」の包含リンクは、属性ノードおよび名前空間ノードにつながっている必要があります。このため、要素ノードの子にアクセスするには(グラフ理論からすれば)、「属性」リンクに従ってすべての包含属性を見つけ、「名前空間」リンクに従ってすべての包含名前空間宣言を見つけ、さらに、「子」リンクに従って包含された要素、テキスト・ノード、処理命令、およびコメントを見つける必要があります。

XPath 式はコンテキスト・ノードを基準にして解釈する必要があり、一連のノードを表します。Net Search Extender セレクター・パターンとして使用する場合、コンテキスト・ノードは自由です。つまり、相対パス・パターン `p` は `//p` として解釈されます。

Net Search Extender XPath セレクター・パターンを以下に示します。

- コンテキスト `N` での Pattern `'|'` LocationPathPattern は、両方ともコンテキスト `N` にある Pattern および LocationPathPattern に一致するノードの和集合を表します。



- コンテキスト N での '/'RelativePathPattern は、この RelativePathPattern がルート・コンテキストで表すすべてのものを表します。
- コンテキスト N での '//RelativePathPattern は、ルートの子孫 (子軸上の) であるすべてのコンテキストで、この RelativePathPattern が表すと解釈されるものの和集合を表します。
- RelativePathPattern '/' StepPattern はコンテキスト N 内のノードに一致します。ただし、そのノードがその親のコンテキストでの StepPattern に一致し、さらに、その親ノードがコンテキスト N での RelativePathPattern に一致する場合には限られます。
- RelativePathPattern '//' StepPattern はコンテキスト N 内のノードに一致します。ただし、そのノードがその親のコンテキストでの StepPattern に一致し、さらに、コンテキスト N での RelativePathPattern に一致する上位ノードをもつ場合に限られます。
- コンテキスト N での 'child'::NodeTest (省略構文: NodeTest) は、N の子 (子軸上の) であり、NodeTest を満たすノードに一致します。
- コンテキスト N での 'attribute'::NodeTest (省略構文: @NodeTest) は、N の属性であり、NodeTest を満たすノードに一致します。
- NodeType '(' ' ' ) は、指定したタイプのノードの場合のみ条件を満たします。
- 'processing-instruction' '(' Literal ' ) は、リテラルを名前として持つすべての処理命令タイプのノードを満たします。
- '\*' は、すべての要素ノードまたは属性ノード (要素名の名前マスク) を満たします。
- NCName ':' '\*' は、名前の接頭部として NCName をもつすべての要素ノードを満たします。
- QName は、指定した名前をもつすべてのノードを満たします。

## 注

NameTest の形式の NodeTest では、ノードは選択した軸上の基本タイプ、つまり、属性軸上の属性タイプおよび子軸上の子タイプであると想定されます。したがって、NameTest が使用できるのは、子ノードおよび属性ノードを選択する場合のみです。コメント・ノードまたは処理命令ノードの選択には使用できません。さらに、このパターンは、名前空間ノード以外のあらゆる種類のノードの選択に使用できますが、軸指定子「名前空間」が許可されていないため、名前空間ノードの選択には使用できません。

パターン例は以下のとおりです。

- chapter | appendix は、すべての章要素および付録要素を表します。
- table はすべての表要素を表します。
- \* はすべての要素を表します (これは child::\* の省略形です)。
- ulist/item は、ulist 親をもつすべての項目要素を表します。
- appendix//subsection は上位の付録をもつすべてのサブセクションを表します。
- / はルート・ノードのみを含む singleton セットを表します。
- comment() はすべてのコメント・ノードを表します。
- processing-instruction() はすべての処理命令を表します。

- `attribute::*` (または `@*`) はすべての属性ノードを表します。

以下は、ロケータ要素の構文です。

```
Locator ::= LocationPathPattern
 | Locator '|' LocationPathPattern
LocationPathPattern ::= '/' RelativePathPattern ?
 | '//'? RelativePathPattern
RelativePathPattern ::= StepPattern
 | RelativePathPattern '/' StepPattern
 | RelativePathPattern '//'? StepPattern
StepPattern ::= ChildOrAttributeAxisSpecifier NodeTest
ChildOrAttributeAxisSpecifier ::=
 ('child' | 'attribute') '::'
 | '@'?
NodeTest ::= NameTest
 | NodeType '(' ')'
 | 'processing-instruction' '(' Literal ')'
NameTest ::= '*' | NCName ':' '*' | QName
NodeType ::= 'comment' | 'processing-instruction'
```

NCName および QName は、XML 命名の推奨で定義されたとおりです。

#### NCName

コロンを含まない XML 名

#### QName

NCName を先行させることができる NCName で、先行する NCName の後にはコロンが続きます。例: NCName:NCName

---

## テキスト・フィールドおよび文書属性の制限

ここでは、テキスト・フィールドおよび文書属性の制限のリスト、および HTML 文書モデルに入れることができるタグのリストを示します。

以下に、テキスト・フィールドおよび文書属性の制限を示します。

- 索引でのフィールドの最大数: 32767
- 1 つの文書での、STRING タイプの 1 つの属性の値の最大数: 1024
- STRING タイプの属性の最大数: 253
- STRING 属性値の文字数は 128 に切り捨てられる
- DATE タイプおよび NUMBER タイプの属性の最大数: 32766
- DATE または NUMBER の属性値の文字数は 128 に切り捨てられる
- NUMBER 属性の場合は、倍精度の浮動小数点数がナンバーとして受け入れられる
- 1 つの文書での、DATE タイプまたは NUMBER タイプの 1 つの属性に指定できる値の最大数: 無制限

HTML 文書モデルに組み込むことのできるタグを以下に示します。

- <A>
- <ADDRESS>
- <AU>
- <AUTHOR>
- <H1>

- <H2>, <H3>, <H4>, <H5>
- <H6>
- <TITLE>

その他のタグを含む場合がある <HEAD> および <BODY> などのタグは、HTML 文書モデルでテキスト・フィールドとして指定することはできません。

---

## Outside-In タグ属性値

このトピックでは、Outside-In 文書プロパティのタグ・タイプに関連するタグ属性が取ることのできる値を示します。

Outside-In 文書プロパティのタグ・タイプに関連するタグ属性が取ることのできる値を以下に示します。

SCCCA\_ABSTRACT  
 SCCCA\_ACCOUNT  
 SCCCA\_ADDRESS  
 SCCCA\_ATTACHMENTS  
 SCCCA\_AUTHORIZATION  
 SCCCA\_BACKUPDATE  
 SCCCA\_BASEFILELOCATION  
 SCCCA\_BILLTO  
 SCCCA\_BLINDCOPY  
 SCCCA\_CARBONCOPY  
 SCCCA\_CATEGORY  
 SCCCA\_CHECKEDBY  
 SCCCA\_CLIENT  
 SCCCA\_COMPANY  
 SCCCA\_COMPLETEDDATE  
 SCCCA\_COUNTCHARS  
 SCCCA\_COUNTPAGES  
 SCCCA\_COUNTWORDS  
 SCCCA\_CREATIONDATE  
 SCCCA\_DEPARTMENT  
 SCCCA\_DESTINATION  
 SCCCA\_DISPOSITION  
 SCCCA\_DIVISION  
 SCCCA\_DOCCOMMENT  
 SCCCA\_DOCTYPE  
 SCCCA\_EDITMINUTES  
 SCCCA\_EDITOR  
 SCCCA\_FORWARDTO  
 SCCCA\_GROUP  
 SCCCA\_KEYWORD  
 SCCCA\_LANGUAGE  
 SCCCA\_LASTPRINTDATE  
 SCCCA\_LASTSAVEDBY  
 SCCCA\_MAILSTOP  
 SCCCA\_MANAGERSCCCA\_MATTER  
 SCCCA\_OFFICE  
 SCCCA\_OPERATOR  
 SCCCA\_OWNER  
 SCCCA\_PRIMARYAUTHOR  
 SCCCA\_PROJECT  
 SCCCA\_PUBLISHER  
 SCCCA\_PURPOSE  
 SCCCA\_RECEIVEDFROM  
 SCCCA\_RECORDEDBY  
 SCCCA\_RECORDEDDATE  
 SCCCA\_REFERENCE  
 SCCCA\_REVISIONDATE  
 SCCCA\_REVISIONNOTES

SCCCA\_REVISIONNUMBER  
SCCCA\_SECONDARYAUTHOR  
SCCCA\_SECTION  
SCCCA\_SECURITY  
SCCCA\_SOURCE  
SCCCA\_STATUS  
SCCCA\_SUBJECT  
SCCCA\_TITLE  
SCCCA\_TYPIST  
SCCCA\_USERDEFINEDPROP  
SCCCA\_VERSIONDATE  
SCCCA\_VERSIONNOTES  
SCCCA\_VERSIONNUMBER

Outside-In 開始および終了タグのサブタイプに関連するタグ属性が取ることのできる値を以下に示します。

SCCCA\_ALTFONTDATA  
SCCCA\_ANNOTATIONREFERENCE  
SCCCA\_CAPTIONTEXT  
SCCCA\_CHARACTER  
SCCCA\_COMPILEDFIELD  
SCCCA\_COUNTERFORMAT  
SCCCA\_CUSTOMDATAFORMAT  
SCCCA\_DATEDEFINITION  
SCCCA\_DOCUMENTPROPERTYNAME  
SCCCA\_ENDNOTEREFERENCE  
SCCCA\_FONTANDGLYPHDATA  
SCCCA\_FOOTNOTEREFERENCE  
SCCCA\_FRAME  
SCCCA\_GENERATEDFIELD  
SCCCA\_GENERATOR  
SCCCA\_HYPERLINK  
SCCCA\_INDEX  
SCCCA\_INDEXENTRY  
SCCCA\_INLINEDATAFORMAT  
SCCCA\_LISTENTRY  
SCCCA\_MERGEENTRY  
SCCCA\_NAMEDCELLRANGE  
SCCCA\_REFERENCEDTEXT  
SCCCA\_STYLE  
SCCCA\_SUBDOCTEXT  
SCCCA\_TOA  
SCCCA\_TOAENTRY  
SCCCA\_TOC  
SCCCA\_TOCENTRY  
SCCCA\_TOF  
SCCCA\_VECTORSAVETAG  
SCCCA\_XREF

表にはすべての文書プロパティ、および INSO フィルターが認識するすべてのタグ・サブタイプが含まれているので注意してください。2つのサブタイプ例外があります。SCCCA\_DOCUMENTPROPERTY および SCCCA\_BOOKMARK。

---

## 第 13 部 リファレンス



## 第 55 章 インスタンス所有者用の管理コマンド

インスタンス所有者の管理は、Net Search Extender の状況チェック、ロックングおよび更新サービス、さらにこれらのサービスの開始および停止から構成されています。

このセクションでは、インスタンス所有者用の管理コマンドの構文について説明しています。インスタンス所有者の管理は、Net Search Extender の状況チェック、ロックングおよび更新サービス、さらにこれらのサービスの開始および停止から構成されています。

これらのコマンドは **db2text** コマンドのサブコマンドであり、これらを使用して DB2 インスタンスに固有の Net Search Extender サービスの管理を行うことができます。

コマンド	目的
『CONTROL コマンド』	フルテキストの索引ロックをリストし、削除する。キャッシュ状態もリストする。
209 ページの『START コマンド』	Net Search Extender インスタンス・サービスを開始する。
211 ページの『STOP コマンド』	Net Search Extender インスタンス・サービスを停止する。
217 ページの『DB2EXTHL コマンド』	強調表示 UDF の入力パラメーターの最大サイズを変更する。

### CONTROL コマンド

Net Search Extender インスタンス・サービスによって管理されるフルテキストの索引ロックをリストしたり削除することができます。

#### 目的

ロックング・サービスおよび更新サービスが実行中の場合は、その状況、およびアクティブ化されたキャッシュに関する情報を表示できます。

パーティション・データベース環境では、現行パーティションにのみ影響を及ぼします。ユーザーには、必要なパーティションに対して DB2 コマンド **db2\_a11** を呼び出す責任があります。

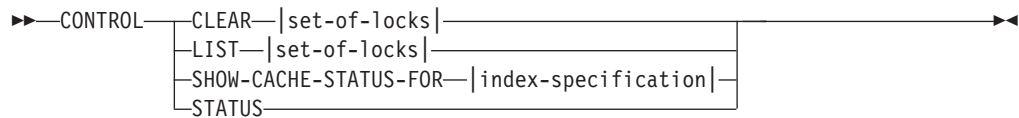
#### 許可

正常にコマンドを発行するには、ユーザーは DATAACCESS を持つ DBADM 権限を所有する、DB2 インスタンス所有者でなければなりません。

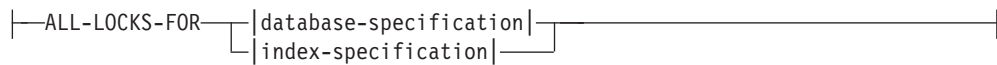
#### 必須接続

このコマンドは DB2 データベース・サーバーから発行する必要があります。

## コマンド構文



### set-of-locks:



### index-specification:



### Database-specification:



## コマンド・パラメーター

**CLEAR** **CLEAR** を使用して一連のロックを強制的にクリーンアップします。ロックに関する問題の原因を確認した後で、このコマンドを慎重に使用してください。

**CLEAR** コマンドの適用先の索引で、索引の更新などのいずれかの索引管理コマンドがアクティブなままになっている場合は、**CLEAR** コマンドを使用しないでください。使用すると、索引が破損し、完全な再作成が必要になることがあります。

**LIST** 特定の索引またはデータベース用に保持されている現行のロックについて、情報を得るには、**LIST** を使用します。更新ロックが存在する場合、コマンドは、それまでに処理された文書数に関する情報も印刷します。

これは、索引にロックが保持されている場合にだけあてはまることに注意してください。

レプリケーション・キャプチャー表を使用する場合、更新操作はありません。その代わりに、挿入操作は、索引が作成されたソース表での挿入または更新操作のいずれかによるものである場合があります。

### *set-of-locks*

指定されたデータベースまたは索引のロックについてのみ処理を行います。

### **SHOW CACHE STATUS FOR**

指定された索引のキャッシュ表のアクティブ化状況を表示します。これは、「アクティブ化されていない (Not Activated)」または「現在アクティブ化されている (Currently Activated)」のいずれかになります。キャッシュがアクティブ化されている場合、キャッシュ・メモリーの使用に関する詳細が表示



されます。例えば、最大キャッシュ・サイズ (メガバイト)、挿入できる文書の最大数、およびキャッシュ表に残っているスペース (キロバイト) などについてです。

**STATUS STATUS** キーワードを使用することによって、このコマンドは Net Search Extender インスタンス・サービスのロッキング・サービスおよび更新サービスが稼働中であるかどうかを表示します。

**DATABASE** *database-name*  
使用中のサーバー上のデータベースの名前。

**INDEX** *index-schema.index-name*  
現在、使用中のテキスト索引のスキーマおよび名前。これは **CREATE INDEX** コマンドで指定されます。

## 使用上の注意

管理コマンド・エラー・メッセージがロックに関する問題の発生を示している場合、競合するタスクが実行中でないことを確認してください。例えば、**UPDATE** コマンドの実行中に **ALTER** コマンドを実行しようとしていないか、などです。

増分索引更新に **SHOW CACHE STATUS FOR** を使用して、指定されたメモリー・サイズが、次の更新時にも引き続き更新情報をすべて収容できるだけの大きさを持っているか、アクティブ化が実行済みであるかどうかをチェックしてください。

---

## START コマンド

フルテキスト索引のロッキング、および DB2 サーバー上のフルテキスト索引の自動更新を制御するデーモンを開始します。

注: コマンドは、索引の一時キャッシュ表をまったくアクティブ化しないため、ストアード・プロシージャーを使用した検索を開始する前に、個別の **ACTIVATE CACHE** コマンドを実行する必要があります。

### 許可

インスタンス所有者は、現在の DB2 インスタンスに対して DATAACCESS を持つ DBADM 権限を保持している必要があります。

### 必須接続

このコマンドは DB2 データベース・サーバーから発行する必要があります。

### コマンド構文

▶▶—START—◀◀

### コマンド・パラメーター

なし

## 使用上の注意

Windows では、このコマンドはサービス DB2EXT - *instance\_name* [-nodenum] を開始します。コントロール パネルまたは **NET START** コマンドを使用して、1 つ以上のサービスを開始することもできます。ただし、ターミナル・サービス・クライアントから Net Search Extender を開始することはできません。

パーティション・データベースと共に使用される DB2 インスタンスでは、標準的な Windows での方法を使用する代わりに、**db2text start** を使用して Net Search Extender インスタンス・サービスを開始することを強くお勧めします。

サービス管理コンソールを使用して、DB2 インスタンスに対する各 DB2EXT サービスを手動で開始または停止することができます。しかし、Net Search Extender を正常な実行状態に維持するためには、DB2 インスタンスに関連する DB2EXT サービスをすべて開始し、またシャットダウンする場合もすべてシャットダウンする必要があります。また、手動で開始および停止する際には、以下の順序に従う必要があります。

**開始** NSE (DB2EXT) サービスの開始は、他の DB2EXT サービスを開始する前に、ホスト上で最も小さい数字のパーティションから始めて、最も大きい数字のパーティションまで順番に作業するようにしてください。

**停止** NSE (DB2EXT) サービスの停止は、ホスト上で最も大きい数字のパーティションから始めて、最も小さい数字のパーティションまで順番に作業するようにしてください。

ただし、**db2text start** および **db2text stop** を使用する場合、システムは自動的にこの必須順序に従います。

DB2EXT サービスでは、「自動」スタートアップ・タイプはサポートされていません。DB2 インスタンスの各 DB2EXT サービスに対して、**db2text start** または **net start** を実行することにより、DB2EXT サービスを手動で開始する必要があります。システムが完全に開始される前に、テキスト検索操作を実行しようとしなくてください。

**START** が失敗した場合は、既に存在しない索引を参照している古いエントリーがスケジューラー内にまだ存在している可能性があります。ファイル `../sql1lib/db2ext/ctedem.dat` を編集して、古いエントリーをすべて除去してください。**START** コマンドを再実行してください。

Net Search Extender が正常に開始されると、システムで **ctelock** プロセス (Windows の場合は **ctelock.exe**) がアクティブになります。数個の共用リソース (共用メモリーおよびセマフォ) が作成されて、UNIX マシンの /tmp ディレクトリーに格納されます。これらのファイルは Net Search Extender で必要です。インスタンスが実行されている間は削除しないでください。一方、**STOP** の後で **START** コマンドが失敗した場合は、/tmp ディレクトリー内の古いアンカー・ファイルがすべて削除されていることを確認してください。すべてのユーザーが /tmp ディレクトリーの読み取り、書き込み、および実行のアクセス権を持っている必要があります。

## STOP コマンド

Net Search Extender のロッキング・サービスおよび更新サービスを停止します。

### 許可

インスタンス所有者は、現在の DB2 インスタンスに対して DATAACCESS を持つ DBADM 権限を保持している必要があります。

### 必須接続

このコマンドは DB2 データベース・サーバーから発行する必要があります。

### コマンド構文

```
▶▶ STOP [FORCE] ▶▶
```

### コマンド・パラメーター

**FORCE** 処理がロックを保留中であつたり、キャッシュ表が任意の索引に対してアクティブ化されている場合であっても、サービスを停止します。**FORCE** を指定しないと、コマンドはこれらのケースでアクティブ・キャッシュに関する警告を出して失敗します。

### 使用上の注意

Net Search Extender インスタンス・サービスを停止すると、特定の Net Search Extender コマンドはそれ以上使用できなくなります。以前、アクティブ化されたキャッシュを索引に対して使用した場合は、サービスを再始動するときに、一時キャッシュを再びアクティブ化する必要があります。

パーティション・データベースと共に使用される DB2 インスタンスでは、標準的な Windows での方法を使用する代わりに、**db2text stop** を使用して Net Search Extender インスタンス・サービスを停止することを強くお勧めします。

サービス管理コンソールを使用して、DB2 インスタンスに対する各 DB2EXT サービスを手動で停止することができます。しかし、NSE を正常な実行状態に維持するためには、DB2 インスタンスに関連するすべての DB2EXT サービスをシャットダウンする必要があります。また、手動で停止する場合、ホスト上で最も大きい数字のパーティションから始めて、最も小さい数字のパーティションまで順番に作業するようにして NSE (DB2EXT) サービスを停止する必要があります。ただし、**db2text stop** を使用する場合、システムは自動的にこの必須順序に従います。

Net Search Extender が正常に停止されると、**ctelock** プロセス (Windows の場合は **ctelock.exe**) が終了します。UNIX マシンの /tmp ディレクトリーにある共用リソースおよびアンカー・ファイルが削除されます。



---

## 第 56 章 データベース管理者用の管理コマンド

データベース管理者は、複数のコマンドを実行して、データベースを Net Search Extender が使用するようにセットアップすることができます。

このセクションでは、データベース管理者用の管理コマンドの構文について説明しています。データベース管理コマンドには、Net Search Extender が使用するためのデータベースのセットアップ、およびこのセットアップを使用できないようにする作業が含まれています。

**ENABLE DATABASE** および **DISABLE DATABASE** コマンドのみは、**DB2TEXT** コマンドのバリエーションですが、次のすべてのコマンドは、データベース・レベル上の管理を可能にします。

コマンド	目的
『ENABLE DATABASE コマンド』	現行データベースがフルテキスト索引を作成できるようにする。
215 ページの『DISABLE DATABASE コマンド』	データベース用に Net Search Extender によって完了した準備作業をリセットする。
217 ページの『DB2EXTHL コマンド』	<b>DB2EXTHL</b> コマンドは、強調表示 UDF の入力パラメーターの最大サイズを変更します。

### ヒント

**db2text** コマンドの一部としてデータベース接続情報が指定されていない場合、**db2text** 実行可能コードは、環境変数 **DB2DBDFT** で指定されたデフォルトのサブシステムに対する暗黙接続を行います。

---

## ENABLE DATABASE コマンド

データベースがテキスト列上にフルテキスト索引を作成して使用できるようになります。**ENABLE DATABASE** コマンドは、検索のための管理表、ビュー、ユーザー定義関数 (UDF)、およびストアド・プロシージャなどの Net Search Extender インフラストラクチャーをデータベースに作成します。

### 許可

ユーザーは **ENABLE DATABASE** コマンドを実行するための **DBADM** 特権を持っている必要があります。

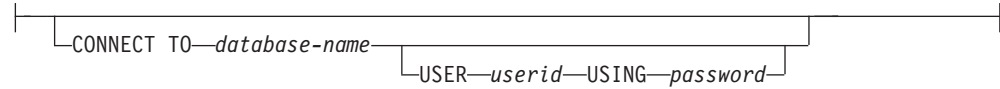
### 前提条件

インスタンス所有者は、**DATAACCESS** を持つ **DBADM** 権限を保持している必要があります。バージョン 9.7 では、**SYSADM** は **SECADM** および **DBADM** 特権を保持しなくなりました。**ENABLE DATABASE** コマンドを実行する前に、**SECADM** はインスタンス所有者に **DATAACCESS** を持つ **DBADM** 権限を明示的に付与しなければなりません。

## コマンド構文



### connection-options:



## コマンド・パラメーター

### **CONNECT TO** *database-name*

このコマンドのターゲットであるデータベースの名前。環境変数 **DB2DBDFT** が設定されていて、必要な DB2 権限を持つユーザー ID の下でユーザーがコマンドを実行している場合は、このパラメーターを省略できます。

### **USER** *userid***USING***password*

*userid* および *password* を使用してデータベースに接続します。

### **AUTOGRANT**

このオプションを指定すると、インスタンス所有者がこのデータベースに対して **DATAACCESS** 特権のある **DBADM** を持っていない場合は、これらの特権の付与が試行されます。特権を正常に付与するには、ユーザーはデータベースに対する **SECADM** 特権を持っている必要があり、ユーザーがインスタンス所有者になることはできません (ユーザーは特権を自分自身に付与することはできません)。

**注:** このオプションはストアード・プロシージャ・インターフェースではサポートされていません。

## 使用上の注意

このコマンドは、接続されたデータベースを **Net Search Extender** で使用するために準備します。これは必須のステップであり、このステップを実行しないと、データベース内の表/列に **Net Search Extender** 索引を作成できなくなります。

コマンド実行後に確立されるデータベース・デフォルトは、**DB2EXT.DBDEFAULTS** カタログ・ビューを使用して表示できます。

### データベースへの変更

このコマンドは、使用可能になったデータベースの **DB2** インスタンスに関連した **DB2** インスタンス所有者に **DBADM** 権限を付与します。

**ENABLE DATABASE** コマンドは、**Net Search Extender** カタログ、UDF およびストアード・プロシージャなどの各種のデータベース・オブジェクトをスキーマ **DB2EXT** に作成します。コマンド実行後に、次のカタログ・ビューが使用できます。

```
db2ext.dbdefaults
db2ext.textindexes
db2ext.textindexformats
db2ext.indexconfiguration
```

これらの表は、データベースのデフォルトの表スペース (IBMDEFAULTGROUP) に配置されます。この表スペースは、db2nodes.cfg で定義されたノード全域で配布されます。

ファイル・システムへの変更  
なし。

---

## DISABLE DATABASE コマンド

Net Search Extender がデータベースに加えた変更を元に戻します。

### 許可

ユーザーは **DISABLE DATABASE** コマンドを実行するための DBADM 特権を持っている必要があります。

### 前提条件

インスタンス所有者は、DATAACCESS を持つ DBADM 権限を保持している必要があります。バージョン 9.7 では、SYSADM は SECADM および DBADM 特権を保持しなくなりました。**DISABLE DATABASE** コマンドを実行する前に、SECADM はインスタンス所有者に DATAACCESS を持つ DBADM 権限を明示的に付与しなければなりません。

### コマンド構文

```
▶▶—DISABLE DATABASE FOR TEXT—|FORCE| |connection-options|▶▶
```

#### connection-options:

```
|CONNECT TO—database-name| USER—userid—USING—password|
```

### コマンド・パラメーター

#### CONNECT TO *database-name*

このコマンドのターゲットであるデータベースの名前。DB2DBDFT が設定され、必要な DB2 権限を持つユーザー ID の下でユーザーがコマンドを実行している場合は、このパラメーターを省略できます。

#### USER *userid* USING *password*

*userid* および *password* を使用してデータベースに接続します。

#### FORCE

データベース内のすべての Net Search Extender 索引を強制的にドロップさせます。

## 使用上の注意

このコマンドは、接続されたデータベースをリセットするため、他の Net Search Extender コマンドからそのデータベースを使用することはできなくなります。データベース内にフルテキスト索引がある場合は、**FORCE** オプションを使用しない限り、このコマンドは失敗します。

このコマンドは、DB2 インスタンス所有者から DBADM 権限を除去しません。

**注:** データベースにテキスト索引が定義されている場合に、そのデータベースを使用不可にしようとする、失敗します。これらの索引を 1 つ 1 つ除去した上で、何か問題が生じるかどうかを検査するようお勧めします。disable database for text force コマンドを使用した場合に保証されるのは、データベース内の Net Search Extender カタログ表が削除されるということに過ぎません。

しかし、完全にドロップできない索引があった場合は、手作業でクリーンアップしなければならないリソースがまだあるかもしれません。以下のものが含まれます。

- 索引ディレクトリー、作業ディレクトリー、およびキャッシュ・ディレクトリー内のファイル
- ctedem.dat 内のスケジューラー項目
- レプリケーション・キャプチャー・オプションを使用して索引を作成した場合には、リモート・データベースの表にある、IBMSNAP\_SIGNAL、IBMSNAP\_PRUNE\_SET、および IBMSNAP\_PRUNCNTL の各項目を手作業で削除しなければなりません。これらの項目は、APPLY\_QUAL="NSE" || <instance name> and TARGET\_SERVER= database\_name コマンドで簡単に識別できます。

次の例では、インスタンスは DB2、データベースは SAMPLE です。

```
DELETE FROM <ccSchema>.IBMSNAP_SIGNAL
WHERE SIGNAL_INPUT_IN IN
 (SELECT MAP_ID FROM <ccSchema>.IBMSNAP_PRUNCNTL
 WHERE APPLY_QUAL= 'NSEDB2' AND TARGET_SERVER= 'SAMPLE');

DELETE FROM <ccSchema>.IBMSNAP_PRUNCNTL
WHERE APPLY_QUAL= 'NSEDB2' AND TARGET_SERVER= 'SAMPLE';

DELETE FROM <ccSchema>.IBMSNAP_PRUNE_SET
WHERE APPLY_QUAL= 'NSEDB2' AND TARGET_SERVER= 'SAMPLE';
```

### データベースへの変更

Net Search Extender を使用可能にする際にデータベースに対して行われる以下の変更は削除されます。

- データベース内での Net Search Extender カタログ・ビュー。
- Net Search Extender によって作成されるすべてのデータベース・オブジェクト。

### ファイル・システムおよび共有メモリーへの変更

**FORCE** オプションを使用する場合、索引ファイルは削除されます。

**FORCE** オプションを使用する場合、索引のアクティブ化されたキャッシュに関するキャッシュが削除されます。



---

## DB2EXTHL コマンド

強調表示 UDF の入力パラメーターの最大サイズを変更する。

### 目的

デフォルトでは、強調表示 UDF は入力として最大 100 KB のサイズの文書を受け取り、200 KB の CLOB を戻します。データベースにある最大の文書のサイズに応じて、入力値を大きくすることができます。最大サイズは 1 GB です。

### 許可

正常にこのコマンドを発行するには、ユーザーは DATAACCESS を持つ DBADM 権限を所有する、DB2 インスタンス所有者でなければなりません。

### 必須接続

このコマンドは DB2 データベース・サーバーから発行する必要があります。また、DB2DBDFT 環境変数が必要です。

### コマンド構文

▶▶—db2exth1—*new-highlight-input-size*————▶▶

### コマンド・パラメーター

*new-highlight-input-size*

強調表示 UDF の新しい結果サイズ (K バイト)。これは 1048576 よりも小さい正の整数です。



## 第 57 章 テキスト表所有者用の管理コマンド

表所有者は、表の中のテキスト索引を変更するコマンドを実行できます。

このセクションでは、テキスト表所有者用の管理コマンドの構文について説明しています。

これらのコマンドは **DB2TEXT** コマンドのサブコマンドです。このコマンドを使用すると、表の所有者は、表の列についてのフルテキスト索引を作成し、それを操作できます。

コマンド	目的
220 ページの『ACTIVATE CACHE コマンド』	キャッシュをアクティブ化する。この結果、ストアード・プロシージャを使用した検索が可能になる
221 ページの『ALTER INDEX コマンド』	索引の特性を変更する
225 ページの『CLEAR EVENTS コマンド』	索引更新時に使用される索引イベント表から索引イベントを削除する
227 ページの『CREATE INDEX コマンド』	フルテキスト索引を作成する
242 ページの『DEACTIVATE CACHE コマンド』	キャッシュを非アクティブ化する。この結果、ストアード・プロシージャを使用した検索が実行できなくなる
245 ページの『DB2EXTTH コマンド』	シソーラス定義ファイルをコンパイルする
243 ページの『DROP INDEX コマンド』	テキスト列用のフルテキスト索引をドロップする
246 ページの『RESET PENDING コマンド』	SET INTEGRITY コマンドを実行した後、拡張ステージング・インフラストラクチャーによって、基本表に対してテキスト保持従属表をリフレッシュする
247 ページの『UPDATE INDEX コマンド』	テキスト列の現行の内容に基づいて索引付け処理を開始する
251 ページの『HELP コマンド』	<b>DB2TEXT</b> コマンド・オプションのリストを表示する
252 ページの『COPYRIGHT コマンド』	Net Search Extender の製品および著作権情報を表示する

### ヒント:

**db2text** コマンドの一部としてデータベース接続情報が指定されていない場合、**db2text** 実行可能コードは、環境変数 **DB2DBDFT** で指定されたデフォルトのサブシステムに対する暗黙接続を行います。

## ACTIVATE CACHE コマンド

DB2 ユーザー表または永続キャッシュ・ファイルのいずれかからキャッシュ済みの表をアクティブ化します。完了後は、ストアード・プロシージャを使用した検索操作が可能です。

このコマンドは、索引が **CACHE TABLE** オプションを指定して作成された場合にのみ使用可能であることにも注意してください。

### 許可

ステートメントの許可 ID には、以下の特権が少なくとも 1 つ含まれていなければなりません。

- テキスト索引が定義されている表またはニックネームに対する **CONTROL** 特権
- **DBADM** 権限

### 前提条件

インスタンス所有者は、**DATAACCESS** を持つ **DBADM** 権限を保持している必要があります。バージョン 9.7 では、**SYSADM** は **SECADM** および **DBADM** 特権を保持しなくなりました。**ACTIVATE CACHE** コマンドを実行する前に、**SECADM** はインスタンス所有者に **DATAACCESS** を持つ **DBADM** 権限を明示的に付与しなければなりません。

### コマンド構文

```
▶▶ ACTIVATE CACHE FOR INDEX index-schema-"." index-name FOR TEXT ▶▶
▶▶ RECREATE |connection-options| ▶▶
```

#### connection-options:

```
| CONNECT TO database-name USER userid USING password |
```

### コマンド・パラメーター

#### *index-schema*

**CREATE INDEX** コマンドで指定されたテキスト索引のスキーマ。スキーマが指定されていない場合は、DB2 接続のユーザー ID が使用されます。

#### *index-name*

テキスト索引の名前 (**CREATE INDEX** コマンドで指定されたもの)。

#### **RECREATE**

永続キャッシュを使用する索引にのみ適用されます。既存のキャッシュは削除されます。アクティブ化なしで更新が完了した場合は、永続キャッシュはデータベースから自動的に再構成されます。

### CONNECT TO *database-name*

このコマンドのターゲットであるデータベースの名前。**DB2DBDFT** が設定されており、ユーザーがサーバーでコマンドを実行している場合は、このパラメーターを省略できます。ユーザー ID には必須 DB2 許可がなければならぬことに注意してください。

### USER *userid* USING *password*

*userid* および *password* を使用してデータベースに接続します。指定しない場合は、パスワードなしで現行ユーザー ID から接続が試行されます。

## 使用上の注意

次のコマンドのいずれかが索引上で現在実行中の場合は、このコマンドを発行できません。

- **UPDATE INDEX**
- **ALTER INDEX**
- **DROP INDEX**
- **CLEAR EVENTS**
- **DEACTIVATE CACHE**

注: 永続キャッシュが使用された場合でも、キャッシュ済み表のアクティブ化の際に、キャッシュ表を最初から再び作成しなければならない可能性があります。これは、永続キャッシュが非アクティブ化されている間に、更新操作が実行された場合に発生します。

キャッシュの構築に使用されるメモリーの量は、現行の文書数と結果の列のサイズから、動的に計算されます。**PCTFREE** 値を使用して、 $100/(100-PCTFREE)$  の係数で計算されたメモリーの最少量を増やします。**PCTFREE** 値は **CREATE INDEX** または **ALTER INDEX** コマンドに指定される値です。

したがって、**PCTFREE** は、キャッシュがアクティブ化されている場合に挿入操作に確保された、割り振り済みキャッシュのパーセンテージを表します。**ACTIVATE CACHE** コマンドを実行するたびに、実際のメモリー・サイズが再評価されるため注意してください。

### ファイル・システムへの変更

永続キャッシュをインプリメントするためのファイルが作成されます。

---

## ALTER INDEX コマンド

**ALTER INDEX** コマンドで、フルテキスト索引の特性を変更できます。

### 目的

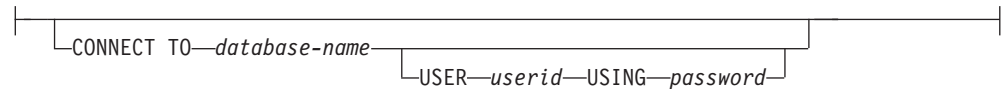
コマンドは、更新オプションおよび保管オプションのような、フルテキスト索引の特性を変更します。

### 許可

ステートメントの許可 ID には、以下の特権が少なくとも 1 つ含まれていなければなりません。



## connection-options:



## コマンド・パラメーター

### *index-schema*

**CREATE INDEX** コマンドで指定されたテキスト索引のスキーマ。スキーマが指定されていない場合は、DB2 接続のユーザー ID が使用されます。

### *index-name*

**CREATE INDEX** コマンドで指定されたテキスト索引の名前。

### **INDEX DIRECTORY** *directory*

テキスト索引の保管先ディレクトリー・パス。ディレクトリーは索引データを含むので、ディレクトリーには、DB2 インスタンス所有者ユーザー ID 用の読み取り、書き込み、および実行権限があることを確認してください。

パーティション・データベース環境では、このディレクトリーはすべてのパーティション上に存在する必要があることに注意してください。サブディレクトリー `NODE<nr>` が、サーバーの論理パーティション上の索引を区別するために、ディレクトリーの下に作成されます。以前の索引ディレクトリーからの索引ファイルは、すべて削除されます。

### **WORK DIRECTORY** *workdirectory*

検索操作と管理操作中に一時ファイルを保管します。新規索引ディレクトリーとは無関係に、別個の作業ディレクトリーを変更できます。

ディレクトリーが存在しない場合は、DB2 インスタンス所有者ユーザー ID を使用して作成されます。ディレクトリーが存在する場合は、ディレクトリーがインスタンス所有者用に UNIX プラットフォーム上での読み取り、書き込み、および実行権限を持っていることを確認してください。

パーティション・データベース環境では、このディレクトリーはすべてのパーティション上に存在する必要があることに注意してください。サブディレクトリー `NODE<nr>` が、サーバーの論理パーティション上の索引を区別するために、ディレクトリーの下に作成されます。以前の索引ディレクトリーからの一時索引ファイルは、すべて削除されます。

### **CACHE TABLE PERSISTENT IN** *directory*

非アクティブ化またはシステム・リブートの後であっても、**CREATE INDEX** のキャッシュ付きの表が永続的であるように指定します。いずれの場合でも、迅速な **ACTIVATE CACHE** の実行が可能になります。永続キャッシュは、指定されたディレクトリーに保管されます。

以前に作成された永続キャッシュは、新しいロケーションに移されます。この操作には、常に非アクティブ化された索引が必要です。

### **CACHE TABLE TEMPORARY**

キャッシュ結果表が、現在一時的であり、それまで存在した永続キャッシュが削除されたことを指定します。この変更操作には、非アクティブ化された索引が必要であることに注意してください。

#### **MAXIMUM CACHE SIZE** *memsize*

**ACTIVATE CACHE** 時に作成されるキャッシュ済み表の新たな最大サイズを指定します。*memsize* パラメーターは、メガバイト単位で正の整数として指定します。

*memsize* が小さすぎると、**ACTIVATE CACHE** コマンドは失敗します。実際のキャッシュ・サイズは **ACTIVATE CACHE** コマンド時に計算されます。この変更には、非アクティブ化された索引が必要です。

#### **PCTFREE** *percentage*

追加の文書のためにフリーにしておくキャッシュのパーセンテージを指定します。パーセンテージは 0 以上で 100 より小さい整数値でなければなりません。それまでの永続キャッシュは削除され、この変更には非アクティブ化された索引が必要であることに注意してください。

#### **UPDATE FREQUENCY**

以下のパラメーターを使用すると、索引更新の頻度は更新が実行される時期を決定します。

- **D** 索引が更新される曜日: \* (毎日) または 0..6 (0=日曜日)
- **H** 索引が更新される時間: \* (毎時) または 0..23
- **M** 索引が更新される分: 0..59
- **NONE** これ以上の索引更新はありません。このパラメーターは、これ以上変更されないテキスト列や、今後は手動更新のみを実行する場合のために用意してあります。

**UPDATE FREQUENCY** キーワードを指定しない場合、頻度設定は未変更のままです。

#### **UPDATE MINIMUM** *minchanges*

索引が増大されて更新されるまでに、テキスト文書に許可される最小の変更の数。 **UPDATE MINIMUM** キーワードを指定しないと、設定は変更されません。

**RECREATE ON UPDATE** オプションを使用して索引を作成しなかった場合は、**UPDATE MINIMUM** しか変更できないことに注意してください。

#### **COMMITCOUNT FOR UPDATE** *count*

更新処理の場合、コミット・カウントを指定できます。これは、**UPDATE** コマンド、および更新処理をスケジュールする **UPDATE FREQUENCY** 指定の両方に適用されます。

初期の更新では、**COMMITCOUNT** 値は無視されます。

**RECREATE ON UPDATE** オプションを使用して索引を作成しなかった場合は、**COMMITCOUNT** しか変更できないことに注意してください。

また、索引に **REPLICATION** 節を指定して作成した場合は、**COMMITCOUNT** を変更できないことにも注意してください。

#### **CONNECT TO** *database-name*

このコマンドのターゲットであるデータベースの名前。**DB2DBDFT** が設定されており、ユーザーがサーバーでコマンドを実行している場合は、このパラメーターを省略できます。ユーザー ID には必須 DB2 許可がなければならぬことに注意してください。



**USER** *userid* **USING** *password*

*userid* および *password* を使用してデータベースに接続します。指定しない場合は、パスワードなしで現行ユーザー ID から接続が試行されます。

## 使用上の注意

次のコマンドのいずれかが索引上で実行されている場合は、`alter index` コマンドを発行できないことに注意してください。

- **ALTER INDEX**
- **CLEAR EVENTS**
- **ACTIVATE CACHE**
- **DROP INDEX**
- **UPDATE INDEX**
- **DEACTIVATE CACHE**

キャッシュ・オプションを指定して索引を作成する場合、索引がアクティブになっているときは、索引ディレクトリーに対して **ALTER INDEX** コマンドを使用できません。まず始めに、キャッシュを非アクティブにします。

パーティション・データベース環境では、キャッシュ・オプション付きのテキスト索引は、単一パーティション表スペース上のみ許可されています。

### データベースへの変更

Net Search Extender カタログ・ビューを変更します。

### ファイル・システムへの変更

- 索引ディレクトリーおよび作業ディレクトリーに `NODE<nr>` サブディレクトリーを作成します。
- 索引ファイルを移動します。
- 永続キャッシュ・ディレクトリーを作成します。
- 永続キャッシュ・ファイルを移動します。

---

## CLEAR EVENTS コマンド

索引のイベント・ビューから索引付けイベントを削除します。イベント・ビューの名前は、`DB2EXT.TEXTINDEXES` ビューの `EVENTVIEWNAME` 列にあります。

### 許可

ステートメントの許可 ID には、以下の特権が少なくとも 1 つ含まれていなければなりません。

- テキスト索引が定義されている表またはニックネームに対する **CONTROL** 特権
- **DBADM** 権限

### 前提条件

インスタンス所有者は、**DATAACCESS** を持つ **DBADM** 権限を保持している必要があります。バージョン 9.7 では、**SYSADM** は **SECADM** および **DBADM** 特権を保持しなくなりました。 **CLEAR EVENTS** コマンドを実行する前に、**SECADM** はイン

スタンス所有者に DATAACCESS を持つ DBADM 権限を明示的に付与しなければなりません。

## コマンド構文

```
▶▶ CLEAR EVENTS FOR INDEX index-schema-".." index-name FOR TEXT
▶▶ COMMITCOUNT count | connection-options ▶▶
```

### connection-options:

```
▶▶ CONNECT TO database-name | USER userid USING password ▶▶
```

## コマンド・パラメーター

### *index-schema*

**CREATE INDEX** コマンドで指定されたテキスト索引のスキーマ。スキーマが指定されていない場合は、DB2 接続のユーザー ID が使用されます。

### *index-name*

**CREATE INDEX** コマンドで指定されたテキスト索引の名前。

### **COMMITCOUNT** *count*

INTEGER 値  $\geq 0$  を指定することにより、1 つのトランザクションで DB2 によって削除される行数を示します。

### **CONNECT TO** *database-name*

このコマンドのターゲットであるデータベースの名前。DB2DBDFT が設定されており、ユーザーがサーバーでコマンドを実行している場合は、このパラメーターを省略できます。ユーザー ID には必須 DB2 許可がなければならぬことに注意してください。

### **USER** *userid* **USING** *password*

*userid* および *password* を使用してデータベースに接続します。指定しない場合は、パスワードなしで現行ユーザー ID から接続が試行されます。

## 使用上の注意

**CREATE INDEX** または **ALTER INDEX** コマンドで **UPDATE FREQUENCY** オプションを使用して通常の更新をスケジュールする場合は、イベント表を定期的にチェックしてください。エラーを示している各イベントの理由を確認し、イベント表に示されたエラーの原因を除去した後で、**CLEAR EVENTS** を使用してイベント表をクリーンアップします。

次のコマンドのいずれかが索引上で実行中の場合は、clear events コマンドを発行できないことに注意してください。

- **UPDATE INDEX**
- **ALTER INDEX**

- **ACTIVATE CACHE**
- **DEACTIVATE CACHE**
- **DROP INDEX**

---

## CREATE INDEX コマンド

Net Search Extender フルテキスト照会を使用するために、テキスト列上にフルテキスト索引を作成します。

### 目的

パーティション・データベース環境では、フルテキスト索引は、ユーザー表が定義された表スペースのすべてのパーティションに作成されます。表スペースの配布に対する以降の変更は許可されておらず、管理コマンドおよび検索処理において、予期しない動作につながります。

### 許可

ステートメントの許可 ID には、以下の特権が少なくとも 1 つ含まれていなければなりません。

次のいずれか 1 つ

- **DBADM** 権限
- テキスト索引が定義されている表またはニックネームに対する **CONTROL** 特権
- テキスト索引が定義されている表またはニックネームに対する **INDEX** 特権

さらに次のいずれか 1 つ

- 索引の暗黙的または明示的スキーマ名が存在しない場合、データベースに対する **IMPLICIT\_SCHEMA** 権限
- 索引のスキーマ名が既存のスキーマを参照する場合、スキーマに対する **CREATEIN** 特権

### 前提条件

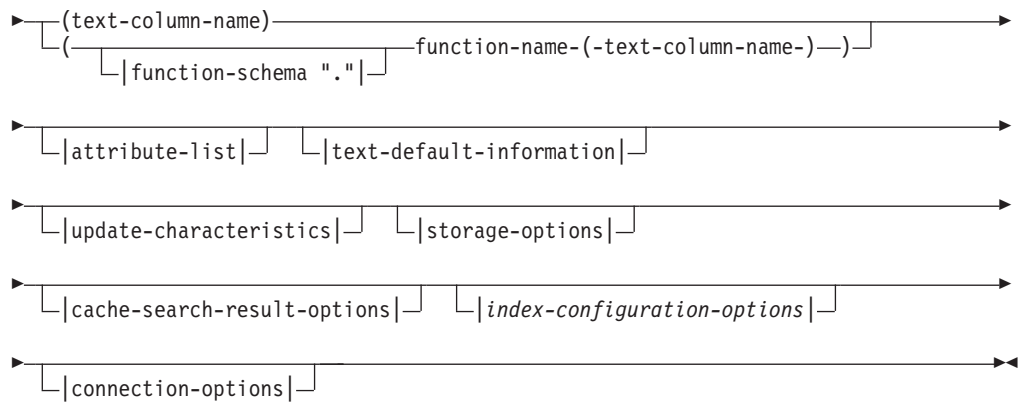
インスタンス所有者は、**DATAACCESS** を持つ **DBADM** 権限を保持している必要があります。バージョン 9.7 では、**SYSADM** は **SECADM** および **DBADM** 特権を保持しなくなりました。**CREATE INDEX** コマンドを実行する前に、**SECADM** はインスタンス所有者に **DATAACCESS** を持つ **DBADM** 権限を明示的に付与しなければなりません。

### コマンド構文

```

▶▶ CREATE INDEX [index-schema-"."] index-name FOR TEXT ON
▶ [table-schema-"."] table-name

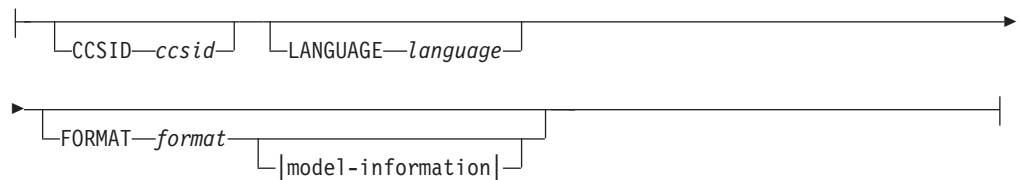
```



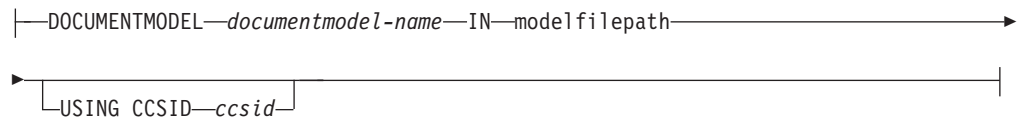
### attribute-list:



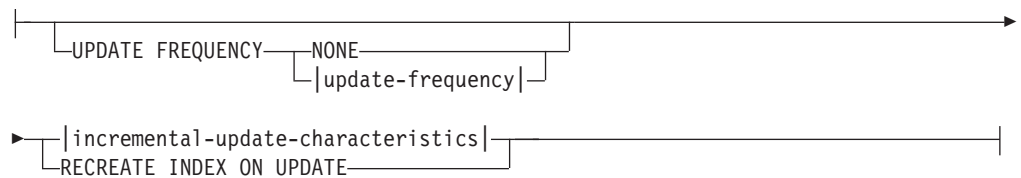
### text-default-information:



### model-information:



### update-characteristics:

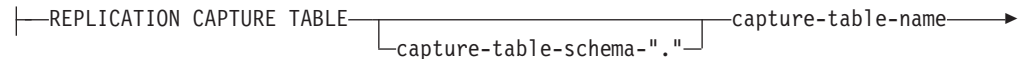


### incremental-update-characteristics:

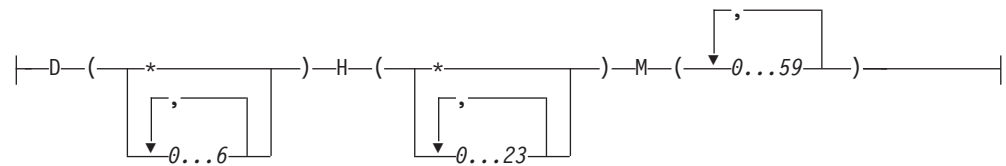




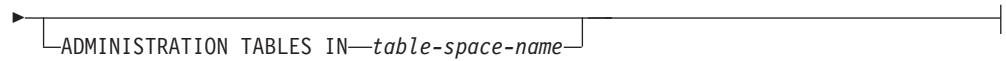
**capture-table-characteristics:**



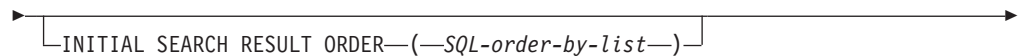
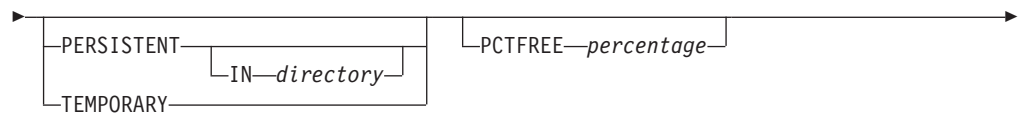
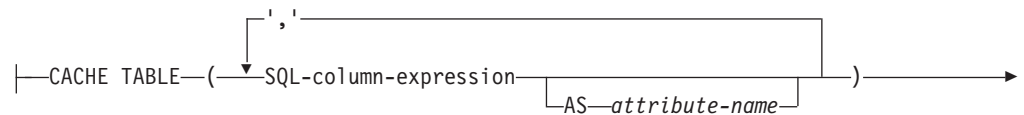
**update-frequency:**



**storage-options:**



**cache-search-results-options:**



KEY COLUMNS FOR INDEX ON VIEW—(SQL-columnname-list)

#### index-configuration-options:

INDEX CONFIGURATION—(option-value)

#### connection-options:

CONNECT TO—database-name—USER—userid—USING—password

## コマンド・パラメーター

### *index-schema*

テキスト索引のスキーマ。索引特定の管理表の DB2 スキーマ名として使用します。スキーマが指定されていない場合は、DB2 接続のユーザー ID が使用されます。索引スキーマは、有効な DB2 スキーマ名でなければならないことに注意してください。

### *index-name*

索引の名前。索引スキーマとともに、この名前はデータベースのフルテキスト索引を一意に識別します。

索引名は、有効な DB2 索引名でなければならないことに注意してください。

### *table-schema*

索引が作成される表、ニックネーム、またはビューのスキーマ。スキーマが指定されていない場合は、DB2 接続のユーザー ID が使用されます。

### *table-name*

フルテキスト索引が作成される列を含む、接続されたデータベース内のテキスト表、ニックネーム、またはビューの名前。

表名が DB2 基本表を参照しない場合は、次の制限があることに注意してください。

- ビューは、ストアード・プロシージャ検索および表値関数検索のみを許可します。したがって、索引またはビューのキー列を **KEY COLUMNS FOR INDEX ON VIEW** 節を使用して指定する必要があります。
- キャプチャー表を使用しないニックネームへの増分索引更新の場合、ログ表が作成されます。ニックネーム表またはビューのデータに変更がある場合は、手動でログ表を指定する必要があります。基本表の場合、この作業は自動的に実行されるので、ユーザーはログ表に触れてはなりません。
- DB2 の述部 CONTAINS、SCORE、および NUMBEROFMATCHES は、基本表またはニックネーム上の索引にのみ許可されており、ビュー上の索引には許可されていません。

- キャッシュ検索結果オプションをコマンドに指定した場合にのみ、ビュー上の索引は許可されます。
- 増分更新をサポートするための、トリガーに基づくログを補完する拡張テキスト保持ステージング・インフラストラクチャーは、パーティションおよび非パーティション基本表に適用することができます。しかし、ビューまたはニックネームに対してはサポートされていません。

#### *text-column-name*

フルテキスト索引の作成に使用されるテキストを含む列の名前。列は以下のいずれかのデータ・タイプである必要があります。

- CHAR (FOR BIT DATA)
- VARCHAR (FOR BIT DATA)
- LONG VARCHAR (FOR BIT DATA)
- CLOB
- DBCLOB
- BLOB
- GRAPHIC
- VARGRAPHIC
- LONG VARGRAPHIC
- XML

列のデータ・タイプがこれらのどれにも当てはまらない場合は、

***function-schema.function-name*** を使用して、トランスフォーメーション関数を指定し、サポートされる列タイプに変換します。

同じ列に複数の索引が許可されていますが、これは次のいずれかの条件を満たす場合のみであることに注意してください。

#### 索引がビュー上に作成されている

したがって、索引は CONTAINS、SCORE または NUMBEROFMATCHES 検索索引数では使用できません。

#### 索引が表上に作成されている

作成されたすべての索引が同期されている場合には、**CREATE INDEX** コマンドで指定される以下の内容について、同じ列のプロパティが等しくなります。

- 関数名およびスキーマ
- ATTRIBUTES
- CCSID
- LANGUAGE
- FORMAT
- DOCUMENTMODEL
- INDEX CONFIGURATION

この場合、CONTAINS、SCORE または NUMBEROFMATCHES 引数によって選択された索引は問題なく使用することができます。

#### *function-schema.function-name*

サポートされないタイプの列にあるテキスト文書にアクセスするために使用

されるユーザー定義関数のスキーマおよび名前。関数は、任意の列タイプを入力パラメーターとして使用して、列タイプの変換を実行します。 Net Search Extender でサポートされるタイプの 1 つとして値を戻します。

#### **ATTRIBUTES** (*SQL-column-expression AS Attribute-name, ...*)

列式の内容がテキスト列に加えて索引付けされることを決定します。これによって、この内容が検索ステートメントの **ATTRIBUTES** 節によって検索できるようになります。SQL 列式は、索引が作成された表の非修飾列名を使用して定義する必要があります。唯一、許可されたデータ・タイプは、**DOUBLE** です。Cast 演算子を列式で使用することができますが、DB2 の暗黙的なキャストは可能ではありません。属性名は、文書モデルの属性名の規則に従う必要があります、索引のモデル定義ファイル内で定義される他のすべての属性名と区別される必要があります。

以下の規則を使用して、式の属性名を定義します。

- 列式の SQL **AS** 節で明示的に指定した場合、指定した名前を使用する。  
例:

```
ATTRIBUTES (C1+C2 AS myname)
```

- 指定された表の列が **AS** を指定せずに使用された場合、列の名前が使用される。例えば、以下のように指定します。

```
ATTRIBUTES (C1)
```

- 式が **AS** を指定せずに使用され、名前付き列を参照しない場合は、**CREATE INDEX** はエラーを報告します。

例えば、以下のように指定します。

```
ATTRIBUTES (CAST(JULIAN_DAY(date) AS DOUBLE) as day, (price1+price2)/2 as avg_price)
```

単一引用符で囲まなかった属性は、大文字にマップされるため、検索時には大文字で指定する必要があることに注意してください。

#### **CCSID** *ccsid*

テキスト文書を索引付けする際に使用されるコード化文字セット ID。デフォルト値は、DB2EXT.DBDEFAULTS ビューからとられます。ここで、**DEFAULTNAME='CCSID'** です。

列のデータ・タイプがバイナリーである場合は、**CCSID** のみを設定してください。

#### **LANGUAGE** *language*

言語パラメーターは、索引構成値 **IndexStopWords** に 0 (索引付け中はストップワードを無視する) を設定した場合に選択されるストップワード・ディクショナリーの言語を指定します。タイ語 (**TH\_TH**) の場合はタイ語の単語の切れ目に対応するために、トルコ語の場合はドット付きとドットなしの "i" を正しく区別するために、必ずこのパラメーターを設定する必要があります。

#### **FORMAT** *format*

列内の **HTML** などのテキスト文書のフォーマット。この情報は、文書の索引作成時に必要になります。

構造化文書フォーマットの場合、文書モデル・ファイルで情報を指定できます。文書モデルが指定されない場合、文書のテキストは、デフォルトの文書モデルを使用して索引付けされます。



フォーマット・キーワードが指定されていない場合、デフォルト値は、`DEFAULTNAME='FORMAT'` である `DB2EXT.DBDEFAULTS` ビューから取得されます。Net Search Extender によって設定される初期デフォルトは `TEXT` です。データ・タイプ `XML` では、`FORMAT XML` 指定子が必須です。

#### **DOCUMENTMODEL** *documentmodel-name* **IN** *modelfilepath*

*modelfilepath* は、モデル・ファイルのロケーションを指定します。

*modelfilepath* は、完全修飾パスである必要があります。モデル・ファイルには、**FORMAT** 節のフォーマット用のモデル定義が含まれています。これは、`DB2` インスタンス所有者が読み取り可能なものでなければなりません。文書モデルを使用することにより、文書の特定のセクションを索引付けし、検索することができます。マークアップ・タグおよびセクション名も、文書モデルで定義できます。文書モデルは、`HTML`、`XML`、`GPP` 構造をサポートする文書フォーマットに結合されています。1 つのモデル・ファイルでは、1 つの文書モデルしか指定できません。

文書モデルは、**CREATE INDEX** コマンド実行時にのみ読み取られるため、この索引に対する以降のあらゆる変更は認識されないことに注意してください。

パーティション・データベース環境では、*modelfilepath* が `Linux` または `UNIX` オペレーティング・システムのすべてのノード上でアクセス可能になるようにファイル共有システムを使用する必要があることに注意してください。ただし `Windows` オペレーティング・システムでは、文書モデル・ファイルの *modelfilepath* を各ノードで使用可能なローカル・パスに設定しなければなりません。

#### **USING CCSID** *ccsid*

モデル・ファイルの内容を解釈する `CCSID` を指定します。デフォルト値は、`DB2EXT.DBDEFAULTS` ビューからとられます。ここで、`DEFAULTNAME='MODELCCSID'` です。

#### **UPDATE FREQUENCY**

索引更新の頻度は、更新が実行される時期を決定します。ユーザー表への変更が **UPDATE MINIMUM** オプションによって指定された値より小さい場合は、索引は更新されません。**UPDATE FREQUENCY** を指定しないと、デフォルトの `NONE` が使用され、索引の更新は行われません。これは、テキスト列に対するこれ以上の変更を想定していない場合や、更新処理を手動制御する場合に有用です。

- **D** 索引が更新される曜日: \* (毎日) または 0..6 (0=日曜日)
- **H** 索引が更新される時間: \* (毎時) または 0..23
- **M** 索引が更新される分: 0..59
- **NONE** これ以上の索引更新はありません。更新は、手作業で開始する必要があります。

デフォルト値は、`DB2EXT.DBDEFAULTS` ビューからとられます。ここで、`DEFAULTNAME='UPDATEFREQUENCY'` です。

**UPDATE FREQUENCY** パラメーターを使用して自動索引更新をスケジュールしない場合は、代わりに `crontab` などのオペレーティング・システムの機能を使用できます。

#### **UPDATE MINIMUM** *minchanges*

索引が **UPDATE FREQUENCY** の設定に基づいて更新される前に、テキスト文書に必要な最小の変更の数。正の整数値のみが許可されます。デフォルト値は、DB2EXT.DBDEFAULTS ビューからとられます。ここで、DEFAULTNAME='UPDATEMINIMUM' です。

**DB2TEXT UPDATE** コマンドを手動で実行した場合は、この値は、無視されることに注意してください。増分更新の場合、ログ表およびトリガーがなければ変更の数を利用できないので、このオプションは、**RECREATE INDEX ON UPDATE** オプションと併用することはできません。

パーティション・データベース環境の場合、**UPDATE MINIMUM** は各パーティションでチェックされます。

#### **REORGANIZE AUTOMATIC | MANUAL**

**REORGANIZE AUTOMATIC** が指定されている場合は、更新頻度の設定に基づいて実行される更新のみが、索引を再編成します。このステップは、更新後、select REORGSUGGESTED from DB2EXT.TEXTINDEXES の値に従って自動的に完了します。

**REORGANIZE MANUAL** は、手動の **UPDATE** コマンド (**REORGANIZE** オプションを指定) と一緒になければ実行できません。

**REORGANIZE** 節が省略された場合、デフォルト値は、DB2EXT.DBDEFAULTS ビューからとられます。ここで、DEFAULTNAME='AUTOMATICREORG' です。

#### **REPLICATION CAPTURE TABLE** *capture-table-schema.capture-table-name* **CONTROL TABLE SCHEMA** *capture-control-schema*

増分更新処理の場合、それ以外の場合に索引に対して作成されるログ表またはテキスト保持ステージング表の代わりに、指定されたレプリケーション・キャプチャー表が取られます。したがって、schemaname、tablename、およびレプリケーション・キャプチャー表名は、ローカル DB2 (フェデレーテッド) データベースのオブジェクトに関係しています。

capture-control-schema は、レプリケーション・コントロール表のスキーマ名です (例: ローカル DB2 上での IBMSNAP\_PRUNE\_SET)。レプリケーション・コントロール表は、レプリケーションを設定した後、ローカル DB2 システム上のニックネームとして使用可能である必要があります。

少なくとも、以下のキャプチャー・コントロール表で使用可能なニックネームがある必要があります。

- IBMSNAP\_SIGNAL
- IBMSNAP\_PRUNE\_SET
- IBMSNAP\_PRUNCNTL
- IBMSNAP\_REGISTER
- IBMSNAP\_REG\_SYNC (DB2 データベース以外のリモート・ソースのみ)

DB2 レプリケーション・センターはリモート・キャプチャー表とキャプチャー・コントロール表のローカル・ニックネームを必ずしも自動的に作成するわけではないため、この作業を手動にすることができます。この作業は、テキスト索引が作成される表のニックネームを作成することに似ています。

ユーザー表ニックネームとキャプチャー表の主キー列の列名は一致している必要があります。また、キャプチャー表ニックネームの列 **IBMSNAP\_OPERATION**、**IBMSNAP\_COMMITSEQ**、および **IBMSNAP\_INTENTSEQ** の名前は変更しないようにしてください。

索引を作成した後、列名 **DB2EXT.TEXTINDEXES(LOGVIEWNAME)** と **DB2EXT.TEXTINDEXES(LOGVIEWSCHEMA)** は両方ともレプリケーション・キャプチャー表のローカル名を参照します。

Net Search Extender には DB2 レプリケーション・センターの機能のすべては必要ないので、変更データ表 (CD) または整合変更データ (CCD) 表は以下の規則に従っている必要があります。

- 変更キャプチャー登録を使用し、フル・リフレッシュ・コピー・オプションは使用しないでください。
- 変更のキャプチャーを水平サブセット化することは許可されていません。例えば、トリガーを使用した方法はその一例です。「DB2 レプリケーションのガイドおよびリファレンス パージョン 8」の第 6 章『レプリケーション環境におけるデータのサブセット化』を参照してください。
- 列のサブセットの変更の登録が許可されるのは、主キー列、テキスト列、および **DB2TEXT CREATE INDEX** コマンドの属性およびキャッシュ表式に含まれるすべての列が含まれている場合だけです。
- キャプチャー表には主キー列が含まれていなければなりません。変更後イメージが十分であることに注意してください。
- キャプチャー表は圧縮できません。各主キーごとに、最新データの入った 1 つの項目が必要です。しかしながら、Net Search Extender では、全履歴が使用可能でなければなりません。
- 表は D/I オプションを使用する必要があります。これにより、ソース表上の主キーへの更新事項をトランスフォームして、挿入/削除の対にすることができます。

DB2 V9.7 でサポートされているリモート・ソースのバージョンについては、技術サポートのページを参照してください。

#### 注:

正しいソース表名を登録表に挿入するようにしてください。リモート DBMS の種類によって、リモート表名を使用しなければならない場合と、ローカル・ニックネームを使用しなければならない場合があります。

- DB2: リモート表名 (リモート・サーバー上の表名)
- DB2 以外: ローカル・ニックネーム (フェデレーテッド DB2 データベース内の対応するニックネーム)

ローカル・ユーザーがニックネームを使用してリモート・データ・ソースにアクセスできるようにユーザー・マッピングがなければならず、リモート・ユーザーは表に対してコントロール特権を持っている必要があります。

DB2 インスタンス所有者ユーザー ID がローカル・ユーザー ID と異なる場合、DB2 インスタンス所有者ユーザー ID 用の追加ユーザー・マッピングが必要です。

指定された基本表名は、1 つのニックネームに対するビューであってはなりません。それは、ビューは複数のニックネームをカバーすることができ、複数の CD および CCD 表を含めることもできるからです。レプリケーション・キャプチャー節には CD または CCD 表を 1 つだけ指定することもできるので、ニックネームのビューはサポートされていません。また、リモート・ビューのニックネームは、主キーが欠落しているため、サポートできません。

CD または CCD 表はニックネームでなければならず、ビューまたは別名であってはなりません。

**create index** コマンドで **REPLICATION CAPTURE TABLE** オプションが使用されている場合、**COMMITCOUNT** オプションは使用できません。

#### **COMMITCOUNT FOR UPDATE** *count*

増分更新処理の場合、コミット・カウントを指定できます。指定されない場合は、デフォルト値は、DB2EXT.DBDEFAULTS ビューからとられます。ここで、DEFAULTNAME='COMMITCOUNT' です。

索引の **COMMITCOUNT FOR UPDATE** 値は、DB2EXT.TEXTINDEXES.COMMITCOUNT にあります。これは、**ALTER INDEX** コマンドを使用して、索引ごとに変更できます。また、**UPDATE FREQUENCY** 指定に従ってスケジュール済みの更新処理にも適用されます。0 の値は、更新が 1 つのトランザクションで完了したことを意味し、0 より大きい値は、1 つのトランザクションで処理する文書の数を指定します。ゼロ以外の **COMMITCOUNT** を使用しないことを推奨します。使用する必要がある場合は、単一の増分更新で実行される即時コミットの数を少数に抑えられるよう、十分大きい値を設定してください。各コミットでは、索引ファイルの移動および索引ログ・ファイルのクリーンアップが実行されますが、これらの処理は、特に繰り返し実行された場合、非常に大量の時間を要します。**COMMITCOUNT** は、ログ表および拡張テキスト保持ステージング・インフラストラクチャー (索引に対して構成されている場合) にリストされた文書に対するすべての変更に応用されます。

**COMMITCOUNT** を設定しない場合、db2ext.textindexes からの NUMBER\_DOCS パラメーターは更新されません。したがって、更新処理中に既に処理された文書の数を表示するには、CONTROL LIST コマンドを使用してください。

#### **RECREATE INDEX ON UPDATE**

これは、増分索引更新を許可せず、更新操作が実行される際に索引を再作成します (コマンドまたはスケジュールされた更新により)。

**注:** **RECREATE INDEX ON UPDATE** を指定した場合、次のようになります。

- ユーザー表にトリガーは作成されません。
- ログ表は作成されません。
- 拡張テキスト保持ステージング・インフラストラクチャーを索引に対して構成することはできません。

#### **INDEX DIRECTORY** *directory*

テキスト索引の保管先ディレクトリー・パス。ディレクトリーは索引データを含むので、ディレクトリーには、DB2 インスタンス所有者ユーザー ID 用の読み取り/書き込み、および実行権限があることを確認してください。

デフォルト値は、DB2EXT.DBDEFAULTS ビューからとられます。ここで、DEFAULTNAME=INDEXDIRECTORY' です。サブディレクトリー NODE<nr> が、サーバーの論理ノード上の索引を区別するために、ディレクトリーの下に作成されます。

パーティション・データベース環境では、このディレクトリーはすべての物理パーティション上に存在する必要があることに注意してください。

#### **WORK DIRECTORY** *directory*

別個の作業ディレクトリーをオプションで指定し、索引検索および管理操作時に一時ファイルの保管に使用することができます。ディレクトリーが存在し、DB2 インスタンス所有者ユーザー ID に対して読み取り/書き込みおよび実行権限を持っている必要があります。

デフォルト値は、DB2EXT.DBDEFAULTS ビューからとられます。ここで、DEFAULTNAME='WORKDIRECTORY' です。サブディレクトリー NODE<nr> が、サーバーの論理ノード上の索引を区別するために、ディレクトリーの下に作成されます。

パーティション・データベース環境では、このディレクトリーはすべての物理パーティション上に存在する必要があることに注意してください。

**WORK DIRECTORY** を指定しなかった場合は、**INDEX DIRECTORY** の下に **work** という名前のディレクトリーが作成されます。

**WORK DIRECTORY** を指定する場合、そのディレクトリーを索引ディレクトリーと同じ物理ファイル・システム上に置くことを強くお勧めします。この推奨事項に従わない場合、作業ディレクトリーからのファイルをリネームではなく索引ディレクトリーへ物理的にコピーする必要があるため、膨大な索引更新によるパフォーマンスの低下をもたらします。

**注:** **INDEX DIRECTORY** と **WORK DIRECTORY** に作成されるファイルは、インスタンス所有者に設定されている **umask** 制限に従うことになります。それらの **umask** 制限では、**fenced** ユーザーのグループ読み取り/書き込みアクセスを必ず許可するようにしてください。

#### **ADMINISTRATION TABLES IN** *table-space-name*

索引用に作成された管理表のための **REGULAR** 表スペースの名前。表スペースが存在する必要があります。指定されない場合は、索引が基本表上に作成されていれば、ユーザー表の表スペースが選択されます。

ニックネームまたはビューの場合は、デフォルトの表スペースが **DB2** によって選択されます。

パーティション・データベース環境でビュー、またはニックネームのテキスト索引を作成する場合や、ストアード・プロシージャ検索用のテキスト索引を作成する場合、管理表用の表スペースは、単一ノード上に指定されている必要があります、このノード上で明示的に呼び出される必要があります。

適切なノードと接続するためには、**DB2NODE** 環境変数を使用します。範囲パーティション表での索引の作成で、**ADMINISTRATION TABLES IN** 節は必須であるという点に注意してください。これを使用しないと、**CREATE INDEX** コマンドでエラーが返されます。エラー・メッセージの詳細については、**CTE0150E** を参照してください。



### CACHE TABLE (SQL-column-expression-list)

指定された列式からなるキャッシュ表が索引に追加して作成されます。このキャッシュは、フルテキスト検索の結果を DB2 表に結合しないで、ストアド・プロシージャ検索を介して戻すために使用されます。CONTAINS 関数をもったフルテキスト索引を使用する通常の DB2 検索は、常に可能です。

索引が作成される表の非修飾列名を使用して SQL 列式を定義します。許可される SQL 列式タイプは、すべて組み込み、およびユーザー定義特殊タイプです。結果セットの列名は、次の規則を使用して決定されます。

- 列式の SQL AS 節で明示的に指定された場合、指定された名前が使用される。例えば、以下のように指定します。

```
CACHE TABLE (C1+C2 AS myname)
```

- 指定された表の列が AS 節を指定せずに使用された場合、列の名前が使用される。例えば、以下のように指定します。

```
CACHE TABLE(C1)
```

- 式が AS を指定せずに使用され、名前付き列を参照しない場合は、**CREATE INDEX** はエラーを報告します。
- 列名の重複は許可されていません。

CLOB データ・タイプはキャッシュ・データ・タイプとしてサポートされていません。CLOB データ・タイプは VARCHARS にキャストする必要があります。

注: 結果セットの列名を分けて指定できない場合、CREATE INDEX コマンドはエラーを戻します。また、キャッシュ表は、作成後、暗黙的にはアクティブ化されません。例えばストアド・プロシージャによる検索は、DB2TEXT ACTIVATE CACHE が実行されるまでは、不可能です。

このオプションは、ユーザー表が単一パーティション表スペースに保管された場合にのみ、パーティション・データベース環境で使用できます。テキスト保持ステー징・インフラストラクチャーのデフォルト構成が、表に対して ON に設定され、その構成がテキスト索引に対して手動で無効にされていない場合、このオプションは使用できません。

### PERSISTENT IN directory

キャッシュも永続的に作成されるように指定します。永続キャッシュは非アクティブ化やシステム再始動の後で、非永続キャッシュよりも高速に再アクティブ化を実行できます。永続キャッシュは、指定されたディレクトリーに保管されます。

ディレクトリーが指定されていない場合、デフォルト値は db2ext.dbdefaults ビューからとられます。ここで、DEFAULTNAME='CACHEDIRECTORY' です。

### TEMPORARY

キャッシュが永続的に保管されないように指定します。PERSISTENT または TEMPORARY が指定されない場合、デフォルト値は、DB2EXT.DBDEFAULTS ビューからとられます。ここで、DEFAULTNAME='USEPERSISTENTCACHE' です。

#### MAXIMUM CACHE SIZE *memsize*

**DB2TEXT ACTIVATE CACHE** 時に作成されるキャッシュ済み表の最大サイズを指定します。*memsize* パラメーターは、メガバイト単位で正の整数として指定する必要があります。*memsize* にはデフォルト値がありません。整数が小さすぎると、**ACTIVATE CACHE** コマンドは失敗します。実際のキャッシュ・サイズは **ACTIVATE CACHE** コマンド時に計算されます。

異なる 32 ビット・プラットフォームごとの最大キャッシュ・サイズの制限は、以下のとおりです。

- Windows: 1024 MB (1 GB = 1073741824 バイト)
- Linux: 2048 MB (2 GB = 2147483647 バイト)

64 ビット・インストールでは、最大キャッシュ・サイズの制限は、使用可能なメモリーに依存します。

#### PCTFREE *percentage*

指定した割合のキャッシュを文書の追加のために、フリーのまま保持されるように指定します。割合は 0 以上 100 未満の整数値である必要があります。指定されない場合、デフォルト値は `db2ext.dbdefaults` ビューの `DEFAULTNAME='PCTFREE'` から取得されます。デフォルトは 50% です。

#### INITIAL SEARCH RESULT ORDER(*SQL-order-by-list*)

初期の索引付け時にユーザー表の内容を検索するために使用される順序を指定します。このオプションを使用して、フルテキスト検索結果の動的ランキングをスキップすると、キャッシュ結果表に保管されているとおりの、索引付けされた順序で文書が戻されます。

定義済みの検索結果順序付けを使用できるのは、ストアード・プロシージャ検索インターフェースを使用している場合のみです。例えば: `INITIAL RESULT ORDER(length(column1) asc, column2+column3 desc)`

定義済みの検索結果順序付けは、SQL スカラー検索関数および表値関数では実装されていません。

**注:** 増分更新後は、新規または変更された文書に対して、索引順序を保証できません。

#### KEY COLUMNS FOR INDEX ON VIEW (*SQL-columnname-list*)

ビュー上に索引が作成された場合は、**KEY COLUMNS FOR INDEX ON VIEW** 節を指定する必要があります。作成されない場合は、指定「しないで」ください。列名のリストは、ビュー内の行を「一意に」識別する列を指定します。

この一意性は、主キーの場合とは異なって DB2 によるチェックができないので、ユーザーが同等の一意性を保証する責任を持っています。指定された列は、索引用のログ表の PK01 列に記入するために使用されます。

#### INDEX CONFIGURATION (*option-value*), ...

索引構成値は次のとおりです。デフォルト値は、下線が引かれています。

オプション	値	説明
TreatNumbersAsWords	0 または 1	1 を設定した場合は、数字のシーケンスが文字に隣接している場合でも、この数字のシーケンスを別個のワードとして解釈する。デフォルトの 0 は、例えば tea42at5 が 1 つのワードと見なされることを意味します。
IndexStopWords	0 または 1	索引付け中にストップワードを無視するか、考慮する。デフォルトの 1 は、ストップワードを含むすべてのテキストを索引付けします。現在、ストップワード・リストは、ディレクトリー <instance>/sqllib/db2ext/resources の UCS-2 ファイル <language>.tsw にある。このファイルを変更しても、索引作成後は影響はない。また、<language> は、 <b>CREATE INDEX</b> コマンドからの <b>LANGUAGE</b> 値であることに注意。
UpdateDelay	秒数	キャプチャー表なしの増分更新の所要時間を秒単位で指定する。この所要時間を超過した古い項目だけが、ログ表から取り去られる。これは更新が失われることのないようにするためのものである。例えば、ユーザー・トランザクションが更新コマンドに干渉するというトランザクションのシナリオで、文書の変更が反映されずに終わることがないようにするためのものである。したがって、UpdateDelay パラメーターは、索引を作成した表に対して実行される、ユーザー書き込みトランザクションの最大所要時間に設定する必要がある。
IgnoreEmptyDocs	0 または 1	IgnoreEmptyDocs を 1 に設定した場合、空の文書 (長さが 0 または NULL 値の内容) は索引に表示されません。このオプションを使用した場合、内容が NULL (空) の文書は、次の増分更新によって索引から削除されます。
AuxLog	ON または OFF	トリガーによって認識されない変更を取り込む追加のログ・インフラストラクチャーの作成を制御します。範囲パーティション表のデフォルト設定は ON で、それ以外は OFF です。非範囲パーティション表に対する AuxLogNorm および範囲パーティション表に対する AuxLogPart の設定について、デフォルト表にあるデフォルト値を変更することができます。

#### CONNECT TO *database-name*

このコマンドのターゲットであるデータベースの名前。DB2DBDFT が設定されており、ユーザーがサーバーでコマンドを実行している場合は、このパラメーターを省略できます。ユーザー ID には必須 DB2 許可がなければならないことに注意してください。

#### USER *userid* USING *password*

*userid* および *password* を使用してデータベースに接続します。指定しない場合は、パスワードなしで現行ユーザー ID から接続が試行されます。

#### データベースへの変更



- Net Search Extender カタログ・ビューを変更します。
- 指定された表スペースに索引ログ表を作成します。これは、**RECREATE INDEX** オプションが指定されておらず、キャプチャー表も指定されていません。
- 指定された表スペースに索引イベント表を作成します。
- 最初の更新の実行時まで据え置き: ユーザー・テキスト表上にトリガーを作成します (**RECREATE INDEX** が指定されておらず、キャプチャー表も使用されていない場合のみ)。
- レプリケーション・キャプチャー表が使用されている場合、次の変更がキャプチャー・コントロール表に加えられます。
  - IBMSNAP\_PRUNCTNL および IBMSNAP\_PRUNE\_SET 表への挿入
 これらの表の項目は、以下の列によって固有に識別されます。
  - APPLY\_QUAL='NSE' || <DB2 instance running NSE>
  - SET\_NAME= <internal index identifier>
  - TARGET\_SERVER=<DB2 database name target to DB2TEXT operation>
- テキスト保持ステージング・インフラストラクチャーが索引に対して構成されている場合、文書の変更情報を取り込むためにアンカー表およびステージング表が作成されます。

#### 共有メモリーへの変更

ACTIVATE の実行時まで据え置き: CACHE TABLE 節が使用された場合、結果表のキャッシュは、共有メモリー に作成されます。

#### ファイル・システムへの変更

- サブディレクトリー NODE<nr> が、索引、作業、およびキャッシュ・ディレクトリーの下に作成されます。
- ディレクトリー *internal\_index\_name* が *indexdirectory/NODE<nr>* の下に作成されます。ここで、*indexdirectory* は、このコマンドの対応するパラメーターを指します。NODE<nr> は、パーティション・データベース環境のパーティション番号に関係付けられます。

#### 使用上の注意

フルテキスト索引の作成には、ユーザー表上の主キーが必要です。DB2 Net Search Extender バージョン 9.1 以降では、マルチコラム DB2 主キーをタイプの制限なしで使用できます。しかし、表値検索を使用する場合、コンパウンド主キーは許可されません。

主キー列の数は、62 個に制限され、すべての主キー列の全長は、ページ・サイズが 4K の表スペースでは 1007 バイト、ページ・サイズが 8K の表スペースでは 2031 バイト、ページ・サイズが 16K の表スペースでは 4079 バイト、ページ・サイズが 32K の表スペースでは 4096 バイトに制限されています。主キーが複数の列で構成されている場合、追加の列ごとに、ここで示した制限から 2 バイトずつ減少させる必要があります。

- ATTRIBUTES、CACHE TABLE および INITIAL SEARCH RESULT ORDER の SQL 式の合計サイズは、24 K バイトを超えてはなりません。

- 初期の索引更新は、常に 1 つの論理トランザクションとして実行されるので、この場合のコミット・カウントはありません。

Windows プラットフォームの場合、**modelfilepath** はローカル・ファイル・システム上になければなりません。Windows ネットワーク・マップ・ドライブはサポートされていません。

#### 注:

主キーが複数の列で構成されている場合は、主キーの定義に含まれている列の順序が **CREATE TABLE** ステートメントでの順序とまったく同じになるようにしてください。

索引の作成後は、主キー列またはビュー・キー列の長さを **ALTER TABLE** コマンドによって変更してはなりません。

ユーザー表、フルテキスト索引およびキャッシュ結果表の間の同期は、**update index** コマンド中に完了します。

---

## DEACTIVATE CACHE コマンド

キャッシュ表を解放します。永続キャッシュは、次の **ACTIVATE** コマンドで再利用するために保持されます。次のアクティブ化まで、ストアード・プロシージャを介した検索操作は、非活動キャッシュでは不可能です。

### 許可

ステートメントの許可 ID には、以下の特権が少なくとも 1 つ含まれていなければなりません。

- テキスト索引が定義されている表またはニックネームに対する **CONTROL** 特権
- **DBADM** 権限

### 前提条件

インスタンス所有者は、**DATAACCESS** を持つ **DBADM** 権限を保持している必要があります。バージョン 9.7 では、**SYSADM** は **SECADM** および **DBADM** 特権を保持しなくなりました。**DEACTIVATE CACHE** コマンドを実行する前に、**SECADM** はインスタンス所有者に **DATAACCESS** を持つ **DBADM** 権限を明示的に付与しなければなりません。

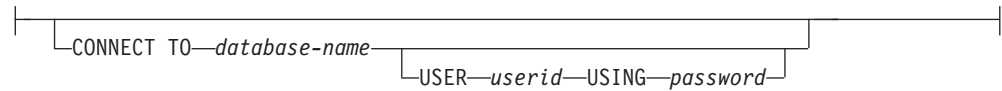
### コマンド構文

```

▶▶ DEACTIVATE CACHE FOR INDEX [index-schema-"."] index-name FOR TEXT ▶▶
▶ [connection-options] ▶▶

```

## connection-options:



## コマンド・パラメーター

### *index-schema*

**CREATE INDEX** コマンドで指定されたテキスト索引のスキーマ。スキーマが指定されていない場合は、DB2 接続のユーザー ID がスキーマ名として使用されます。

### *index-name*

**CREATE INDEX** コマンドで指定されたテキスト索引の名前。

### **CONNECT TO** *database-name*

このコマンドのターゲットであるデータベースの名前。**DB2DBDFT** が設定されており、ユーザーがサーバーでコマンドを実行している場合は、このパラメーターを省略できます。ユーザー ID には必須 DB2 許可がなければならぬことに注意してください。

### **USER** *userid* **USING** *password*

*userid* および *password* を使用してデータベースに接続します。指定しない場合は、パスワードなしで現行ユーザー ID から接続が試行されます。

## 使用上の注意

次のコマンドのいずれかが索引上で実行中の場合には、このコマンドを発行できないことに注意してください。

- **ACTIVATE CACHE**
- **DEACTIVATE CACHE**
- **UPDATE INDEX**
- **ALTER INDEX**
- **DROP INDEX**
- **CLEAR EVENTS**

注: 永続キャッシュの非アクティブ化後、ストアード・プロシージャーによる検索に対して、キャッシュはアクセス不能になります。ただし、その間に更新が実行されない限り、高速に **ACTIVATE** することが可能です。

非アクティブ化中に更新が行われた場合、永続キャッシュは、**ACTIVATE CACHE** コマンドを使用して自動的に最初から再作成されます。

---

## DROP INDEX コマンド

テキスト列用のフルテキスト索引をドロップする。索引のキャッシュがアクティブ化されている場合は、このコマンドを使用して削除されます。

## 許可

ステートメントの許可 ID には、以下の特権が少なくとも 1 つ含まれていなければなりません。

- テキスト索引が定義されている表またはニックネームに対する CONTROL 特権
- DBADM 権限

## 前提条件

インスタンス所有者は、DATAACCESS を持つ DBADM 権限を保持している必要があります。バージョン 9.7 では、SYSADM は SECADM および DBADM 特権を保持しなくなりました。DROP INDEX コマンドを実行する前に、SECADM はインスタンス所有者に DATAACCESS を持つ DBADM 権限を明示的に付与しなければなりません。

## コマンド構文

```
➤➤ DROP INDEX index-schema-".." index-name FOR TEXT ➤➤
|-----|
|connection-options|
```

### connection-options:

```
|-----|
|CONNECT-TO database-name |
|-----|
|USER userid USING password|
```

## コマンド・パラメーター

### *index-schema*

**CREATE INDEX** コマンドで指定されたテキスト索引のスキーマ。スキーマが指定されていない場合は、DB2 接続のユーザー ID がスキーマ名として使用されます。

### *index-name*

**CREATE INDEX** コマンドで指定された索引の名前。索引スキーマとともにデータベースのフルテキスト索引を一意に識別します。

### **CONNECT TO** *database-name*

このコマンドのターゲットであるデータベースの名前。DB2DBDFT が設定されており、ユーザーがサーバーでコマンドを実行している場合は、このパラメーターを省略できます。ユーザー ID には必須 DB2 許可がなければならぬことに注意してください。

### **USER** *userid* **USING** *password*

*userid* および *password* を使用してデータベースに接続します。指定しない場合は、パスワードなしで現行ユーザー ID から接続が試行されます。

## 使用上の注意

キャッシュ表のアクティブ化状況に無関係に索引は削除されます。

次のコマンドのいずれかが索引上で実行中の場合は、コマンドを発行してはいけないことに注意してください。

- **UPDATE INDEX**
- **CLEAR EVENTS**
- **ALTER INDEX**
- **ACTIVATE CACHE**
- **DEACTIVATE CACHE**
- **DROP INDEX**

注: DB2 のユーザー表がドロップされる前か後に、索引を手動でドロップする必要があります。ドロップしない場合は、ディレクトリーが正しくクリーンアップされません。

#### データベースへの変更

- Net Search Extender カタログ・ビューを変更します。
- DB2 索引をドロップします。
- 索引ログ、ステージング表、およびイベント表をドロップします。
- ユーザー・テキスト表上のトリガーを削除します。

レプリケーション・キャプチャー表を使用する場合、  
IBMSNAP\_PRUNE\_SET および IBMSNAP\_PRUNCTRNL 表内の項目が除去されます。

#### 共有メモリーへの変更

キャッシュ表が削除されます。

#### ファイル・システムへの変更

- ディレクトリー *internal\_index\_name* が、索引およびドロップされた索引の作業ディレクトリーから削除されます。
- 索引の永続キャッシュを削除します。

---

## DB2EXTTH コマンド

この独立ユーティリティーは、シソーラス定義ファイルをコンパイルします。シソーラス・コンパイラーを実行した後で、検索指数構文のシソーラス関連フィーチャーを使用できます。

### 許可

なし。このコマンドは、必ずしも表所有者を制限しませんが、照会の状況においてのみ意味をなします。

### コマンド構文

```
▶▶—db2extth— [quiet] —ccsid—code page— -f—definition-file-name—▶▶
├──quiet──┘
├──-h──┘
├──-H──┘
├──-?──┘
└──-copyright──┘
```

## コマンド・パラメーター

### **-f** *definition-file-name*

シソーラス定義を含むファイルの名前。ファイル名には、絶対パスまたはファイルへの相対パスが含まれる必要があります。ファイル名は、8+3 文字に制限されており、拡張子はオプションです。

シソーラス・ディクショナリーは、定義ファイルと同じディレクトリーに作成され、同じ名前を持ちます。唯一の違いは、ディクショナリーには、次の拡張子がある点です。wdf、wdv、grf、grv、MEY、ROS、NEY、SOS、および *lkn*。ただし、*n* は数字。既存のシソーラス・ファイルが同じ名前を持っていた場合、上書きされることに注意してください。

### **-ccsid** *code page*

シソーラス定義ファイルが書き込まれるコード・ページ。

**-quiet** 出力情報は、表示されません。

### **-copyright**

製品の内部ビルド番号を戻します。問題を報告する場合は、この番号を使用します。

### **-h | -H | -?**

ヘルプ情報を表示します。

## 使用上の注意

このコマンドを使用して、シソーラス定義ファイルをバイナリー・シソーラス定義フォーマットにコンパイルします。

---

## RESET PENDING コマンド

拡張テキスト保持ステージング・インフラストラクチャーを使用する場合、特定のコマンドが原因でステージング表が保留モードになることがあります。保留モードは、他のデータベースやテキスト検索操作をブロックします。**RESET PENDING** コマンドを使用すると、特定の表と関連したすべてのテキスト保持ステージング表に保全性の設定を実行することができます。**SET INTEGRITY** コマンドを各表に対して実行するために、すべてのテキスト索引および関連するステージング表を検索する必要はありません。

データ・パーティションをデタッチした後に、ステージング表の内容を更新するため、**RESET PENDING** コマンドを実行する必要があります。

### 許可

表に対する **CONTROL** 特権が必要です。

**SECADM** が、基本表に関連したテキスト保持ステージング表へのアクセス権を変更した場合でも、それらのテキスト保持ステージング表に対して保全性の設定ステートメントを実行する特権は必要です。

### 必須接続

このコマンドは **DB2** データベース・サーバーから発行する必要があります。

## コマンド構文

```
▶▶—RESET PENDING FOR TABLE table-schema.table-name FOR TEXT—————▶▶
|—————|
| |connection-options| |
```

### Connection-options:

```
|—————|
| CONNECT TO—database-name—————|
|—————|
| USER—userid—USING—password—|
```

## コマンド・パラメーター

### Table-name

テキスト保持ステージング・インフラストラクチャーが追加され、健全性処理を必要とする表の名前。

### Table-schema

従属表が保留モードになる原因となるコマンドを実行する対象の表のスキーマ。

## 使用上の注意

従属表が保留モードになる原因となるコマンド (**INSERT** パラメーターを指定した **LOAD** コマンドなど)、または従属表をリフレッシュするための **SET INTEGRITY** ステートメントが必要になるコマンド (**ALTER TABLE ... DETACH** など) を実行した後、**RESET PENDING** コマンドを使用してください。

---

## UPDATE INDEX コマンド

実際の索引更新が開始される前に、更新操作に必要なディスク・スペースを確認します。問題がなければ索引の更新処理が続行されます。

### 目的

単一の更新処理に必要なディスク・スペースが計算され、必要と見積もられたスペースがない場合は、索引の更新が終了します。57 ページの『第 7 部 計画に関する考慮事項』を参照してください。

索引付け処理を開始し、索引に関連付けられたテキスト列の現在の内容を反映し、索引を最新の状態に更新します。

更新の実行中も、**CONTAINS** 述部を使用した検索が可能です。アクティブ化されたキャッシュ結果表を持つ索引の場合、更新時にストアード・プロシージャによる検索も可能です。ただし、変更されたテキストがフルテキスト索引にまだコミットされていないにもかかわらず、キャッシュ表の列が新しい値を表示する可能性があります。

**CREATE INDEX** コマンドで **RECREATE INDEX ON UPDATE** オプションを使用すると、再作成前に索引がクリアされます。更新が完了するまで、空の結果が戻されます。

## 許可

ステートメントの許可 ID には、以下の特権が少なくとも 1 つ含まれていなければなりません。

- テキスト索引が定義されている表またはニックネームに対する CONTROL 特権
- DATAACCESS 権限

## 前提条件

インスタンス所有者は、DATAACCESS を持つ DBADM 権限を保持している必要があります。バージョン 9.7 では、SYSADM は SECADM および DBADM 特権を保持しなくなりました。UPDATE INDEX コマンドを実行する前に、SECADM はインスタンス所有者へ DATAACCESS を持つ DBADM 権限を明示的に付与しなければなりません。

## コマンド構文

```
▶▶ UPDATE INDEX index-schema."index-name FOR TEXT REORGANIZE
▶ COMMITCOUNT count USING UPDATE MINIMUM SKIPDISKSPACECHECK
▶ connection-options
```

### connection-options:

```
| CONNECT TO database-name USER userid USING password |
```

## コマンド・パラメーター

### *index-schema*

テキスト索引のスキーマ。これは **CREATE INDEX** コマンドで指定されます。スキーマが指定されていない場合は、DB2 接続のユーザー ID が使用されます。

### *index-name*

テキスト索引の名前。これは **CREATE INDEX** コマンドで指定されます。

### REORGANIZE

テキスト列が頻繁に更新されると、索引へのその後の更新が非効率になる可能性があります。更新処理を再び効率的にするには、索引を再編成します。索引が再編成を必要としているかどうかを判別するには、DB2EXT.TEXTINDEXES ビューを使用します。

手動による索引のチェックおよび再編成を回避するには、**CREATE INDEX** コマンドの **REORGANIZE AUTOMATIC** オプションを使用します。

注: 再編成処理は、通常の更新の後に行われます。



## USING UPDATE MINIMUM

指定した変更の数に達した場合に限り、**CREATE INDEX** コマンドから **UPDATE MINIMUM** 設定値を使用して、増分更新を開始します。テキスト保持ステージング・インフラストラクチャーが構成されている場合、合計の変更数が考慮されます。デフォルトでは、無条件に更新を開始します。

分散データベースの場合、**UPDATE MINIMUM** は各パーティションでチェックされます。

## COMMITCOUNT *count*

0 以上の **INTEGER** 値は、増分索引更新のために、検索エンジンおよび DB2 により 1 つのトランザクションで処理される文書の数を決めます。この値は、**ALTER INDEX** コマンドを使用して変更できます。

ただし、**CREATE INDEX** コマンドの後の最初の更新、または **RECREATE INDEX ON UPDATE** オプションによる更新などの初期更新の場合、**COMMITCOUNT** を無視する論理トランザクションが 1 つだけ存在します。

索引付けのスループットに大きな悪影響を与えるため、ゼロ以外の **COMMITCOUNT** を使用することは推奨されません。

## SKIPDISKSPACECHECK

このオプションを指定すると、増分索引更新のためのディスク・スペースのチェックがスキップされます。デフォルトでは、ディスク・スペースは索引更新の操作ごとにチェックされます。

## CONNECT TO *database-name*

このコマンドのターゲットであるデータベースの名前。 **DB2DBDFT** が設定されており、ユーザーがサーバーでコマンドを実行している場合は、このパラメーターを省略できます。ユーザー ID には必須 DB2 許可がなければならぬことに注意してください。

## USER *userid* USING *password*

*userid* および *password* を使用してデータベースに接続します。指定しない場合は、パスワードなしで現行ユーザー ID から接続が試行されます。

## 使用上の注意

このコマンドは同期的に実行します。パーティション・データベース環境内のすべての必要な DB2 論理/物理パーティション上で更新処理を開始します。所要時間は、索引付けする文書数および既に索引付けされた文書数によって異なります。更新の状況は、索引ごとに作成されるビューを介して表示することができます。このビューの名前は、列 **EVENTVIEWNAME** の **DB2EXT.TEXTINDEXES** から検索できます。

処理済みのコミット文書の数を表示するには、2 つのオプションがあります。どのくらいの文書が索引にコミット済みであるかを判別するには、**DB2EXT.TEXTINDEXES (NUMBER\_DOCS)** ビューを使用します。変更の開始、コミット、および更新処理の完了に関する情報も、索引に関連したイベント・ビューを使用します。

索引更新がまだアクティブである間に、既に処理された文書の数を表示するには、**CONTROL LIST ALL LOCKS FOR INDEX** コマンドを使用します。

注: ビューには、接続したパーティションからの情報だけが表示されます。

複数の物理パーティションを使用する基本表に対して増分更新を行う場合は、各パーティションの時刻が同期している必要があります。時刻が同期していないと、更新が失われたり、まったく実行されなかったりする可能性があります。

次のコマンドのいずれかが索引上で実行中の場合には、このコマンドを発行できません。

- **CLEAR EVENTS**
- **ALTER INDEX**
- **DROP INDEX**
- **ACTIVATE CACHE**
- **DEACTIVATE CACHE**
- **UPDATE INDEX**

非アクティブ化された永続キャッシュ結果表を使用して索引を更新した後、永続キャッシュは削除され、次の **ACTIVATE CACHE** コマンドによって、データベースの内容に基づいて再作成されます。

ユーザーがこのコマンドに割り込んだ場合、更新機能に関する処理はすべて停止します。 **COMMITCOUNT** が増分更新で使用された場合は、一部の更新が既にコミットされていて索引内で参照でき、他の更新については新たな更新コマンドが必要な場合があります。索引更新処理を強制的に中断すると、索引が破損する可能性があります。

索引の自動更新を停止するには、更新サービスに使用されるパーティション上で索引更新コマンドを実行中の **DB2** インスタンス所有者処理を検索します。すべてのパーティション上でこの処理および更新処理を停止します。

注: コマンドは、すべてのパーティションでの索引作成および初期の索引更新のために 2 つの別個のフェーズで機能するため、**db2text drop index** コマンドを発行して、索引が一部しか利用できないことがないようにします。このコマンドが発行されない場合は、手動更新コマンドまたは更新頻度オプションによって起動される次の更新で、一貫性のある状態を確保するために、完全な再索引付けが実行されます。

#### データベースへの変更

- イベント表への挿入
- 索引ログ表、および索引に対するテキスト保持ステージング表 (構成されている場合) からの削除

レプリケーション・キャプチャー表を使用する場合、データベースに以下の変更が加えられます。

- 最初の更新が開始する前に、**IBMSNAP\_SIGNAL** 表にシグナルが追加される。
- 増分更新後に、**IBMSNAP\_PRUNE\_SET** の同期点が変更される。

## HELP コマンド

使用可能な **DB2TEXT** コマンドのリスト、または個別の **DB2TEXT** コマンドの構文を表示します。

### 許可

なし

### コマンド構文



### コマンド・パラメーター

#### HELP | ?

指定したコマンドまたは理由コードのヘルプを表示します。

#### *command*

**DB2TEXT** コマンドを示す最初のキーワードです。以下のとおりです。

- ENABLE
- DISABLE
- CREATE
- DROP
- ALTER
- UPDATE
- CLEAR
- START
- STOP
- CONTROL
- ACTIVATE
- DEACTIVATE
- RESET PENDING

#### *reasoncode*

CTE0192 のような Net Search Extender コマンドからの理由コード。

### 使用上の注意

最初のキーワード以外にもキーワードが指定された場合、残りのキーワードは無視され、識別されたコマンドの構文が表示されます。

? または **HELP** の後に *command* パラメーターが指定されない (または、パラメーターがまったく指定されない) 場合は、**DB2TEXT** は選択可能な **DB2TEXT** コマンド・パラメーターをすべてリストします。

---

## COPYRIGHT コマンド

Net Search Extender の製品および著作権の情報を表示します。

### 許可

なし

### コマンド構文

```
▶▶-----▶▶
└─COPYRIGHT─┘
└─LEVEL──┘
```

### コマンド・パラメーター

#### **COPYRIGHT | LEVEL**

製品のバージョンの著作権文、バージョン番号およびビルド情報を表示します。

---

## 第 58 章 UNIX における Net Search Extender インストールおよびアンインストール・コマンド・リファレンス

Net Search Extender を UNIX にインストールまたは UNIX からアンインストールしようとする場合は、これらの処理に使用する構文に関する知識を持っていることが大切です。

このセクションでは、UNIX 用の Net Search Extender インストール・コマンドおよびアンインストール・コマンドの構文について説明します。DB2 バージョン 9.1 以降の Net Search Extender 製品のインストール済みコピーを表示する `db2nse1s` コマンドについても説明します。

---

### db2nse\_install コマンド

新規バージョンの Net Search Extender を UNIX オペレーティング・システムにインストールします。

#### 許可

root ユーザー権限

#### コマンド構文

```
▶▶ db2nse_install package-file path [-s] [-f] [--v]
```

#### コマンド・パラメーター

##### *package-file*

Net Search Extender 製品を格納しているファイルの名前。

*path* Net Search Extender をインストールしようとしている DB2 データベースのパス。

**-s** サイレント・インストールです。検査が実行されて、/tmp ディレクトリーにログ・ファイルが書き込まれます。

**-f** 強制インストールです。検査は実行されません。

**-v** プログラム・バージョンを表示して終了します。

#### 使用上の注意

このコマンドは、パラメーターとして渡されたパッケージ・ファイル内の Net Search Extender をインストールします。追加のパラメーターを指定しなかった場合、インストール・プログラムは、Net Search Extender をインストールできる適切な DB2 データベース・システムのコピーがシステムに存在するかどうかを確認し

ます。すべての DB2 データベースのインストール・パスがリストされますが、使用可能なサブセットのみが選択できます。インストール・パスを選択するようにシステムからメッセージが表示されます。

サイレント・インストールを選択した場合は、ユーザー対話はありません。インストール処理の結果は、コマンド・シェルやシェル・スクリプトも含む、呼び出し側プログラムに戻りコードとして渡されます。戻りコードを正しく処理するかどうかはユーザーに任されています。インストールについて記述したログ・ファイルが、/tmp ディレクトリーに書き込まれます。

強制インストールを選択した場合は、ユーザー対話も、これ以降の検査も実行されません。インストール処理の結果は、コマンド・シェルやシェル・スクリプトも含む、呼び出し側プログラムに戻りコードとして渡されます。戻りコードを正しく処理するかどうかはユーザーに任されています。インストールについて記述したログ・ファイルが、/tmp ディレクトリーに書き込まれます。

パラメーター **-s** とパラメーター **-f** の両方を使用する場合は、*path* を渡す必要があります。これらのパラメーターについては、デフォルトのパスはありません。

**-v** パラメーターが渡された場合は、プログラムのバージョンのみを表示してプログラムが終了し、それ以降のアクションは行われません。

---

## db2nse\_deinstall コマンド

Net Search Extender を UNIX オペレーティング・システムから削除します。

**db2nse\_deinstall** コマンドは、Net Search Extender をインストールしてある各 DB2 コピーのインストール・サブディレクトリーに配置されています。

### 許可

root ユーザー権限

### コマンド構文

▶▶ db2nse\_deinstall  **-v** ▶▶

### コマンド・パラメーター

**-v** プログラム・バージョンを表示して終了します。

### 使用上の注意

このコマンドは、DB2 バージョン 9.1 以降の Net Search Extender を削除します。このコマンドは、このコマンドを発行したディレクトリーにある Net Search Extender のコピーを除去します。他の場所にインストールされているその他の Net Search Extender のコピーは除去されません。

## db2nse1s コマンド

DB2 バージョン 9.7 以降の Net Search Extender のインストール済みのコピーをすべてリストします。

**db2nse1s** コマンドは、`/usr/local/bin` ディレクトリーに配置されています。

### 許可

root ユーザー権限

### コマンド構文

```
▶▶ db2nse1s [-c] [-v] ▶▶
```

### コマンド・パラメーター

- c** インストール済みの Net Search Extender バージョンを、コロンで分離した単純で簡潔なリストで表示して終了します。
- v** プログラム・バージョン (9.7 など) を表示して、終了します。

### 使用上の注意

コマンドの結果は、表、または項目をコロンで分離した簡潔なリストのいずれかで表示されます。結果には、Net Search Extender インストールのパス、バージョン、およびフィックスパック番号が含まれます。

例えば、パラメーターなしで **db2nse1s** を呼び出した場合の出力は、次のようになります。

```
db2nse1s
```

Install path	Level	FP
/opt/ibm/db2/V9.7	9.7.0.0	0
/test/V9.7	9.7.0.0	0

**-c** パラメーターを使用した場合に返される出力は、各情報項目がコロンで区切られた簡潔なリストです。このタイプの出力はプログラム、またはシェル・スクリプトで処理しやすくなっています。例えば、以下のように指定します。

```
db2nse1s -c
```

```
#PATH:VRMF:FIXPACK
/opt/ibm/db2/V9.7:9.7.0.0:0
/test/V9.7:9.7.0.0:0
```





---

## 第 59 章 検索指数の構文

検索指数は、テキスト文書で用語を検索する際に指定する条件です。検索パラメーターと 1 つ以上の検索語から構成されています。

検索指数の例は、137 ページの『第 42 章 SQL 検索指数の指定』 および Net Search Extender samples ディレクトリーにある search という名前のファイルに記載されています。

検索指数を使用する SQL スカラー検索関数は、次のとおりです。

### CONTAINS

この関数は、特定のテキスト文書でテキストを検索するために検索指数を使用します。文書にテキストまたは検索指数に指定された関係が含まれている場合は、INTEGER 値 1 を返します。それ以外の場合は、0 を返します。

### NUMBEROFMATCHES

この関数は、テキスト文書内で検索するために検索指数を使用し、文書ごとに一致した結果の数を示す INTEGER 値を返します。

### SCORE

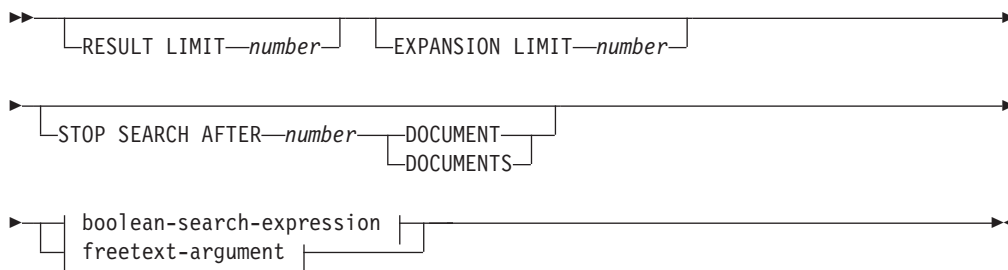
この関数は、テキスト文書内で検索するために検索指数を使用します。検出された文書ごとに値を返し、同じ索引での他の文書と比較して、検出された文書が検索指数によってどの程度適格に記述されているかを示します。

注: ストアード・プロシージャ検索および SQL 表値関数の検索指数には同じ構文を使用します。

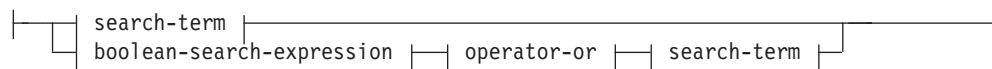
---

## 検索指数の構文

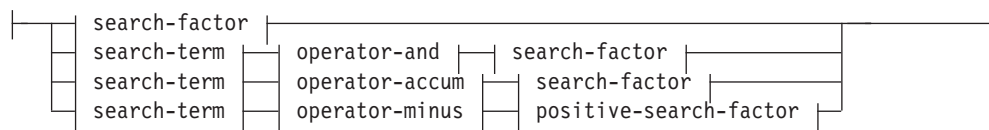
検索時にはさまざまな構文指数を使用できます。



### Boolean-search-expression:



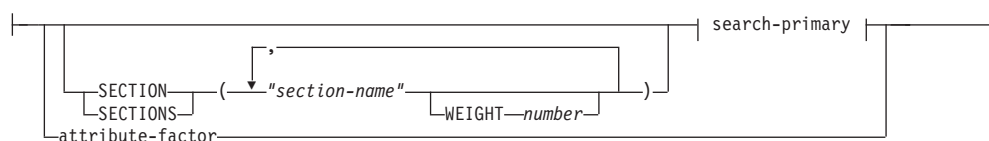
### search-term:



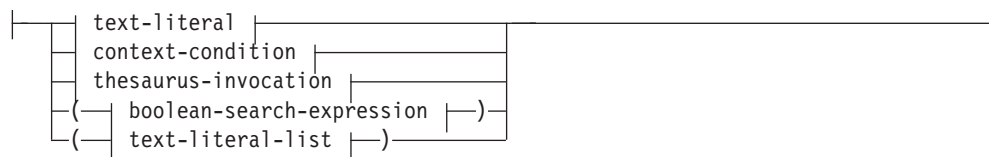
### Search-factor:



### Positive-search-factor:



### Search-primary:



### Operator-and:



### Operator-or:



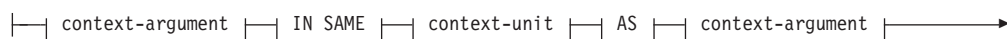
### Operator-accum:



### Operator-minus:



### Context-condition:





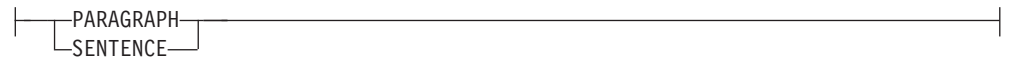
**Context-argument:**



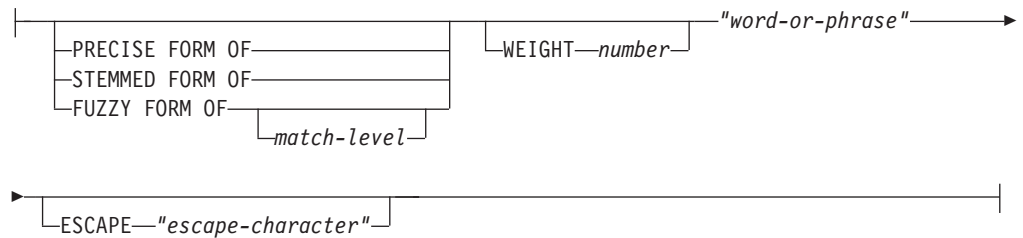
**Text-literal-list:**



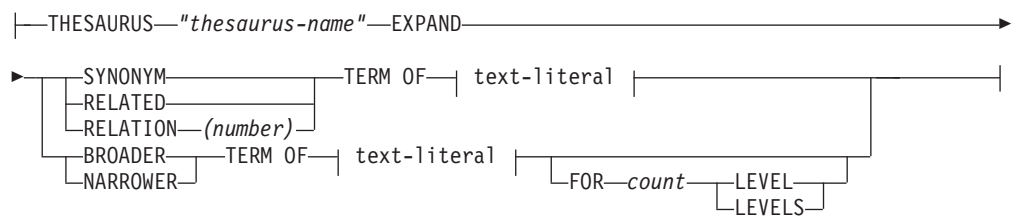
**Context-unit:**



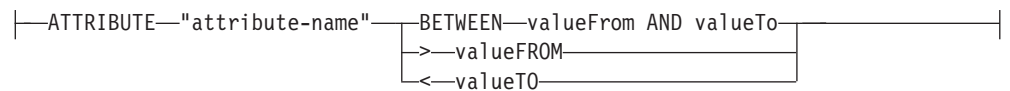
**Text-literal:**



**thesaurus-invocation:**



**Attribute-factor:**



## freetext-argument:

|—IS ABOUT—|——"word-or-phrase"—|——|  
|——language——| |——ESCAPE—"escape-character"—|

## 例

例は、137 ページの『第 42 章 SQL 検索引数の指定』に記載されています。

---

## 検索パラメーター

このトピックでは、検索時に使用する各種のパラメーターについて、パラメーターの説明も含めて示します。

### パラメーター

#### RESULT LIMIT *number*

フルテキスト検索によって戻される結果の最大数を指定するキーワード。

RESULT LIMIT は SCORE 関数とともに使用して、戻される結果にスコアが付けられ、最も一致する結果のみが処理されるようにします。

#### EXPANSION LIMIT *number*

検索のためにワイルドカード用語を拡張できるようにする用語の最大数を指定するキーワード。例えば、検索語 'a\*' を何回拡張するかを決める場合に使用します。索引が非常に大きく、多数のワイルドカード用語を使用している場合、より大きい結果セットを取得するには、このキーワードの値を調整する必要があります。拡張順序は、テキスト索引の内部編成に依存しています。あらかじめ決定しておくことはできません。ワイルドカード表現が一般すぎる、また「EXPANSION LIMIT」によって指定されているよりも多くの検索語に拡張される可能性がある場合、検索はエラーを戻し、この制限に達したための検索結果が切り捨てられたことを示します。

#### STOP SEARCH AFTER *number* DOCUMENT | DOCUMENTS

検索しきい値を指定するキーワード。検索した文書数が指定の値に達すると、検索は停止されて、中間結果が戻されます。値が低い場合は、検索のパフォーマンスが上がるものの、結果が少なくなり、潜在的にランクの高い文書が省略される可能性があります。

デフォルト値はなく、*number* 値は、正の整数でなければなりません。

#### boolean-search-expression

検索用語および検索要因は、構文図に従って、ブール演算子 NOT、AND、OR、ACCUM および MINUS を使用して、結合することができます。演算子には、次の優先順位があります (優先順位の高い順に): NOT > MINUS = ACCUM = AND > OR。これについて次の例を参照してください。

```
"Pilot" MINUS "passenger" & "vehicle" | "transport" & "public"
```

は、以下のように評価されます。

```
((("Pilot" MINUS "passenger") & ("vehicle"))) | ("transport" & "public")
```

ブール引数の 1 つが真であると評価した場合、演算子 ACCUM は真であると評価します (これは、OR 演算子に類似しています)。ランク値は、両方の

オペランドからランク値を累算して計算されます。ACCUM 演算子は、AND と同じバインディング (優先順位) を持っています。演算子 MINUS は、左のオペランドが真であると評価されると、真であると評価されます。ランク値は、右のオペランドが真であると評価されると、左のオペランドのランク値を取り、ペナルティーを減算することによって計算されます。

### search-primary

text-literal-list から成る search-primary は、文書 (の指定されたセクション) に text-literals が検出された場合に真と評価されます。thesaurus-invocation から成る search-primary は、文書 (の指定されたセクション) に拡張された text-literal が検出された場合に真と評価されます。

### SECTION | SECTIONS *section-name*

検索がその内部に制限される構造化文書で、1 つまたは複数のセクションを指定するキーワード。セクション名は、索引作成時に指定されたモデル・ファイルに指定するか、XPath 表記で表現されている必要があります。

セクション名には、大文字小文字の区別があります。モデル・ファイル内と照会内での、セクション名の大/小文字が一致するようにしてください。

このモデルは、識別可能なセクションを含む文書の構造を記述しているため、これらのセクションの内容を個別に検索できます。セクション名は、マスク文字を使用してマスクすることはできません。指定のセクションの 1 つで search-primary が検出された場合、SECTION 節を使用する *positive-search-factor* は真であると評価されます。

セクション名は、照会の実行時に評価される、有効な XPath 式ではありません。モデル・ファイルを使用しない場合は、デフォルトのセクション名が XPath 表記で記述されます。セクションを識別するための名前として、エレメントの絶対パス式 (/father/child/grandchild など) が使用されます。セクション名では、絶対 XPath 式はサポートされません。

### context-argument IN SAME context-unit AS context-argument AND context-argument ...

この条件を使用すれば、text-literal の組み合わせが同じ段落または文に出現する箇所を検索できます。コンテキスト引数は、つねに text-literal-list に同等であり、シソーラス拡張を使用して text-literal をそのようなリストに拡張することができます。

それぞれの拡張された context-argument の少なくとも 1 つの text-literal を含む context-unit (段落、または文) が文書にある場合、条件は真であると評価されます。これについて次の例を参照してください。

```
("a","b") IN SAME PARAGRAPH AS ("c","d")
AND THESAURUS "t1" EXPAND SYNONYM TERM OF "e".
```

e1、e2 が e の同義語であると想定すると、次の段落が一致します。

```
".. a c e .." , ".. a c e1.." , "a c e2..",
".. a d e .." , ".. a d e1.." , "a d e2..",
".. b c e .." , ".. b c e1.." , "b c e2..",
".. b d e .." , ".. b d e1.." , "b d e2..".
```

### PRECISE FORM OF

この **PRECISE FORM OF** の後に指定する語 (または句中のそれぞれの語) を、入力したとおりの形で検索するよう指定するキーワード。この形式の検索

は、大文字小文字を区別します。つまり、大文字および小文字の使用が有効です。例えば、mice を検索する場合、「Mouse」は検出されません。

このパラメーターを使用するには構成パラメーター `Respect case` に `yes` が設定されている必要があります。索引の作成後にこの構成設定を変更することはできません。

### STEMMED FORM OF

この **STEMMED FORM OF** の後に指定する語 (または句中のそれぞれの語) を、まずその語幹に縮めてから検索を実行するよう指定するキーワード。この検索形式では大文字小文字が区別されません。例えば、mouse を検索する場合、「Mouse」も検出されます。

語が語幹形式に縮められる方法は、言語によって異なります。現在、英語ステミングのみがサポートされており、語は、規則的な語尾変化に従う必要があります。

### FUZZY FORM OF

「ファジー」検索のためのキーワードです。検索項目とつづりの類似した語を検索します。これは、光学式文字認識 (OCR) プログラムで作成した文書を検索する場合に特に役に立ちます。そのような文書には、つづりの誤った語が含まれていることがあります。例えば、OCR プログラムは、economy という用語を economy と認識することがあるかもしれません。突き合わせ成功は、最初の 3 文字が一致する文書内の語の場合にのみ戻されることに注意してください。前述の例では、ecanomy は一致しません。検索要素内の語にマスク文字が含まれている場合、ファジー検索は使用できません。

#### *match-level*

類似性の度合いを指定する 1 から 100 までの整数。100 の方が 1 よりも類似性が高いことを示します。100 は「完全一致」を指定し、60 は、非常に「ファジーな値」と認識されます。突き合わせレベル (match level) がファジーなほど、より多くの文書が検索に対して適格となるため、検索に要する時間が長くなります。デフォルトの match level は 70 です。

### WEIGHT *number*

`text-literal` を重み値に関連付けてデフォルトのスコアを変更します。許可された重み値は、0 (最小スコア重み) から 1000 (最大スコア重み) の整数です。デフォルト値は 100 です。

#### *word-or-phrase*

検索する語または句。語の中で使用できる文字は、言語によって異なります。語と語の間を区切り文字で区切る必要があるかどうかは言語によって異なります。英語およびその他の大部分の言語の場合、句の中の語と語の間は空白文字で区切る必要があります。

二重引用符を含む文字ストリングを検索するには、二重引用符を二度入力します。例えば、テキスト "wildcard" 文字を検索するには、次のようになります。

```
""wildcard"" character"
```

この例では、1 組の引用符しか検索できないことに注意してください。1 つの文から 2 組の引用符は検索できません。また、各語または句の最大長は 128 バイトです。

## Masking characters

語に次のマスク文字を含めることができます。

### \_ (下線)

任意の 1 文字を表します。

### % (パーセント)

数が不定の任意の文字を表します。単一の % からなる語を指定した場合は、任意の長さのオプションの語を表します。1 つの単語をマスク文字だけにすることはできません。ただし、例外として単一の % でオプションの語を表すことは可能です。マスク文字を使用する場合は、THESAURUS キーワードを使用できません。マスク文字は、シソーラス照会部分では使用できません。組み合わせると、検索結果は予測不能になります。マスク文字は、英数字以外の文字の後に続けることはできません。マスク文字は常に単一の語にのみ拡張されるので、マスク文字は、ファジー検索内では使用できません。

## ESCAPE *escape-character*

次の文字を検索対象の文字として識別し、マスク文字として使用される文字としては識別しない文字。例えば、エスケープ文字が \$ であれば、\$%、\$\_、および \$\$ は、%、\_、および \$ を意味します。% および \_ 文字の前に \$ がいない場合は、マスク文字を表します。

検索時には、1 バイトのエスケープ文字のみを使用できます。2 バイト文字の使用は許可されていません。

## THESAURUS *thesaurus-name*

text-literal の拡張に使用されるシソーラスの名前を指定するために使用されるキーワード。シソーラス名は、シソーラス・コンパイラを使用してコンパイルされたシソーラスのファイル名 (拡張子なし) です。

<os-dependent>/sql1lib/db2ext/thes に配置されている必要があります。あるいは、絶対パスをファイル名の前に指定することもできます。

## EXPAND *relation*

どの関係がシソーラスを使用して text-literal の拡張に使用されるかを指定します。シソーラスには、DB2EXTTH コマンドで記述された事前定義の関係があります。これらは、次のキーワードを使用して参照されます。

- SYNONYM。同等を表す対称のリレーションシップ。
- RELATED。関連を表す対称のリレーションシップ。
- BROADER。指定された深さレベルを後に続けることができる、指定階層リレーションシップ。
- NARROWER。指定された深さレベルを後に続けることができる、指定階層リレーションシップ。

ユーザー定義の関係の場合、DB2EXTTH での関係定義に対応する RELATION(number) を使用します。

## TERM OF *text-literal*

シソーラスから他の検索項目が追加される text-literal。

## count LEVELS



指定された関係の検索項目を拡張するために使用される、シソーラスでの項目のレベルの数 (深さ) を指定するために使用するキーワード。このキーワードを指定しない場合は、カウント 1 が想定されます。深さの値は、正の整数でなければなりません。

#### **ATTRIBUTE** *attribute-name*

指定された条件に一致する属性を持つ文書を検索します。 *attribute-name* は、CREATE INDEX コマンドでの属性表現の名前、または文書モデル・ファイルの属性定義を指します。

属性係数は、ダブルのタイプの属性にのみ許可されます。値の精度は 15 桁が保証されています。16 桁以上の数は丸められます。マスク文字の使用は、*attribute-name*、*valueFrom* および *valueTo* では許可されません。詳細については、次の説明を参照してください。

#### **BETWEEN** *valueFrom* AND *valueTo*

属性の値が *valueFrom* より大きく (等しくない)、*valueTo* より小さい (等しくない) 場合に、**BETWEEN** 属性係数は真であると評価されます。

#### **>***valueFrom*

属性の値が *valueFrom* より大きい (等しくない) 場合に、"**>**" 属性係数は真であると評価されます。

#### **<***valueTo*

属性の値が *valueTo* より小さい (等しくない) 場合に、"**<**" 属性係数は真であると評価されます。

**CREATE INDEX** コマンドの属性名を引用符を付けて指定するかモデル・ファイルで定義すると、指定した属性名は正確に一致しなければなりません。これに対し CREATE INDEX コマンドで引用符を指定しない場合、属性名は英大文字でなければなりません。

#### **IS ABOUT** *language word-or-phrase*

フリー・テキスト検索指数を指定できるようにするオプション。**IS ABOUT** を使用すると、文書内に任意の順序で存在する、*word-or-phrase* で指定した任意の (必ずしもすべてではない) 語を検索できます。*word-or-phrase* で使用される用語がより近くに集まっていて、より多くの用語が 1 つの文書に含まれているほど、その文書に対してより高いスコアが戻されます。

パラメーター言語はオプションです。タイ語 (TH\_TH) の場合にトークン化の目的で必要とされるため、また、トルコ語 (TR\_TR) の場合に特有のケース・マッピングが必要とされるため、設定する必要があります。

**IS ABOUT** が有用なのは、文書のスコア値を要求した場合であり、検索結果がスコア値の順に配列される場合のみであることを注意してください。



## 第 60 章 SQL スカラー検索関数および SQL 表値関数

Net Search Extender には、DB2 に保管されているテキスト文書を検索する SQL スカラー検索関数および SQL 表値関数が備わっています。

このセクションでは、以下の SQL 検索関数について説明します。

検索関数	目的
『CONTAINS スカラー関数』	特定の文書のテキストを検索する。
266 ページの『NUMBEROFMATCHES スカラー関数』	検索して、検出された一致の数を返す。
266 ページの『SCORE スカラー関数』	検索して、検出されたテキスト文書のスコア値を返す。
266 ページの『DB2EXT.TEXTSEARCH コマンド』	SQL 表値関数は主キーの表、一致の数、およびスコア値を返す。
270 ページの『DB2EXT.HIGHLIGHT』	その文書が検索結果に該当した理由に関する情報の入手。

### CONTAINS スカラー関数

Net Search Extender によって索引付けされたテキスト文書を検索します。文書にテキストまたは検索引数に指定された関係が含まれている場合は、INTEGER 値 1 を返します。それ以外の場合は、0 を返します。

#### 関数の構文

►►—CONTAINS—(—*column-name*—,—*search-argument*—)————►►

#### 関数パラメーター

CONTAINS 関数パラメーター

*column name*

表列の名前。列には、関連したテキスト索引が必要です。テキスト索引は、管理コマンド **DB2TEXT CREATE INDEX** を使用して作成することができます。

*search-argument*

検索する用語を含む、タイプが VARCHAR のストリング。

注: CONTAINS 照会は、ビューに関して作成されたテキスト索引については使用できません。

---

## NUMBEROFMATCHES スカラー関数

テキスト文書内を検索し、文書ごとに一致した結果の数を示す INTEGER 値を返します。

### 関数の構文

▶▶—NUMBEROFMATCHES—(—column-name—,—search-argument—)————▶▶

### 関数パラメーター

*column name*

表列の名前。列には、関連したテキスト索引が必要です。テキスト索引は、管理コマンド **DB2TEXT CREATE INDEX** を使用して作成することができます。

*search-argument*

検索する用語を含む、タイプが VARCHAR のストリング。

注: NUMBEROFMATCHES 照会は、ビューに関して作成されたテキスト索引については使用できません。

---

## SCORE スカラー関数

テキスト文書内を検索して、検出された文書が検索回数でどの程度よく表されているかを示すスコア値を、検出された文書ごとに返すことができます。

SCORE は、DOUBLE 値を返します。検索語がより頻繁に文書で出現するほど、文書のスコアは増加します。

### 関数の構文

▶▶—SCORE—(—column-name—,—search-argument—)————▶▶

### 関数パラメーター

*column name*

列の名前。列には、関連したテキスト索引が必要です。テキスト索引は、管理コマンド **DB2TEXT CREATE INDEX** を使用して作成することができます。

*search-argument*

検索する用語を含む、タイプが VARCHAR のストリング。

注: SCORE 照会は、ビューに関して作成されたテキスト索引については使用できません。SCORE の戻す値。同じ索引から取得した他の値と比較する場合にのみ意味があります。

---

## DB2EXT.TEXTSEARCH コマンド

ストアード・プロシージャ検索および SQL スカラー検索関数に加えて、Net Search Extender は、ストアード・プロシージャに極めてよく似た 2 つの SQL 表値関数を備えています。

## 目的

それらの表値関数は両方とも `db2ext.textsearch` と呼ばれます。2つの違いは、その一方が `HIGHLIGHT` 関数をサポートし、2つの追加パラメーター `numberOfHits` と `hitInformation` を持っているということだけです。

どちらの表値関数も、ユーザー表から取得した結果を `CREATE INDEX` コマンドの `INITIAL SEARCH RESULT ORDER` パラメーターで定義されたソート基準に従ってソートして戻します。表値関数が含まれる SQL 照会ステートメントの終わりに結合 (`where T.primkey = S.key` など) が存在する場合は、結果行の順序は `CREATE INDEX` コマンドで定義した順序ではなく、結合メソッドに基づきます。

- コンパウンド主キーを持つ表では、表値関数を使用できません。
- この表値関数は、ユーザー表が単一パーティション表スペースに保管された場合にのみ、パーティション・データベース環境で使用できます。適切なノードに `DB2NODE` 環境変数を使用して接続していることを確認することも必要です。

### 1. HIGHLIGHT 関数サポートを利用しない場合の `db2ext.textsearch`

```
db2ext.textSearch
(
 query VARCHAR(4096),
 indexSchema VARCHAR(128),
 indexName VARCHAR(128),
 resultFirstRow INTEGER,
 resultNumberRows INTEGER,
 primKeyBinding <supported types>, // same type as primary key
)

return table
(
 primKey <supported types>, // same type as primary key
 numberOfMatches INTEGER,
 score DOUBLE,
 totalNbResults INTEGER
)
```

### 2. HIGHLIGHT 関数サポートを利用する場合の `db2ext.textsearch`

```
db2ext.textSearch
(
 query VARCHAR(4096),
 indexSchema VARCHAR(128),
 indexName VARCHAR(128),
 resultFirstRow INTEGER,
 resultNumberRows INTEGER,
 primKeyBinding <supported types>, // same type as primary key
 numberOfHits INTEGER
)

return table
(
 primKey <supported types>, // same type as primary key
 numberOfMatches INTEGER,
 score DOUBLE,
 totalNbResults INTEGER
 hitInformation BLOB(20K)
)
```

## 関数パラメーター

入力パラメーターを以下に示します。

**query** 詳細については、257 ページの『第 59 章 検索索引数の構文』を参照してください。

**indexSchema, indexName**

検索する索引を識別します。

**resultFirstRow**

照会の結果リストは、分けて戻されます。このパラメーターは、照会結果リストのどの行が、表値関数の結果表に最初書き込まれるかを記述します。値は 0 以上である必要があります。

番号 0 は、照会結果リストの最初の行を表わします。

**resultNumberRows**

このパラメーターは、照会結果リストのうちの何行が表値関数の結果表に書き込まれるかを記述します。0 は、すべての結果が戻されなければならないことを意味します。

このパラメーターは、照会結果リストの最大サイズを決定する結果限界照会パラメーターとは異なることに注意してください。

**primaryKeyBinding**

このパラメーターのタイプによって、**primaryKey** Output パラメーターのタイプが決まります。<type1> のタイプの主キーをもった基本表に対して、テキスト索引を作成した場合、primaryKeyBinding も、<type1> のタイプである必要があります。

さらに、このパラメーターによって、テキスト検索の範囲が決まります。

**primaryKeyBinding** を NULL ("CAST(NULL as <type1>)" に設定した場合、検索範囲は、索引に保管されているすべての文書になります。あるいは、**primaryKeyBinding** がバインドされている文書だけに検索を制限することができます。

例えば、**primaryKeyBinding** を CAST(5 as BIGINT) に設定する場合、検索を BIGINT 主キー値の「5」を持った単一の文書に制限します。

サポートされているのは、SMALLINT、INTEGER、BIGINT、REAL、DOUBLE、VARCHAR FOR BIT DATA、DATE、TIME、および TIMESTAMP のタイプの、単一の主キーのみであることに注意してください。

**numberOfHits**

このオプションは、強調表示関数 db2ext.highlight を使用して強調表示される用語の数を指定します。0 を指定した場合は、最大 1100 個までのヒットがすべて強調表示されます。この処理には時間がかかる可能性があります。

## 関数パラメーター

以下の戻り値は、さらに結果が要求された場合にユーザー表に結合する必要のある、一時表に保管されます。SELECT ステートメントで要求されている場合にのみ、NUMBEROFMATCHES、SCORE、TOTALNUMBEROFRESULTS、および HITINFORMATION が計算されることに注意してください。

**primKey**

検索された文書の主キー。

**numberofmatches**

NUMBEROFMATCHES は、各文書で一致した個数を示す INTEGER 値です。

**score** score は、DOUBLE 値を戻します。文書での検索語の頻度が増加すれば、文書スコアが増加します。

**totalNumberOfResults**

照会結果リストは、検索された結果の個数を示します。各行は同じ値を持つことに注意してください。

また、STOP SEARCH AFTER、または RESULT LIMIT を SCORE 構文とともに照会で使用する場合は、この数が既に信頼できなくなっていることに注意してください。

**hitInformation**

db2ext.textsearch から戻されるヒット情報は、強調表示処理に必要です。現在のところ、約 1100 ヒットのヒット情報がこの出力パラメーターに含まれています。ヒット数がこのしきい値を越えた場合は、超過した分のヒット情報が無視されます。

この値は numberOfHits を指定した場合にのみ戻されます。

**使用法**

SQL 表値関数を使用すれば、ストアード・プロシージャ検索と同じ方法でビュー上を検索できます。SQL 表値関数 とともに使用する場合のみ、共用メモリーは必要なく、索引にアクティブ化しなければならないキャッシュは必要ありません。

この関数は、主としてストアード・プロシージャ検索内で SQL 照会を使用してきたユーザー用です。ただし、基本表上の単一系列の主キーのみがサポートされているという制約事項があります。

次の例は、複数列の主キー表に対してどのように作業できるかを示しています。

```
select s.id from
db2ext.sample s, table (db2ext.textSearch(
 "characteristics",
 'DB2EXT',
 'COMMANDS',
 1,
 20,
 cast(NULL as INTEGER))) t
where s.id = t.primkey
```

この例では、まずこの表上で単一のユニーク・キーを使用してビューを作成し、次にこのビュー上で索引を作成する必要があります。

SQL 表値関数を db2ext.highlight 関数とともに使用する例については、270 ページの『DB2EXT.HIGHLIGHT』を参照してください。

## DB2EXT.HIGHLIGHT

db2ext.highlight 関数を使用して、特定の文書を検索結果に適格であると判断した理由を表示するために使用できる情報を取得します。

より具体的には、以下の作業を行うためにこの関数を使用できます。

- ヒットの入手
- ヒットおよび周囲のテキストの入手
- ヒットを周囲のユーザー定義の強調表示タグを含む文書の入手

db2ext.highlight 関数は db2ext.textsearch 表値関数と一緒にしか使用できないことに注意してください。この表値関数は索引を検索し、その結果を HIGHLIGHT 関数が使用します。

### 関数の構文

```
▶▶—db2ext.highlight—————▶▶
▶—(—document-content—,—hit-information—,—hit-processing-information—)—▶▶
```

### 関数パラメーター

入力パラメーターを以下に示します。

#### document content CLOB(100K)

TEXT またはシリアライズされた XML フォーマットの UTF8 文書のみがサポートされています。ネイティブな状態で格納された XML 文書を強調表示するには、XMLSERIALIZE SQL/XML 関数を使用して XML データを CLOB にシリアライズする必要があります。

CLOB の値を大きくするには、245 ページの『DB2EXTTH コマンド』を使用します。

#### hit information BLOB(20K)

ヒット情報を含むストリング。これは、numberOfHits パラメーターが指定されている場合に、db2ext.textsearch 関数によって戻されます。

#### hit processing information VARCHAR(1024)

このパラメーターはオプションと値の対のリストです。この対は、" " 文字で囲んだストリング文字をコンマ「,」文字で区切ったものです。指定した文書で強調表示を処理する方法を指定します。オプションを何も指定しないと、元の文書の内容が変更せずに戻されます。

#### TAGS = ("STRING", "STRING")

このオプションを使用すると、ユーザーは文書内のヒットの前後に挿入するタグを指定できます。このオプションを省略した場合は、文書内のヒットの前後にタグは追加されません。

#### WINDOW\_NUMBER = INTEGER

このオプションは、強調表示関数が文書の部分 (またはウィンドウ) をいくつ戻すようにするかを指定します。各ウィンドウには 1 つ以上のヒットが含まれています。各ウィンドウの最初のヒットによ

り、ユーザーに戻される文書の部分が決まります。これらのヒットは、周囲のテキストで囲まれている場合とそうでない場合があります。

このオプションを省略した場合、デフォルトとして 0 を取り、開始および終了タグ (指定されている場合) を含む文書全体が戻されます。このとき、WINDOW\_SIZE オプションは無視されます。

#### **WINDOW\_SIZE = INTEGER**

このオプションはウィンドウの推奨サイズをバイト数で指定します。この実際のサイズは、ヒット数、ヒットの長さ、および開始タグと終了タグのサイズによって異なります。このオプションを省略した場合、0 がデフォルトとなり、周囲のテキストなしでヒットだけが戻されます。

#### **WINDOW\_SEPARATOR = "STRING"**

このオプションは、1 つのウィンドウと次のウィンドウとを区切るために使用するタグを指定します。このオプションを省略した場合、"..." がデフォルト値になります。

#### **FORMAT = "STRING"**

このオプションは文書のフォーマットを指定します。有効な値は XML または TEXT です。このオプションを省略した場合、デフォルト値として TEXT を取ります。フォーマット値が、索引作成時に指定した値と同じになるようにしてください。

#### **MODEL\_NAME = "STRING"**

このオプションは、指定した XML 文書に関連するモデル名を指定します。FORMAT が TEXT の場合、このオプションを指定するとエラー条件になるので注意してください。

#### **SECTIONS = ("section-name1", ..., "section-nameN")**

XML 文書の場合、強調表示を関係のあるセクションに制限することができます。例えば、そのセクションをモデル・ファイルに定義できます。セクションを指定するには、1 つ以上のセクション名をコンマで区切ります。このオプションを省略した場合、XML 文書全体が強調表示されます。FORMAT が TEXT の場合は、このオプションは無視されるので注意してください。

DB2EXT.HIGHLIGHT で使用するセクション指定

("section-name1",..., "section-nameN") は、DB2EXT.TEXTSEARCH 関数の場合と同じでなければなりません。

## **関数パラメーター**

戻りパラメーターを以下に示します。

#### **CLOB(200K)**

HIGHLIGHT 関数は、関数自身による文書の変更部分を含む CLOB 値を戻します。

## **使用法**

次の例は、HIGHLIGHT 関数の使い方を示しています。



```

select p.id,
 p.title,
 db2ext.highlight(p.content,
 t.hitinformation,
 'TAGS = ("<bf>", "</bf>"),
 WINDOW_NUMBER = 5,
 WINDOW_SIZE = 200,
 WINDOW_SEPARATOR = "...",
 FORMAT = "XML",
 SECTIONS = ("section1-name", "section2-name"))')

FROM patent p, table (db2ext.textsearch(
 "relational database systems",
 'DB2EXT',
 'TI_FOR_CONTENT',
 0,
 20,
 CAST(NULL as BIGINT),
 15)) t

WHERE p.id = t.primkey

```

100 KB を超える文書を使用すると、SQL 照会が終了し、SQL エラーが発生します (SQL1476N と sql エラー -433)。これを回避するには、db2exthl コマンドを使用して、許可される文書内容のサイズを大きくしてください。

注: 特殊文字 (「改行」など) は現状のまま戻されます。

ネイティブな状態で格納された XML 文書を強調表示する場合は、これらの XML 文書を CLOB にシリアライゼーションした後、HIGHLIGHT 表値関数に渡します。次の例は、XMLSERIALIZE SQL/XML 関数を使用して、ネイティブな状態で格納された XML 文書に HIGHLIGHT 関数を使用する方法を示しています。次のサンプルの patent の内容は、ネイティブ XML として保管されています。FORMAT="XML" も指定されていることに注意してください。

```

select p.id,
 p.title,
 db2ext.highlight(XMLSERIALIZE(p.content AS CLOB(100K)),
 t.hitinformation,
 'TAGS = ("<bf>", "</bf>"),
 FORMAT = "XML",
 SECTIONS = ("section1-name", "section2-name"))')

FROM patent p, table (db2ext.textsearch(
 "xml database systems",
 'DB2EXT', 'TI_FOR_XML',
 0,
 20,
 CAST(NULL as BIGINT),
 15)) t

WHERE p.id = t.primkey

```

## 制約事項

- XML およびフラット・テキストの文書のみサポートされています。
- UTF8 のデータベースのみがサポートされています。バイナリー文書の場合は、文書が UTF8 でエンコードされていることを確認する必要があります。
- タイ語の文書はサポートされていません。
- 索引付けのときと照会のときとで使用した文書フォーマットが同じでない場合、HIGHLIGHT 関数から予測不能な結果が戻されます。これは、索引付け時にテキ



ストを取得する目的でトランスフォーメーション関数を使用するケースに特に当てはまります。索引付けの時刻と、出力内のテキスト・トークンの位置に影響を与える検索要求をサブミットする時刻との間で、トランスフォーメーション関数に何らかの変更を加えると、強調表示関数の結果は無効になります。

- 文書のテキスト部分で検出されたヒットだけが強調表示されます。
- 強調表示表値関数は `db2ext.textsearch` 関数と一緒にしか使用できません。
- ストリング値には " 文字を含めることはできません。



---

## 第 61 章 ストアード・プロシージャ検索関数

Net Search Extender は、事前定義の結果表を戻すストアード・プロシージャ検索を備えています。結果表は、索引の作成中に `CACHE TABLE` セクションで指定されます。

少ない数の結果を特定の順序で戻す場合には、ストアード・プロシージャ検索を使用します。

例えば、Web アプリケーションで、最もスコアがよい 20 行を戻し、残りの結果も 20 行ずつ増分して戻す場合があります。

注: ストアード・プロシージャ関数は、ユーザー表が単一パーティション表スペースに保管されている場合にのみ、パーティション・データベース環境で使用できません。

適切なパーティションに `DB2NODE` 環境変数を使用して接続していることを確認することも必要です。

---

### ストアード・プロシージャ検索 用の `DB2EXT.TEXTSEARCH`

ストアード・プロシージャによって戻された結果セットの列は、`DB2TEXT CREATE INDEX` コマンドの `CACHE TABLE` オプションにより表示されます。 `scoringFlag=1` の場合は、タイプがダブルの列が追加されます。

#### 関数の構文

```
db2ext.TextSearch(
 IN query VARCHAR(4096),
 IN indexSchema VARCHAR(128),
 IN indexName VARCHAR(128),
 IN resultFirstRow INTEGER,
 IN resultNumberRows INTEGER,
 IN scoringFlag INTEGER,
 IN searchTermCountsFlag INTEGER,
 OUT searchTermCounts VARCHAR(4096),
 OUT totalNumberOfResults INTEGER)
```

#### 関数パラメーター

入力パラメーターを以下に示します。

**Query** 詳しくは、257 ページの『第 59 章 検索指数の構文』を参照してください。

**indexSchema, indexName**

検索する索引の識別。

**resultFirstrow**

照会結果リストは、部分的に戻されます。パラメーターは、照会結果リスト

のどの行が、ストアード・プロシージャの結果セットに最初に書き込まれるかを記述します。照会結果リストの最初の行は、番号 0 で識別されます。

#### **resultNumberRows**

このパラメーターは、照会結果リストのうちの何行がストアード・プロシージャの結果セットに書き込まれるかを記述します。

これを、照会結果リストの最大サイズを判別する、照会の "result limit" 式と混同してはなりません。

値は 0 以上でなければなりません。0 の場合は、すべての結果を戻す必要があることを意味します。

**注:** 大きい結果セットを要求する場合は、一時ユーザー表スペースを使用可能にしてください。使用できるものがない場合は、表スペースを作成してください。以下の例は、UNIX オペレーティング・システム上で表スペースを作成しています。

```
db2 "create user temporary tablespace tempts managed by system
 using ('/work/tempts.ts')"
```

#### **scoringFlag**

0 は、スコアがないことを意味し、1 はスコアがあることを意味します。スコアリングが要求されている場合に、追加の列に値の高い順でスコア値が入れられて戻されます。

#### **searchTermCountsFlag**

このパラメーターは searchTermCounts 処理を制御します。

**searchTermCountsFlag** が 0 のときは、searchTermCounts は計算されません。

## **関数パラメーター**

出力パラメーターを以下に示します。

#### **searchTermCounts**

索引内の検索項目照会ごとの出現数。これらのカウントは、照会の検索項目と同じ順序でブランクで区切られたリストとして戻されます。

詳細については、**searchTermCountsFlag** を参照してください。

#### **totalNumberOfResults**

照会結果リストで検出された結果の合計数。

また、STOP SEARCH AFTER、または RESULT LIMIT を scoringFlag 構文とともに照会で使用する場合は、この数が既に信頼できなくなっていることに注意してください。

## **使用法**

ストアード・プロシージャによって戻された結果セットの列は、**DB2TEXT CREATE INDEX** コマンドの **CACHE TABLE** オプションにより表示されます。scoringFlag=1 の場合は、タイプがダブルの列が追加されます。この列には、SCORE 値が入っています。

最初の照会と同じストリングが指定されている 2 番目の照会のパフォーマンスを向上させるには、以下のオプションを使用します。ただし、totalNumberOfResults が不要な別のカーソル・ウィンドウで作業しなければならないことに注意してください。

- スコアリングが必要ない場合は、次の構文を追加します。STOP SEARCH AFTER  $x$  DOCUMENTS (ここで、 $x$  の値は resultFirstRow + resultNumberRows です)。
- スコアリングが必要である場合は、次の構文を追加します。STOP SEARCH AFTER  $y$  DOCUMENTS (ここで、 $y$  の値は最初の照会の totalNumberOfResults と同じです)。

検索のために適切なノードと接続するには、**DB2NODE** 環境変数を設定する必要があるかもしれません。

UNIX の場合には、次のコマンドを使用してください。

```
export DB2NODE=<no>
```

すべての物理ノードの時刻が同期していることが重要であることに注意してください。

Windows の場合には、次のコマンドを実行します。

```
set DB2NODE= <no>
```

<p><b>注:</b> インスタンス所有者 ID と異なる fenced ユーザー ID では、パーティション・データベースを扱う作業ができません。</p>
---------------------------------------------------------------------------------



---

## 第 62 章 Windows システム・エラー

以下に、Windows システム・エラーのリストを示します。

### システム・エラー

- 1 ファンクションが間違っています。
- 2 指定されたファイルが見つかりません。
- 3 指定されたパスが見つかりません。
- 4 ファイルを開くことができません。
- 5 アクセスが拒否されました。
- 6 ハンドルが無効です。
- 8 このコマンドを実行するのに十分な記憶域がありません。
- 14 この操作を完了するのに十分な記憶域がありません。
- 15 指定されたドライブが見つかりません。
- 29 指定されたデバイスに書き込めません。
- 30 指定されたデバイスから読み取れません。
- 32 プロセスはファイルにアクセスできません。別のプロセスが使用中です。
- 36 開かれている共有ファイルが多すぎます。
- 38 ファイルの終わりです。
- 39 ディスクがいっぱいです。
- 80 ファイルがあります。
- 82 ディレクトリまたはファイルを作成できません。
- 100 ほかのシステム セマフォを作成できません。
- 101 排他セマフォは、ほかのプロセスが所有しています。
- 102 セマフォが設定されています。閉じることができません。
- 103 セマフォを再設定できません。
- 104 割り込み時間には排他セマフォを要求できません。
- 105 このセマフォの以前の所有権は終了しました。
- 110 指定されたデバイスまたはファイルを開けません。
- 111 ファイル名が長すぎます。
- 112 ディスクに十分な空きスペースがありません。
- 121 セマフォがタイムアウトしました。
- 126 指定されたモジュールが見つかりません。
- 127 指定されたプロシージャが見つかりません。
- 147 このコマンドを処理するにはリソースが足りません。

- 155 ほかのスレッドを作成できません。
- 161 指定されたパスは無効です。
- 164 このシステムでは、これ以上のスレッドを作成できません。
- 170 要求されたリソースは使用中です。
- 183 既に存在するファイルを作成することはできません。
- 187 指定されたシステム セマフォ名が見つかりません。
- 206 ファイル名または拡張子が長すぎます。
- 267 ディレクトリ名が無効です。
- 288 呼び出し側が所有していない mutex を解放しようとしています。
- 298 1 つのセマフォに対するポストが多すぎます。
- 998 メモリ ロケーションへのアクセスが無効です。
- 1051 別の実行中のサービスが依存しているサービスにストップ コントロールが送信されました。
- 1052 要求された制御はこのサービスに対して無効です。
- 1053 そのサービスは指定時間内に開始要求または制御要求に応答しませんでした。
- 1054 サービスに対してスレッドを作成できませんでした。
- 1055 サービス データベースはロックされています。
- 1056 サービス インスタンスは既に実行されています。
- 1057 アカウント名が無効か、または存在しません。または、指定されたアカウント名に対してパスワードが無効です。
- 1058 指定されたサービスは無効であるか、または有効なデバイスが関連付けられていないため、開始できません。
- 1059 循環するサービス依存関係が指定されました。
- 1060 指定されたサービスはインストールされたサービスとして存在しません。
- 1061 そのサービスは現時点でコントロール メッセージを受け付けることができません。
- 1062 サービスが開始していません。
- 1063 サービス プロセスをサービス コントローラに接続できませんでした。
- 1064 制御要求を処理しているときに、サービスで例外が発生しました。
- 1066 そのサービスからサービス固有エラー コードが返されました。
- 1067 プロセスを途中で強制終了しました。
- 1068 依存関係サービスまたはグループを起動できませんでした。
- 1069 ログオンに失敗したため、サービスを開始できませんでした。
- 1070 サービスは開始後に開始待ち状態でハングしました。
- 1071 指定されたサービス データベース ロックは無効です。
- 1072 指定されたサービスは削除の対象としてマークされています。



- 1073 指定されたサービスは既に開始されています。
- 1078 その名前は、サービス名またはサービス表示名として既に使われています。
- 1079 このサービスに対して指定されたアカウントは、同じプロセス内で実行中のほかのサービスに対して指定されたアカウントと異なります。
- 1082 このサービスに構成されたりカバリー・プログラムが何もありません。
- 1154 このアプリケーションの実行に必要なライブラリ ファイルの 1 つが壊れています。
- 1219 入力内容が、既存の資格情報のセットと一致しませんでした。
- 1242 そのサービスは既に登録されています。
- 1243 指定されたサービスはありません。
- 1244 ユーザーが認証されていないため、要求された操作は実行されませんでした。
- 1245 ユーザーがネットワークにログオンしていないため、要求された操作は実行されませんでした。指定されたサービスはありません。
- 1392 ファイルまたはディレクトリが壊れているため、読み取ることができません。
- 1455 ページング ファイルが小さすぎるため、この操作を完了できません。
- 1793 ユーザーのアカウントは有効期限が切れています。



---

## 第 63 章 Net Search Extender インフォメーション・カタログ

Net Search Extender は、デフォルト設定、構成、テキスト索引、およびフォーマットに関する重要な情報をカタログ表に保管します。表上にいくつかのビューを照会して、この情報を表示することができます。

次のビューおよび表は、使用中のシステムの現行の構成を反映しています。

- データベース・レベル情報ビュー：
  - db2ext.dbdefaults
- 索引レベル情報ビュー：
  - db2ext.textindexes
  - db2ext.textindexformats
  - db2ext.indexconfiguration
- テキスト索引の表ビュー：
  - イベント・ビュー
  - ログ表ビュー
  - ステージング表ビュー

---

### データベース・レベル情報のビュー

ビュー db2ext.dbdefaults は、Net Search Extender データベースのデフォルト値をすべて表示します。

データベース・レベル上のデフォルト値は変更することができ、このビューで属性値のペアとして選択可能です。

db2ext.dbdefaults

```
db2 select DEFAULTNAME, DEFAULTVALUE from DB2EXT.DBDEFAULTS
```

表 3. db2ext.dbdefaults ビュー

属性	デフォルト値	注
CCSID	データベースの CCSID	文書のデフォルト CCSID。これは、CCSID が CREATE INDEX コマンドに指定されていない場合に適用されます。
FORMAT	TEXT	文書のデフォルト・フォーマット。これは、フォーマットが CREATE INDEX コマンドに指定されていない場合に適用されます。

表 3. db2ext.dbdefaults ビュー (続き)

属性	デフォルト値	注
INDEXDIRECTORY	注のパス名を参照	フルテキスト索引ファイル用のディレクトリー。これは、索引ディレクトリーが <b>CREATE INDEX</b> コマンドに指定されていない場合に適用されます。  Linux および UNIX オペレーティング・システムの場合、  <i>INSTHOME</i> /sql1lib/db2ext/indexes です ( <i>INSTHOME</i> はインスタンス所有者のホーム・ディレクトリー)。  Windows オペレーティング・システムの場合、  <i>INSTPROFDIR</i> ¥instance-name¥db2ext¥indexes です ( <i>INSTPROFDIR</i> はインスタンス・プロファイル・ディレクトリー)。
LANGUAGE	EN_US	文書の言語。
MODELCCSID	データベースの CCSID	文書モデル・ファイルの CCSID。
UPDATECOMMITCOUNT	0	更新中に 1 つのトランザクションで処理された変更の数。
CLEARCOMMITCOUNT	0	<b>CLEAR INDEX</b> コマンドの実行中に 1 つのトランザクションで処理された変更の数。
UPDATEFREQUENCY	NONE	新規索引での更新をチェックする場合。
UPDATEMINIMUM	1	更新が実行される前の変更の最小数。
WORKDIRECTORY	注のパス名を参照	索引一時ファイル用のディレクトリー。  Linux および UNIX オペレーティング・システムの場合、  <i>INSTHOME</i> /sql1lib/db2ext/indexes です ( <i>INSTHOME</i> はインスタンス所有者のホーム・ディレクトリー)。  Windows オペレーティング・システムの場合、  <i>INSTPROFDIR</i> ¥instance-name¥db2ext¥indexes です ( <i>INSTPROFDIR</i> はインスタンス・プロファイル・ディレクトリー)。

表 3. db2ext.dbdefaults ビュー (続き)

属性	デフォルト値	注
CACHEDIRECTORY	注のパス名を参照	<p><b>CREATE INDEX</b> コマンドの <b>PERSISTENT CACHE</b> オプションのデフォルト・ディレクトリー。</p> <p>Linux および UNIX オペレーティング・システムの場合、 <i>INSTHOME</i>/sql1lib/db2ext/indexes です (<i>INSTHOME</i> はインスタンス所有者のホーム・ディレクトリー)。</p> <p>Windows オペレーティング・システムの場合、 <i>INSTPROFDIR</i>¥instance-name¥db2ext¥indexes です (<i>INSTPROFDIR</i> はインスタンス・プロファイル・ディレクトリー)。</p>
PCTFREE	50	将来の挿入に備えて解放されたままのキャッシュのパーセント。
USERPERSISTENTCACHE	1	永続キャッシュを使用します。
AUTOMATICREORG	1	<b>CREATE INDEX</b> コマンドの <b>REORGANIZE</b> オプション。これは自動再編成を意味します。
TREATNUMBERSASWORDS	0	文字と数字のシーケンスを、隣接している場合でも、別個のワードとして解釈しません。例えば、デフォルトの 0 は、tea42at5 が 1 つのワードと見なされることを意味します。
INDEXSTOPWORDS	1	ストップワードを含むすべてのテキストを索引付けします。
VERSION		NSE V9.7 Net Search Extender の現行バージョン番号です。
UPDATEDELAY	0	キャプチャー表なしの増分更新の所要時間を秒単位で指定する。この所要時間を超過した古い項目だけが、ログ表から取り去られる。これは更新が失われることのないようにするためのものである。例えば、ユーザー・トランザクションが更新コマンドに干渉するというトランザクションのシナリオで、文書の変更が反映されずに終わることがないようにするためのものである。したがって、UpdateDelay パラメーターは、索引を作成した表に対して実行される、ユーザー書き込みトランザクションの最大所要時間に設定する必要がある。

表 3. db2ext.dbdefaults ビュー (続き)

属性	デフォルト値	注
AUXLOGNORM	OFF	非パーティション表に対しては、拡張テキスト保持ステー징・インフラストラクチャーをデフォルトで有効にしません。明示的な索引構成 AUXLOG ON でテキスト索引に対してステー징・インフラストラクチャーを有効にすることができます。
AUXLOGPART	ON	デフォルトで、範囲パーティション表に対して拡張テキスト保持ステー징・インフラストラクチャーを有効にします。明示的な索引構成 AUXLOG OFF でテキスト索引に対してステー징・インフラストラクチャーを無効にすることができます。
LOCKSCHEULERFILE	0	スケジューラー・ファイル ctedem.dat への同時書き込みアクセスをブロックするには、「1」に設定します。これを設定しない場合、並列処理の度合いが高い状態で作成、ドロップ、または更新頻度の変更をするシナリオでは、更新処理が不必要に開始される可能性があります。

注: Windows オペレーティング・システムで、デフォルトの索引ディレクトリーが変更されました。DB2 Net Search Extender バージョン 9.5 では、値は `DB2-installation-path%db2ext%indexes` でした。DB2 のインストール・パスについては、インストール済み DB2 データベース製品のディレクトリー構造 (Windows) を参照してください。

## 索引レベル情報のビュー

ビューを使用して索引レベルの情報を照会できます。

以下の Net Search Extender ビューを使用して、索引レベルで情報を照会できます。

- db2ext.textindexes
- db2ext.textindexformats
- db2ext.indexconfiguration
- <index eventview name schema>.<index eventview name>

旧バージョンとの互換性の理由から、DB2 テキスト情報エクステンダー・ビュー db2ext.textcolumns、db2ext.formats、および db2ext.models は引き続きサポートされていますが、使用すべきではありません。

db2ext.textcolumns ビューでは、OPERATION、OPERATIONBEGIN、および OPERATIONEND 列がサポートされなくなっていることに注意してください。

## db2ext.textindexes ビュー

Net Search Extender に対して使用可能な各データベースには、db2ext.textindexes ビューが含まれています。このビューには、このデータベースに作成されたテキスト索引の設定値、統計およびデフォルトに関する情報が含まれています。

テキスト索引を作成すると、db2ext.textindexes に新規項目が作成されます。作成したテキスト索引をドロップすると、これらの項目も削除されます。

索引に関する情報を得るためにビューを照会できます。次は、索引スキーマを使用した例です。

```
db2 "select COLNAME from DB2EXT.TEXTINDEXES where INDSHEMA='myschema'
and INDNAME='myindex'"
```

ただし、通常の SQL データ操作コマンドを使用してビューを変更したり、カタログ・ビューを明示的に作成またはドロップすることはできないことに注意してください。ビューの詳細な内容については、次の表を参照してください。

レプリケーション・パラメーターがこのビューに組み込まれていないことにも注意してください。

表 4. db2ext.textindexes ビュー

属性	タイプ	コメント
INDSCHEMA	VARCHAR(128)	テキスト索引のスキーマ名。
INDNAME	VARCHAR(128)	テキスト索引の名前。
TABSCHEMA	VARCHAR(128)	基本表、ニックネーム、およびビューのスキーマの表名。
TABNAME	VARCHAR(128)	索引が作成された別名。
COLNAME	VARCHAR(128)	索引が作成された列。
CCSID	INTEGER	この索引の文書 CCSID。
LANGUAGE	VARCHAR(5)	この索引の文書の言語。
FUNCTIONSCHEMA	VARCHAR(128)	列マッピング機能のスキーマ。
FUNCTIONNAME	VARCHAR(18)	列マッピング機能の名前。
INDEXDIRECTORY	VARCHAR(256)	フルテキスト索引ファイル用のディレクトリ。
WORKDIRECTORY	VARCHAR(256)	索引一時ファイル用のディレクトリ。
CACHEDIRECTORY	VARCHAR(256)	永続キャッシュのディレクトリ (persistentcache=1 の場合)。
UPDATEFREQUENCY	VARCHAR(300)	この索引に自動更新を適用するためのトリガー基準。
UPDATEMINIMUM	INTEGER	更新を実行する前に変更が必要な文書の最小数。
EVENTVIEWSHEMA	VARCHAR(128)	この索引用に作成されるイベント・ビューのスキーマ。
EVENTVIEWNAME	VARCHAR(128)	この索引用に作成されるイベント・ビューの名前。
LOGVIEWSHEMA	VARCHAR(128)	この索引用に作成されるログ・ビューのスキーマ。

表 4. db2ext.textindexes ビュー (続き)

属性	タイプ	コメント
LOGVIEWNAME	VARCHAR(128)	索引用に作成されるログ・ビューの名前 (ビュー上の増分更新にとって重要)。
COMMITCOUNT	INTEGER	コミット・カウント更新用のデフォルト。
NUMBER_DOCS	INTEGER	索引内の現在の文書数の合計。索引の更新中にこの値が更新されるのは、commitcount が設定されている場合だけであることを注意。
REORG_SUGGESTED	INTEGER	UPDATE INDEX REORGANIZE の実行によってパフォーマンス向上が可能かどうかを示す。このパラメーターが真 (1) になるのは、最低 1 つのノードで索引の再編成が提案されている場合だけである。
REORGAUTOMATIC	INTEGER	更新操作時に索引が自動的に再編成された場合は、1。
RECREATEONUPDATE	INTEGER	更新操作時に索引が自動的に再編成された場合は、1。
CREATIONTIME	TIMESTAMP	索引作成の時刻。
UPDATETIME	TIMESTAMP	最終の更新の時刻。UPDATE TIME が CREATION TIME と等しい場合は、更新は処理されていません。
PERSISTENTCACHE	INTEGER	永続キャッシュが使用される場合は、1。
MAXIMUMCACHESIZE	INTEGER	キャッシュの最大サイズ。
PCTFREE	INTEGER	将来の挿入に備えて解放されたままのキャッシュのパーセント。
CACHETABLE	VARCHAR(32000)	CACHE TABLE 用の列式リスト。
RESULTORDER	VARCHAR(32000)	INITIAL RESULT ORDER 用の SQL による順序付け。
ATTRIBUTES	VARCHAR(32000)	ATTRIBUTES 用の列式リスト。
VIEWKEYCOLUMNS	VARCHAR(32000)	ビュー上の索引のキー列。
AUXSTAGINGSHEMA	VARCHAR(16)	テキスト保持ステージング表用のスキーマ。SYSIBMTS に設定します。
AUXSTAGINGNAME	VARCHAR(48)	テキスト保持ステージング表のシステム生成名 (構成されている場合のみ)。

## db2ext.indexconfiguration ビュー

索引構成パラメーターは、db2ext.indexconfiguration ビューで使用できます。

ビューは、通常の SQL 照会機能を介して使用ができます。次は、索引名を使用した例です。

```
db2 "select VALUE from DB2EXT.INDEXCONFIGURATION where INDSHEMA='myschema'
and INDNAME='myindex' and PARAMETER ='INDEXSTOPWORDS'"
```



ビューの詳細な内容については、次の表を参照してください。

表 5. *db2ext.indexconfiguration* ビュー

属性	タイプ	コメント
INDSCHEMA	VARCHAR(128)	索引のスキーマ名。
INDNAME	VARCHAR(128)	索引の名前。
PARAMETER	VARCHAR(30)	パラメーターのタイプ。
VALUE	VARCHAR(512)	パラメーターの値。

PARAMETER 属性と VALUE 属性には、選択可能な値がいくつかあります。

表 6. *db2ext.indexconfiguration* ビュー

属性と値	属性と値
PARAMETER	VALUE
- TREATNUMBERASWORDS	- 0 または 1
- INDEXSTOPWORDS	- 0 または 1
- UPDATEDELAY	- 秒数 >= 0
AUXLOGPART	- ON または OFF
AUXLOGNORM	- ON または OFF

詳細については、**CREATE INDEX** コマンドの **CONFIGURATION** オプションを参照してください。

## db2ext.textindexformats ビュー

索引のフォーマットおよびモデル情報は、*db2ext.textindexformats* ビューで入手できます。

次は、索引名を使用した例です。

```
db2 "select FORMAT from DB2EXT.TEXTINDEXFORMATS where INDSCHEMA='myschema'
and INDNAME='myindex'"
```

ビューの詳細な内容については、次の表を参照してください。

表 7. *db2ext.textindexformats* ビュー

属性	タイプ	注
INDSCHEMA	VARCHAR(128)	索引のスキーマ名 (ログ表での索引名およびスキーマ名の接頭部として使用)。
INDNAME	VARCHAR(128)	CREATE INDEX コマンドで指定される索引名。
FORMAT	VARCHAR(30)	モデルはこのフォーマットにバインドされています。
MODELNAME	VARCHAR(30)	文書モデルの名前。
MODELFILE	VARCHAR(256)	モデル定義を含むファイル。
MODELCCSID	INTEGER	MODELFILE の CCSID。
DEFAULT	INTEGER	現行は 1。索引での複数のフォーマットは現在サポートされていないため。

## テキスト索引の表ビュー

イベント・ビューとログ表ビューを使用して、情報を索引レベルで照会できます。

以下の DB2 Net Search Extender ビューを使用して、索引レベルで情報を照会できます。

- イベント・ビュー
- ログ表ビュー

## イベント・ビュー

文書が見つからない、など索引付け中に問題が発生した場合、このビューを使用すれば、索引付けの状況およびエラー・イベントに関する情報を入手できます。これらの索引更新イベントは、その後、索引のイベント表に書き込まれます。

文書が見つからない、など索引付け中に問題が発生した場合、このビューを使用すれば、索引付けの状況およびエラー・イベントに関する情報を入手できます。これらの索引更新イベントは、その後、索引のイベント表に書き込まれます。

スキーマおよび名前は、db2ext.textindexes ビューに保管されます。イベント・ビューの名前を取得するには、次の例を使用します。

```
db2 "select EVENTVIEWSHEMA, EVENTVIEWNAME from DB2EXT.TEXTINDEXES
 where INDSHEMA = 'myschema' and INDNAME = 'myindex'
```

索引のイベント・ビューは、次の列から構成されています。

表8. イベント・ビュー

属性	タイプ	コメント
OPERATION	INTEGER	フルテキスト索引に反映されるユーザー表上の操作 (挿入 = 0/更新 = 1/削除 =2)。  レプリケーション・キャプチャー表を使用している場合は、更新操作が削除操作と挿入操作に分けられます。このとき、イベント表への挿入操作は、索引が作成されたソース表での挿入または更新操作のいずれかによるものである場合があります。
TIME	TIMESTAMP	イベント項目作成のタイム・スタンプ。
REASON	INTEGER	理由コード。理由コードのリストについては、293 ページの『第 64 章 テキスト検索エンジン理由コード』を参照してください。
SEVERITY	INTEGER	表の項目の重大度。例えば、1 は通知用、4 は警告、8 は表の項目のエラーを示す。
MESSAGE	VARCHAR(1024)	追加のテキスト情報。
KEY1, ... KEY14	ユーザー表に応じて異なる	ユーザー表の最初の主キー列から最後の主キー列 (最大 14)。
PARTITION	INTEGER	このエラーが発生したデータベース・パーティション番号。パーティション・データベース環境でない場合は 0。

イベントは、**DB2TEXT CLEAR EVENTS** コマンドを使用してクリアできます。

注: 更新処理の開始、コミット、および終了などの情報イベントは、このビューでも選択可能です。

この場合、Key1、... Key14 および OPERATION は、すべて NULL 値が指定されています。

ビュー上の索引の場合、PK01、...、PK14 列は、**CREATE INDEX** コマンドの **KEY COLUMNS** 節で指定された列に関係しています。

## ログ表、ビュー、およびニックネーム

ログ表の目的は、外部のフルテキスト索引との同期を必要とするユーザー表またはビュー上に変更操作を保管することにあります。

通常の表またはニックネーム表上で作成された索引の場合、変更情報をログ表に送るためにユーザー表上に作成されたトリガーがあります。ただし、レプリケーション・キャプチャー表を使用している場合、ログ表は作成されず、代わりにレプリケーション・キャプチャー表が使用されます。

ログ表の場合、更新コマンドは項目を読み取り、正常な同期の後、削除します。

ただし、ビューに関する索引の場合、トリガーはログ表に書きこむことができません。ビューは更新できるので、この作業はユーザーが行います。

表 9. ログ表ビュー

属性	タイプ	コメント
OPERATION	INTEGER	索引の同期が必要となるユーザー表の変更タイプ: (0 = 挿入、1 = 更新、2 = 削除)。
TIME	TIMESTAMP	この表の行の作成時のタイム・スタンプ。
PK01 ... PKnm	ユーザー表と同じ	エラーの場合は、問題が起きた列。これらは、ビュー上の索引の場合のユーザー表の主キー列または同等のキー列のコピーです。

表を作成したユーザーは、このビューを更新、挿入、および削除できます。

表 10. テキスト保持ステージング表

属性	タイプ	注
PK 1..n	ユーザー表と同じ	ユーザー表の主キー定義のコピー
Globaltransid	CHAR(8)	内部トランザクション ID
Globaltranstime	CHAR(13)	タイム・スタンプ
Operationtype	Integer	挿入 1 削除 -1

表の内容は、テキスト検索によって保守されますが、管理者が表の項目を削除することもあります。

索引の作成コマンドにレプリケーション・キャプチャー表を指定する場合、ログ表は作成されず、代わりにレプリケーション・キャプチャー表が使用されます。レプリケーション・キャプチャー表には、以下の列が含まれていることが必要です。

表 11. レプリケーション・キャプチャー表

属性	タイプ	注
IBMSNAP_OPERATION	INTEGER	索引の同期が必要となる CD または CCD 表の変更タイプ: (I = 挿入、U = 更新、D = 削除)。  レプリケーション・キャプチャー表を使用している場合は、更新操作が削除操作と挿入操作に分けられます。このとき、イベント表への挿入操作は、索引が作成されたソース表での挿入または更新操作のいずれかによるものである場合があります。
IBMSNAP_COMMITSEQ	CHAR	CD または CCD 表の対応する列へのマップ。
IBMSNAP_INTENTSEQ	CHAR	CD または CCD 表の対応する列へのマップ。
PK01 ... PKnm	ユーザー表と同じ	エラーの場合は、問題が起きた列。これらの列はユーザー表の主キー列です。

表を定義したユーザーは、 GRANT オプションを指定して更新、挿入、および削除できます。

---

## 第 64 章 テキスト検索エンジン理由コード

テキスト検索エンジンにより、エラー・コードとエラーの理由を生成できます。

- 0 操作が正常に実行されました - エラーは発生しませんでした。
- 1 関数に無効ハンドルが渡されました。
- 2 関数が十分なメモリーを割り振れませんでした。
- 3 アクセス制限またはセキュリティー制限のため、関数が実行されませんでした。
- 4 このバージョンのテキスト検索エンジン・ランタイムでは、この操作はサポートされていません。
- 5 この操作は現在、使用可能になっていません。
- 6 アプリケーションがテキスト検索エンジン・プロトコルに違反しました。正しくない順番でテキスト検索エンジン関数を呼び出しました。
- 7 予期しないエラーが発生しました。エラーをサービス担当者にお知らせください。
- 8 無効な言語が指定されました。
- 9 指定した言語は有効ですが、テキスト検索エンジン・ランタイムでサポートされていません。
- 10 無効な CCSID が指定されました。
- 11 指定した CCSID は有効ですが、テキスト検索エンジン・ランタイムでサポートされていません。
- 12 無効な文書 ID が指定されました。
- 13 指定した文書フォーマットは有効ですが、テキスト検索エンジン・ランタイムでサポートされていません。
- 14 無効な文書フォーマットが指定されました。
- 15 ファイル入出力時のアクセス制限のため、操作が成功しませんでした。
- 16 ファイル入出力時の読み取りエラーのため、操作が成功しませんでした。
- 17 ファイル入力時の読み取りエラーのため、操作が成功しませんでした。
- 18 ファイル出力時の書き込みエラーのため、操作が成功しませんでした。
- 19 ファイル入出力時のシーク・エラーのため、操作が成功しませんでした。
- 20 ファイル入出力時の tell エラーのため、操作が成功しませんでした。
- 21 ファイル入出力時のクローズ・エラーのため、操作が成功しませんでした。
- 22 名前変更操作時のエラーのため、操作が成功しませんでした。
- 23 除去操作時のエラーのため、操作が成功しませんでした。
- 24 mkdir 操作時のエラーのため、操作が成功しませんでした。

- 25 1 つ以上の関数引数が無効値をもっています (例: 予期しない NULL ポインターまたは無効な列挙型値)。
- 26 指定したディレクトリーが存在しません。
- 27 予期しないテキスト検索エンジン・エラーが発生しました。詳細については、エラー情報オブジェクトのテキスト検索エンジン・エラー・コードを参照してください。
- 28 予期しない COS エラーが発生しました。このエラーを報告してください。
- 29 空の文書を更新しようとしてしました。
- 30 この操作では、指定した引数はサポートされていません。
- 31 日付属性パーサーが、日付属性の解析中に無効値を検出しました。
- 32 数値属性パーサーが、数値属性の解析中に無効値を検出しました。
- 33 属性名が無効です。長過ぎる可能性があります。
- 35 この番号は将来のため予約されています。
- 36 入力文書に、属性の長さ制限を超える属性 (DATE、NUMBER、または STRING) が含まれています。属性テキストがその制限まで切り捨てられました。
- 38 ユーザーが設定した警告しきい値に到達しました。この結果、このエラーが生成されました。
- 39 入力文書に索引を作成できませんでした。含まれているネストされたフィールドが多過ぎます。
- 40 この索引で、1 つの属性タイプに対する異なる属性数の制限を超えました。
- 46 イテレーターのリストが空または削除されているため、イテレーターが有効ではありません (または有効でなくなりました)。
- 47 渡された種類のハンドルでは、この関数はサポートされていません。照会結果イテレーターでないリスト・イテレーターで `itlQueryResultEntryObtainData` を使用した場合などに、このエラーが発生します。
- 48 指定した言語およびリソース・パスでストップワード・ファイルが検出されない場合に、この警告が出されます。
- 49 ストップワード・ファイルにストップワードが含まれていない場合に、この警告が出ます。
- 50 ストップワード・ファイルに無効データが含まれている場合に、この警告が出ます。
- 100 指定した名前およびディレクトリーの索引が存在しないため、索引をオープンできませんでした。
- 101 指定した索引名は無効な索引名です。
- 102 指定した索引ディレクトリーは、無効なディレクトリー名です。
- 103 テキスト索引エンジンが、索引構造または索引ファイル・セットの破壊されたことを検知したため、操作を実行できません。

- 104 指定した名前およびディレクトリーの索引が既に存在するため、指定した索引を作成できません。
- 109 この索引では、ロールバック操作を実行してからでなければ、その他の操作を実行することはできません。
- 110 索引構成ファイルには、エラー・コンテキストで指定されているような必須セクションは含まれていません。
- 111 索引構成ファイルには、エラー・コンテキストで指定されているような必須オプションは含まれていません。
- 112 索引構成ファイルは、エラー・コンテキストで指定されているような必須オプションに無効データを含んでいます。
- 113 索引構成ファイルは、テキスト検索エンジンのバージョンと一致しません。
- 200 指定した文書モデル名は無効なモデル名です。
- 201 指定した文書モデル・フィールド名は無効なフィールド名です。
- 202 指定した文書モデルは定義されていません。
- 203 指定した文書モデルは既に存在するため、再定義することはできません。
- 204 索引に追加された文書モデルが、多過ぎるか大き過ぎます。
- 205 文書モデルに含まれる要素が多過ぎます。
- 206 文書モデル要素にこのタイプの要素には許可されていないパラメーター (XML 属性) が含まれています。
- 207 文書モデル要素にこのタイプのパラメーターには許可されていないパラメーター値 (XML 属性) が含まれています。
- 208 文書モデル要素に、「名前」などの、必要なパラメーター (XML 属性) が含まれていません。
- 209 文書モデルが XML と認識されないか、予期しない XML 要素で始まっています。
- 210 指定した XPath (ロケーター値) に予期しないトークンが含まれています。
- 211 指定した XPath (ロケーター値) に予期しない軸指定子 (2 つのコロンに続く名前) が含まれています。
- 212 指定した XPath (ロケーター値) に予期しないノード・テストが含まれています。
- 213 文書モデル・ディレクトリー・ファイル (拡張子 .mdx) が破壊されています。
- 214 文書モデル索引ファイル (拡張子 .mox) が破壊されています。
- 215 特定の文書属性にマップされ、さらに別の文書属性を含む XML 要素が文書に含まれています。内部属性は無視されます。
- 216 指定したパラメーター値が、GPP タグまたは HTML タグとして長過ぎます。
- 217 文書モデルに、重複したフィールド定義が含まれています。
- 218 文書モデルに、重複した属性定義が含まれています。



- 300 テキスト索引エンジンが、文書名マッピングに使用される索引ファイルで壊れた部分を検出したため、操作を実行できません。
- 301 テキスト索引エンジンが無効な文書番号を検出したため、操作を実行できません。
- 302 テキスト索引エンジンが無効な文書 ID を検出したため、操作を実行できません。
- 303 テキスト・サーチ・エンジンが文書 ID の索引項目を検出しなかったため、操作を実行できません。
- 304 テキスト索引エンジンが文書番号の索引項目を検出しなかったため、操作を実行できません。
- 305 テキスト索引エンジンが、使用された文書番号でオーバーフローを検出したため、操作を実行できません。
- 306 アプリケーションが索引の作成を試みた文書 ID が、文書リストに既に表示されています。テキスト検索エンジンは、1 つの索引作成順序 (つまり、更新がコミットされる以前の) での重複した文書 ID をサポートしていません。
- 340 用語の強さが無効です。
- 341 関係番号が無効です。in でなければなりません。
- 342 関係タイプが無効です。API に記述されたどれかの定義を使用してください。
- 343 句 (用語) が長過ぎます。
- 344 読み取り中に予期しないファイル終わりが検出されました。
- 345 索引/シソーラス・ファイルの読み取り中にバージョンの矛盾が検出されました。
- 346 シソーラス・バッファーでのオーバーフロー。
- 347 ファイルまたはディレクトリーの名前が無効です。名前が長過ぎる可能性があります。
- 348 検索でディクショナリー用語 (句) が検出されませんでした。あるいは、定義ファイルの項目に必須用語が含まれていません。
- 349 定義ファイルが空です。
- 350 入力パラメーターで指定されたシソーラス・ディクショナリーまたは定義ファイルが存在しません。
- 351 定義ファイルでの構文エラー。
- 352 リレーションシップが誤って指定されました。
- 352 リレーションシップ番号が範囲外です。
- 360 無効な単一文字マスキングが使用されました。
- 361 無効な複数文字マスキングが使用されました。
- 362 演算子 arity が、照会で指定されたオペランド数よりも小さい値です。
- 363 ItlEnOperator 列挙型で定義された範囲外の演算子値



- 364 列挙型範囲外のランク公式の値。
- 365 近接セグメントを定義する数値が範囲外です。
- 366 照会は準備中であり、再定義またはリセットできません。
- 367 以前の検索結果として与えられた有効範囲が空の結果を示します。
- 368 最初のフィールド名を設定する前にフィールド名の追加を要求している無効な呼び出し。
- 369 索引内容との無効な比較を要求する無効な検索フラグは無視されます。大文字小文字を区別しないで作成された索引に対して、大文字小文字を区別した比較を行うよう要求した場合などに、この理由コードがエラー情報に表示されます。
- 370 タイ語または DBCS 言語では、ストリングのマスキングはサポートされていません。
- 371 有効な照会入力がありません。例えば、検索項目が使用可能です。
- 372 無効な比較演算が要求されました。
- 373 無効な比較演算が要求されました。
- 374 空の索引に対して検索索引ハンドルが要求されました。
- 375 演算子と、要求された演算子モードとの組み合わせはサポートされていません。
- 380 検索結果は不完全です。しきい値のため、検索が中断されました。
- 381 索引検索によって、照会にストップワードが含まれていることが判明しました。
- 401 テキスト索引エンジンが、フィールド/属性の名前マッピングに使用される索引ファイルに壊れた部分を検出したため、操作を実行できません。
- 402 テキスト索引エンジンが無効なフィールド名または属性名を検出したため、操作を実行できません。
- 403 指定したフィールド名または属性名は定義されていないため、操作を実行できません。
- 404 この索引で、属性タイプのいずれかに対する異なる属性の制限、あるいは異なるフィールドに対する制限を超えました。
- 500 文書/データに無効な文字シーケンスが含まれています (UTF8 または UTF16 または DBCS のソース内で)。
- 501 コード・ページ・コンバーターにエラーが発生しました。
- 502 文書/データに不完全な文字シーケンスが含まれています (UTF8 または UTF16 または DBCS のソース内で)。
- 503 コード・ページ・コンバーターに無効な記述子があります。
- 600 XML 文書に非同期エンティティが含まれています。例えば、引用符で囲まれていない XML 属性値。
- 602 無効な文字参照 (or など)。
- 603 無効なバイナリー・エンティティ参照。

- 604 XML パーサー Expat を作成できませんでした。
- 605 タグの属性名は重複しない名前であればなりません。
- 607 XML パーサーは無効な外部エンティティー参照を検出しました。
- 608 文書には、< または > などの欠落した、誤ったトークンが含まれていません。
- 609 XML 文書は囲みタグをもっている必要があります。この囲み終了タグの後では、テキストは許可されていません。
- 610 この位置に処理命令を使用することはできません。例えば、最初の処理命令を `prolog <?xml .. ?>` にすることはできません。
- 611 要素は、開始タグ、内容、終了タグのシーケンスです。「<s> text /s>」などの場合、終了タグが誤っているため、このエラーが発生します。
- 612 XML パーサーでメモリー割り振りが失敗しました。
- 614 無効なパラメーター・エンティティー参照です。
- 615 不完全な文字です。2 バイト UTF8 文字の最初のバイトだけの可能性があります。
- 616 再帰的エンティティー参照です。
- 617 XML 構文エラー。例えば、囲み開始タグおよび終了タグの外にテキストがある場合など。
- 618 すべての開始タグには組み合わせの終了タグが必要です。
- 619 データ・セクションが閉じられていません。
- 620 トークンが閉じられていません。例えば、文書内の最後のトークンの後にテキストがある場合など。
- 621 解決できない文書がエンティティー内にあります。
- 622 予期しないエラーです。
- 631 メタ・タグ内のフィールド情報または属性情報を解析できませんでした。タグは `<meta name="abc" content="xyz">` のフォーマットにする必要があります。メタ・タグの属性名または内容が誤っている可能性があります。
- 632 エンティティーを文字にトランスフォームできませんでした。
- 650 異なるフィールド定義が同じ開始タグで始まっています。
- 651 開始タグに別の開始タグが含まれているため、タグがあいまいになっています。
- 652 フィールドおよび属性に同じ開始タグを使用する場合、それらには同じ終了タグを使用するか、両方ともに終了タグを使用しないようにする必要があります。
- 653 文書が終了していますが、フィールドが閉じられていません。
- 654 構造化フォーマットに対して文書モデルが指定されていません。文書は、フィールド情報または属性情報のないプレーン・テキストとして構文解析されます。

- 670** 「Outside In」(TM) ライブラリーが必要ですが、検出できなかったため、操作を実行できませんでした。
- 671** 「Outside In」(TM) ライブラリーから必要なプロシージャーをロードできなかったため、操作を実行できませんでした。ライブラリーは有効期限が切れているか、破壊されている可能性があります。
- 672** 「Outside In」で文書在处理中にエラーが発生しました。



---

## 第 14 部 トラブルシューティング



---

## 第 65 章 障害のトレース

IBM 担当員にエラーを知らせる必要がある場合には、エラーの原因を検出するために使用するファイルに情報を書き込めるようにするために、トレースをオンにするように依頼されることがあります。

### このタスクについて

トレース機能をオンにするとシステム・パフォーマンスに影響が出るため、トレース機能は、IBM サポート担当者またはお客様の技術サポート担当者から依頼があった場合にだけ使用してください。

### 手順

- トレースをオンにするには、次の DB2 機能を使用します。

```
db2trc on
```

- Net Search Extender にユニークな情報を受け取るには、次のようにコンポーネントのマスクに 96 を指定します。

```
db2trc on -m *.*.96.*.*
```

### 次のタスク

重大エラーの場合には、**db2diag** ログ・ファイルを参照すると役に立ちます。





---

## 第 66 章 正しい Net Search Extender コマンドを使用せずに DB2 オブジェクトをドロップする

---

### 表のドロップ

1 つ以上のテキスト索引を持つ表をドロップする前に、以下のコマンドを発行することが必要です。

#### このタスクについて

1 つ以上のテキスト索引を持つ表をドロップする前に、各テキスト索引に対して以下のコマンドを発行することが必要です。

```
db2text drop index index_name for text
```

誤って、索引をドロップする前に表をドロップしてしまうと、管理表やテキスト索引ファイルなど一部の索引が残ってしまいます。

これらのファイルを除去するには、表が既に存在していないとしても、**db2text drop index** コマンドを使用して索引をドロップします。

---

### データベースのドロップ

1 つ以上のテキスト索引を持つデータベースをドロップする前に、各テキスト索引に対して以下のコマンドを発行することが必要です。このコマンドを使用しない場合は、*index\_directory* と *index\_work\_directory* にあるすべての索引ファイルを手動で削除することが必要になります。

#### 手順

1 つ以上のテキスト索引を持つデータベースをドロップする前に、以下のようになります。

1. 各テキスト索引に対し、以下のコマンドを発行します。

```
db2text drop index index_name for text
```

このコマンドを使用しない場合は、*index\_directory* と *index\_work\_directory* にあるすべての索引ファイルを手動で削除することが必要になります。

2. ドロップしたデータベースに属する索引が自動更新中に作成された場合には、スケジューラー・ファイル *ctedem.dat* の編集が必要になります。

a. 次のように入力します。

- UNIX の場合:

```
db2text stop force
cd ~/sql1lib/db2ext
```

- Windows の場合:

```
db2text stop force
cd db2_install_path\sql1lib\%db2_instance_name\db2ext
```

- b. ディレクトリー内のファイル `ctedem.dat` を開き、ドロップしたデータベースを参照するすべての項目を削除します。

---

## 第 67 章 Windows でのインストール戻りコード

このトピックでは、Windows でのインストール戻りコードに関する情報を提供します。

### Windows での `setup.exe` の戻りコード

`setup.exe` プログラムは、`setup.log` ファイルに以下のコードを戻します。

- 0 成功
- -1 一般エラー
- -2 無効なモード
- -3 必須データが `setup.iss` ファイルから検出できません
- -4 使用可能なメモリーが不足しています
- -5 ファイルが存在しません
- -6 応答ファイルに書き込めません
- -7 ログ・ファイルに書き込めません
- -8 インストール・シールド・サイレント応答 (`.iss`) ファイルのパスが無効です
- -9 有効なリスト・タイプではありません
- -10 データ・タイプが無効です
- -11 セットアップ時に不明なエラーが発生しました
- -12 ダイアログ・ボックスに異常があります
- -51 指定のフォルダーを作成できません
- -52 指定のファイルまたはフォルダーにアクセスできません
- -53 無効なオプションを選択しました



---

## 第 68 章 ヒント

このトピックでは、許可、言語、db2cli.ini ファイルの変更、およびクライアント・サーバー・インターオペラビリティなどのトピックについての有益なヒントを示します。

**許可** Windows で **DB2TEXT START** コマンドを発行する場合は、必ず管理者グループのメンバーであることを確認してください。メンバーでない場合、**DB2TEXT START** コマンドは失敗して、次のメッセージを戻します。 CTE0218 関数 "OpenSCManager()" がエラー・コード "5" で失敗しました。

**許可** Windows の場合、Net Search Extender のインスタンス・サービス DB2EXT-DB2\_instance\_name が、システム・アカウントではなくユーザー・アカウントで実行されていることを確認してください。システム・アカウントで実行している場合は、データベースを使用可能にできません。

**言語** Net Search Extender のイベント・ログ・メッセージは、常に DB2 サーバーの言語で表示されるため、DB2 コマンド行プロセッサから発行されたコマンドのイベント・ログ・メッセージは、異なる言語で表示される場合があります。

### db2cli.ini ファイルを変更しています

db2cli.ini ファイルを変更した後で Net Search Extender の使用に問題が発生した場合は、元のバージョンの db2cli.ini をリストアしてください。

### クライアント・サーバー・インターオペラビリティ

同じレベルの Net Search Extender フィックスパックをサーバーとクライアントの両方にインストールする必要があります。

クライアントからサーバーおよびサーバーからクライアントへのインターオペラビリティは、サポートされているオペレーティング・システムでのみ実行できます。

### ログ・サイズ

使用可能な DB2 ログ・サイズを超えるスペースが必要であるというエラー・メッセージおよび警告メッセージが出力されるために索引付け処理が完了しない場合は、DB2 はトランザクション全体をロールバックし、ログ表の項目をコミットしません。これは、項目を参照できないということを意味します。

この状態を避けるためにトランザクション・ログ・サイズを大きくする方法については DB2 の資料を確認してください。

### DBCS オブジェクト名

**db2text** 管理コマンドで DBCS オブジェクト名を使用する場合は、それらの名前を二重引用符で囲んで、大文字にトランスフォームされないようにする必要があります。

### ニックネームにおける増分索引更新

2 つ以上の索引の初期更新が同時に開始された場合、更新コマンドが SQL0803N エラーを戻す場合があります。この場合は、更新コマンドを再度実行してみてください。

### 単一マスキングおよび文字正規化

über のような単語は正規化されて、正規化した形式 (ueber) で索引に格納されます。したがって、\_ber のように単一の文字マスキングを含んだ照会を発行した場合、über は検出されません。

### 重複するキャッシュ列名の使用

重複するキャッシュ列名を使用した場合、テキスト索引の作成時や索引更新時にエラーは出力されませんが、検索は実行できません。検索しようとする、重複する列が使用されたということを示す SQL エラー・メッセージが出力されます。

### 誤った共用メモリー・サイズ

**db2text activate cache** コマンドにおいて、指定の最大キャッシュ・サイズが小さすぎる場合、結果のエラー・メッセージに表示される必要なキャッシュ・サイズは誤っています。

DB2EXT.MAXIMUM\_CACHE\_SIZE 関数および DB2EXT.PCTFREE 関数を使用して正しいキャッシュ・サイズを確認してください。**db2text alter index** コマンドを使用して最大キャッシュ・サイズを訂正してから、キャッシュを再度アクティブ化してください。

### 非 Unicode データベースに存在する Unicode 表

データベースが Unicode をサポートしていない場合、Unicode 表にテキスト索引を作成することはできません。

### Linux での照会において LANG 変数のコード・ページが誤っています

照会の LANG 変数の設定に 7 ビット ASCII コード・ページを使用すると、「検索インデックスの処理上の問題です。」という内容のエラーが表示されます。

これを回避するためには、LANG 変数を 8 ビット LANG 値に変更して、DB2 を再始動してから、検索を再実行してみてください。

### ファイル・アクセスの問題

文書モデル、テキスト索引、またはシソーラスなどのファイルにアクセスできない場合は、正しいパスワードを使用していること、および Net Search Extender インスタンス・サービスを実行するための正しい権限があることを確認してください。これは、マップされたネットワーク・ドライブ上の共有リソースの場合に特に当てはまります。

### キャッシュを使用できません

キャッシュの検索やアクティブ化中に、次のエラー・メッセージが表示される場合があります。CTE0271 キャッシュは使用可能ではありません。

DEACTIVATE および ACTIVATE RECREATE が必要です。この問題を解決するためには、システム設定を確認して、ページング・スペースおよび空きメモリーの量を増やしてください。

### アンインストール後にインスタンス・サービスがドロップされていません

Net Search Extender をアンインストールしてもインスタンス・サービスが除去されていない場合は、`ctereg instancename unregister` ツールを使用して、サービスを手動でドロップしてください。例: `ctereg db2-0 unregister`

### UNIX における cteprcx の異常終了

使用したインスタンス所有者が、別々の fenced ユーザー ID を持たないこ

とを確認してください。これを確認するためには、`instance_home_dir/sql/lib/adm`にある `.fenced` ファイルを開いて、インスタンス所有者が `fenced` ユーザーでもあるかどうかを確認してください。

#### SQL0668N で索引の更新が失敗する

SQL0668N (rc=1) で索引の更新が失敗する場合、従属表に対して保全性処理を必要とする基本表でのデータベース操作が原因で、テキスト保持ステージング表へのアクセスがブロックされています。

**RESET PENDING** コマンドを使用してステージング表へのアクセスのブロックを解除し、索引更新コマンドを再実行してください。

**複数のパーティションにわたる表に対して SQL スカラー検索関数を使用できない場合** テキスト検索関数がパーティション表を直接参照しない状況、またはその関数が OUTER JOIN 節を含む副選択のメンバーである場合、検索はエラー (SQL0270N rc=109) を戻します。このエラーが発生した場合、異なる形式での照会の再書き込みを手動で試行し、照会を再発行してください。

#### パーティション・データベース環境セットアップ検証中の CTE0249 エラー

Windows の場合、パーティション・データベース環境セットアップを検証し、エラー「CTE0249 実行可能プログラム "ctepriisc" は異常終了しました」を検出した場合、次のアクションを実行してください。

1. パーティション・データベース環境構成の各コンピューターで、に従って、パーティション DB2 サーバーに対する環境の準備が完了しているかチェックしてください。具体的には、「Active Directory ユーザーとコンピュータ」コンソールで、各コンピューターのアカウントの「プロパティ」ダイアログ・ボックスの「全般」タブにある「コンピュータを委任に対して信頼する」チェック・ボックスがチェックされていることを確認してください。
2. 参加するすべてのコンピューター上で Windows サービス「DB2 Remote Command Server」が開始済みであることを確認してください。

#### ADMINISTRATION TABLES IN 節を使用せずに、範囲パーティション表でテキスト索引を作成する場合の CTE0150E エラー

範囲パーティション表にテキスト索引を作成する場合は、必ず ADMINISTRATION TABLES IN 節を指定してください。詳細については、CTE0150E を参照してください。





---

## 第 15 部 付録



---

## 付録 A. DB2 技術情報の概説

DB2 技術情報は、さまざまな方法でアクセスすることが可能な、各種形式で入手できます。

DB2 技術情報は、以下のツールと方法を介して利用できます。

- DB2インフォメーション・センター
  - トピック (タスク、概念、およびリファレンス・トピック)
  - サンプル・プログラム
  - チュートリアル
- DB2 資料
  - PDF ファイル (ダウンロード可能)
  - PDF ファイル (DB2 PDF DVD に含まれる)
  - 印刷資料
- コマンド行ヘルプ
  - コマンド・ヘルプ
  - メッセージ・ヘルプ

**注:** DB2 インフォメーション・センターのトピックは、PDF やハードコピー資料よりも頻繁に更新されます。最新の情報を入手するには、資料の更新が発行されたときにそれをインストールするか、[ibm.com](http://ibm.com) にある DB2 インフォメーション・センターを参照してください。

技術資料、ホワイト・ペーパー、IBM Redbooks® 資料などのその他の DB2 技術情報には、オンライン ([ibm.com](http://ibm.com)) でアクセスできます。DB2 Information Management ソフトウェア・ライブラリー・サイト (<http://www.ibm.com/software/data/sw-library/>) にアクセスしてください。

### 資料についてのフィードバック

DB2 の資料についてのお客様からの貴重なご意見をお待ちしています。DB2 の資料を改善するための提案については、[db2docs@ca.ibm.com](mailto:db2docs@ca.ibm.com) まで E メールを送信してください。DB2 の資料チームは、お客様からのフィードバックすべてに目を通しますが、直接お客様に返答することはありません。お客様が関心をお持ちの内容について、可能な限り具体的な例を提供してください。特定のトピックまたはヘルプ・ファイルについてのフィードバックを提供する場合は、そのトピック・タイトルおよび URL を含めてください。

DB2 お客様サポートに連絡する場合には、この E メール・アドレスを使用しないでください。資料を参照しても、DB2 の技術的な問題が解決しない場合は、お近くの IBM サービス・センターにお問い合わせください。

## DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式)

以下の表は、IBM Publications Center ([www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss](http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss)) から利用できる DB2 ライブラリーについて説明しています。英語および翻訳された DB2 バージョン 10.1 のマニュアル (PDF 形式) は、[www.ibm.com/support/docview.wss?rs=71&uid=swg27009474](http://www.ibm.com/support/docview.wss?rs=71&uid=swg27009474) からダウンロードできます。

この表には印刷資料が入手可能かどうかを示されていますが、国または地域によっては入手できない場合があります。

資料番号は、資料が更新される度に大きくなります。資料を参照する際は、以下にリストされている最新版であることを確認してください。

注: DB2 インフォメーション・センターは、PDF やハードコピー資料よりも頻繁に更新されます。

表 12. DB2 の技術情報

資料名	資料番号	印刷資料が入手可能かどうか	最終更新
管理 API リファレンス	SA88-4671-00	入手可能	2012 年 4 月
管理ルーチンおよびビュー	SA88-4672-01	入手不可	2013 年 1 月
コール・レベル・イン ターフェース ガイドお よびリファレンス 第 1 巻	SA88-4676-01	入手可能	2013 年 1 月
コール・レベル・イン ターフェース ガイドお よびリファレンス 第 2 巻	SA88-4677-01	入手可能	2013 年 1 月
コマンド・リファレン ス	SA88-4673-01	入手可能	2013 年 1 月
データベース: 管理の 概念および構成リファ レンス	SA88-4662-01	入手可能	2013 年 1 月
データ移動ユーティリ ティー: ガイドおよび リファレンス	SA88-4693-01	入手可能	2013 年 1 月
データベースのモニタ リング ガイドおよびリ ファレンス	SA88-4663-01	入手可能	2013 年 1 月
データ・リカバリーと 高可用性 ガイドおよび リファレンス	SA88-4694-01	入手可能	2013 年 1 月
データベース・セキュ リティ・ガイド	SA88-4695-01	入手可能	2013 年 1 月

表 12. DB2 の技術情報 (続き)

資料名	資料番号	印刷資料が入手可能 かどうか	最終更新
DB2 ワークロード管理 ガイドおよびリファレ ンス	SA88-4685-01	入手可能	2013 年 1 月
ADO.NET および OLE DB アプリケーション の開発	SA88-4665-01	入手可能	2013 年 1 月
組み込み SQL アプリ ケーションの開発	SA88-4666-01	入手可能	2013 年 1 月
Java アプリケーション の開発	SA88-4669-01	入手可能	2013 年 1 月
Perl、PHP、Python お よび Ruby on Rails ア プリケーションの開発	SA88-4670-00	入手不可	2012 年 4 月
IBM データ・サーバー 用の RDF アプリケー ション開発	SA88-5083-00	入手可能	2013 年 1 月
SQL および外部ルーチ ンの開発	SA88-4667-01	入手可能	2013 年 1 月
データベース・アプリ ケーション開発の基礎	GI88-4279-01	入手可能	2013 年 1 月
DB2 インストールおよ び管理 概説 (Linux お よび Windows 版)	GI88-4280-00	入手可能	2012 年 4 月
グローバリゼーショ ン・ガイド	SA88-4696-00	入手可能	2012 年 4 月
DB2 サーバー機能 イ ンストール	GA88-4679-01	入手可能	2013 年 1 月
IBM データ・サーバ ー・クライアント機能 インストール	GA88-4680-00	入手不可	2012 年 4 月
メッセージ・リファレ ンス 第 1 巻	SA88-4688-01	入手不可	2013 年 1 月
メッセージ・リファレ ンス 第 2 巻	SA88-4689-01	入手不可	2013 年 1 月
Net Search Extender 管 理およびユーザース・ ガイド	SA88-4691-01	入手不可	2013 年 1 月
パーティションおよび クラスタリングのガイ ド	SA88-4697-01	入手可能	2013 年 1 月
Preparation Guide for DB2 10.1 Fundamentals Exam 610	SC27-4540-00	入手不可	2013 年 1 月

表 12. DB2 の技術情報 (続き)

資料名	資料番号	印刷資料が入手可能 かどうか	最終更新
<i>Preparation Guide for DB2 10.1 DBA for Linux, UNIX, and Windows Exam 611</i>	SC27-4541-00	入手不可	2013 年 1 月
<i>pureXML ガイド</i>	SA88-4686-01	入手可能	2013 年 1 月
<i>Spatial Extender ユーザーズ・ガイドおよびリファレンス</i>	SA88-4690-00	入手不可	2012 年 4 月
<i>SQL プロシージャ言語: アプリケーションのイネーブルメントおよびサポート</i>	SA88-4668-01	入手可能	2013 年 1 月
<i>SQL リファレンス 第 1 巻</i>	SA88-4674-01	入手可能	2013 年 1 月
<i>SQL リファレンス 第 2 巻</i>	SA88-4675-01	入手可能	2013 年 1 月
<i>Text Search ガイド</i>	SA88-4692-01	入手可能	2013 年 1 月
<i>問題判別およびデータベース・パフォーマンスのチューニング</i>	SA88-4664-01	入手可能	2013 年 1 月
<i>DB2 バージョン 10.1 へのアップグレード</i>	SA88-4678-01	入手可能	2013 年 1 月
<i>DB2 バージョン 10.1 の新機能</i>	SA88-4684-01	入手可能	2013 年 1 月
<i>XQuery リファレンス</i>	SA88-4687-01	入手不可	2013 年 1 月

表 13. DB2 Connect 固有の技術情報

資料名	資料番号	印刷資料が入手可能 かどうか	最終更新
<i>DB2 Connect Personal Edition</i> インストールおよび構成	SA88-4681-00	入手可能	2012 年 4 月
<i>DB2 Connect サーバー機能</i> インストールおよび構成	SA88-4682-01	入手可能	2013 年 1 月
<i>DB2 Connect ユーザーズ・ガイド</i>	SA88-4683-01	入手可能	2013 年 1 月

## コマンド行プロセッサから SQL 状態ヘルプを表示する

DB2 製品は、SQL ステートメントの結果として生じる可能性がある状態に対応した SQLSTATE 値を戻します。SQLSTATE ヘルプは、SQL 状態および SQL 状態クラス・コードの意味を説明します。

## 手順

SQL 状態ヘルプを開始するには、コマンド行プロセッサを開いて以下のように入力します。

```
? sqlstate または ? class code
```

ここで、*sqlstate* は有効な 5 桁の SQL 状態を、*class code* は SQL 状態の最初の 2 桁を表します。

例えば、? 08003 を指定すると SQL 状態 08003 のヘルプが表示され、? 08 を指定するとクラス・コード 08 のヘルプが表示されます。

---

## 異なるバージョンの DB2 インフォメーション・センターへのアクセス

他のバージョンの DB2 製品の資料は、ibm.com<sup>®</sup> のそれぞれのインフォメーション・センターにあります。

### このタスクについて

DB2 バージョン 10.1 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v10r1> です。

DB2 バージョン 9.8 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9r8/> です。

DB2 バージョン 9.7 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/> です。

DB2 バージョン 9.5 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5> です。

DB2 バージョン 9.1 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9/> です。

DB2 バージョン 8 のトピックについては、DB2 インフォメーション・センターの URL (<http://publib.boulder.ibm.com/infocenter/db2luw/v8/>) を参照してください。

---

## コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの更新

ローカルにインストールした DB2 インフォメーション・センターは、定期的に更新する必要があります。

### 始める前に

DB2 バージョン 10.1 インフォメーション・センターが既にインストール済みである必要があります。詳しくは、「DB2 サーバー機能 インストール」の『DB2 セットアップ・ウィザードによる DB2 インフォメーション・センターのインストール』のトピックを参照してください。インフォメーション・センターのインストールに適用されるすべての前提条件と制約事項は、インフォメーション・センターの更新にも適用されます。

## このタスクについて

既存の DB2 インフォメーション・センターは、自動で更新することも手動で更新することもできます。

- 自動更新は、既存のインフォメーション・センターのフィーチャーと言語を更新します。自動更新を使用すると、手動更新と比べて、更新中にインフォメーション・センターが使用できなくなる時間が短くなるというメリットがあります。さらに、自動更新は、定期的に行う他のバッチ・ジョブの一部として実行されるように設定することができます。
- 手動更新は、既存のインフォメーション・センターのフィーチャーと言語の更新に使用できます。自動更新は更新処理中のダウン時間を減らすことができますが、フィーチャーまたは言語を追加する場合は手動処理を使用する必要があります。例えば、ローカルのインフォメーション・センターが最初は英語とフランス語でインストールされており、その後ドイツ語もインストールすることにした場合、手動更新でドイツ語をインストールし、同時に、既存のインフォメーション・センターのフィーチャーおよび言語を更新できます。しかし、手動更新ではインフォメーション・センターを手動で停止、更新、再始動する必要があります。更新処理の間はずっと、インフォメーション・センターは使用できなくなります。自動更新処理では、インフォメーション・センターは、更新を行った後に、インフォメーション・センターを再始動するための停止が発生するだけで済みます。

このトピックでは、自動更新のプロセスを詳しく説明しています。手動更新の手順については、『コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの手動更新』のトピックを参照してください。

## 手順

コンピューターまたはイントラネット・サーバーにインストールされている DB2 インフォメーション・センターを自動更新する手順を以下に示します。

1. Linux オペレーティング・システムの場合、次のようにします。
  - a. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、DB2 インフォメーション・センターは、`/opt/ibm/db2ic/V10.1` ディレクトリーにインストールされています。
  - b. インストール・ディレクトリーから `doc/bin` ディレクトリーにナビゲートします。
  - c. 次のように `update-ic` スクリプトを実行します。

```
update-ic
```
2. Windows オペレーティング・システムの場合、次のようにします。
  - a. コマンド・ウィンドウを開きます。
  - b. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、DB2 インフォメーション・センターは、`<Program Files>\IBM\DB2 Information Center\バージョン 10.1` ディレクトリーにインストールされています (`<Program Files>` は「Program Files」ディレクトリーのロケーション)。



- c. インストール・ディレクトリーから doc¥bin ディレクトリーにナビゲートします。
- d. 次のように update-ic.bat ファイルを実行します。

```
update-ic.bat
```

## タスクの結果

DB2 インフォメーション・センターが自動的に再始動します。更新が入手可能な場合、インフォメーション・センターに、更新された新しいトピックが表示されます。インフォメーション・センターの更新が入手可能でなかった場合、メッセージがログに追加されます。ログ・ファイルは、doc¥eclipse¥configuration ディレクトリーにあります。ログ・ファイル名はランダムに生成された名前です。例えば、1239053440785.log のようになります。

---

## コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの手動更新

DB2 インフォメーション・センターをローカルにインストールしている場合は、IBM から資料の更新を入手してインストールすることができます。

### このタスクについて

ローカルにインストールされた DB2 インフォメーション・センター を手動で更新するには、以下のことを行う必要があります。

1. コンピューター上の DB2 インフォメーション・センター を停止し、インフォメーション・センターをスタンドアロン・モードで再始動します。インフォメーション・センターをスタンドアロン・モードで実行すると、ネットワーク上の他のユーザーがそのインフォメーション・センターにアクセスできなくなります。これで、更新を適用できるようになります。DB2 インフォメーション・センターのワークステーション・バージョンは、常にスタンドアロン・モードで実行されます。を参照してください。
2. 「更新」機能を使用することにより、どんな更新が利用できるかを確認します。インストールしなければならない更新がある場合は、「更新」機能を使用してそれを入手およびインストールできます。

**注:** ご使用の環境において、インターネットに接続されていないマシンに DB2 インフォメーション・センター の更新をインストールする必要がある場合、インターネットに接続されていて DB2 インフォメーション・センター がインストールされているマシンを使用して、更新サイトをローカル・ファイル・システムにミラーリングしてください。ネットワーク上の多数のユーザーが資料の更新をインストールする場合にも、更新サイトをローカルにミラーリングして、更新サイト用のプロキシーを作成することにより、個々のユーザーが更新を実行するのに要する時間を短縮できます。

更新パッケージが入手可能な場合、「更新」機能を使用してパッケージを入手します。ただし、「更新」機能は、スタンドアロン・モードでのみ使用できます。

3. スタンドアロンのインフォメーション・センターを停止し、コンピューター上の DB2 インフォメーション・センター を再開します。

注: Windows 2008、Windows Vista (およびそれ以上) では、このセクションの後の部分でリストされているコマンドは管理者として実行する必要があります。完全な管理者特権でコマンド・プロンプトまたはグラフィカル・ツールを開くには、ショートカットを右クリックしてから、「管理者として実行」を選択します。

## 手順

コンピューターまたはイントラネット・サーバーにインストール済みの *DB2* インフォメーション・センターを更新するには、以下のようになります。

1. *DB2* インフォメーション・センターを停止します。
    - Windows では、「スタート」 > 「コントロール パネル」 > 「管理ツール」 > 「サービス」をクリックします。次に、「**DB2** インフォメーション・センター」サービスを右クリックして「停止」を選択します。
    - Linux では、以下のコマンドを入力します。

```
/etc/init.d/db2icdv10 stop
```
  2. インフォメーション・センターをスタンドアロン・モードで開始します。
    - Windows の場合:
      - a. コマンド・ウィンドウを開きます。
      - b. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、*DB2* インフォメーション・センターは、`Program_Files¥IBM¥DB2 Information Center¥バージョン 10.1` ディレクトリーにインストールされています (`Program_Files` は Program Files ディレクトリーのロケーション)。
      - c. インストール・ディレクトリーから `doc¥bin` ディレクトリーにナビゲートします。
      - d. 次のように `help_start.bat` ファイルを実行します。

```
help_start.bat
```
    - Linux の場合:
      - a. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、*DB2* インフォメーション・センターは、`/opt/ibm/db2ic/V10.1` ディレクトリーにインストールされています。
      - b. インストール・ディレクトリーから `doc/bin` ディレクトリーにナビゲートします。
      - c. 次のように `help_start` スクリプトを実行します。

```
help_start
```
- システムのデフォルト Web ブラウザーが開き、スタンドアロンのインフォメーション・センターが表示されます。
3. 「更新」ボタン (🔄) をクリックします。(ブラウザーで JavaScript が有効になっている必要があります。) インフォメーション・センターの右側のパネルで、「更新の検索」をクリックします。既存の文書に対する更新のリストが表示されます。
  4. インストール・プロセスを開始するには、インストールする更新をチェックして選択し、「更新のインストール」をクリックします。
  5. インストール・プロセスが完了したら、「完了」をクリックします。

6. 次のようにして、スタンドアロンのインフォメーション・センターを停止します。

- Windows の場合は、インストール・ディレクトリーの `doc\bin` ディレクトリーにナビゲートしてから、次のように `help_end.bat` ファイルを実行します。

```
help_end.bat
```

注: `help_end` バッチ・ファイルには、`help_start` バッチ・ファイルを使用して開始したプロセスを安全に停止するのに必要なコマンドが含まれています。`help_start.bat` は、Ctrl-C や他の方法を使用して停止しないでください。

- Linux の場合は、インストール・ディレクトリーの `doc/bin` ディレクトリーにナビゲートしてから、次のように `help_end` スクリプトを実行します。

```
help_end
```

注: `help_end` スクリプトには、`help_start` スクリプトを使用して開始したプロセスを安全に停止するのに必要なコマンドが含まれています。他の方法を使用して、`help_start` スクリプトを停止しないでください。

7. **DB2 インフォメーション・センター** を再開します。

- Windows では、「スタート」 > 「コントロール パネル」 > 「管理ツール」 > 「サービス」をクリックします。次に、「**DB2 インフォメーション・センター**」サービスを右クリックして「開始」を選択します。

- Linux では、以下のコマンドを入力します。

```
/etc/init.d/db2icdv10 start
```

## タスクの結果

更新された **DB2 インフォメーション・センター** に、更新された新しいトピックが表示されます。

---

## DB2 チュートリアル

DB2 チュートリアルは、DB2 データベース製品のさまざまな機能について学習するための支援となります。この演習をとおして段階的に学習することができます。

### はじめに

インフォメーション・センター (<http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/>) から、このチュートリアルの XHTML 版を表示できます。

演習の中で、サンプル・データまたはサンプル・コードを使用する場合があります。個々のタスクの前提条件については、チュートリアルを参照してください。

### DB2 チュートリアル

チュートリアルを表示するには、タイトルをクリックします。

「*pureXML* ガイド」の『**pureXML**®』

XML データを保管し、ネイティブ XML データ・ストアに対して基本的な操作を実行できるように、DB2 データベースをセットアップします。

---

## DB2 トラブルシューティング情報

DB2 データベース製品を使用する際に役立つ、トラブルシューティングおよび問題判別に関する広範囲な情報を利用できます。

### DB2 の資料

トラブルシューティング情報は、「問題判別およびデータベース・パフォーマンスのチューニング」または *DB2* インフォメーション・センター の『データベースの基本』セクションにあります。ここには、以下の情報が記載されています。

- DB2 診断ツールおよびユーティリティーを使用した、問題の切り分け方法および識別方法に関する情報。
- 最も一般的な問題のうち、いくつかの解決方法。
- DB2 データベース製品で発生する可能性のある、その他の問題の解決に役立つアドバイス。

### IBM サポート・ポータル

現在問題が発生していて、考えられる原因とソリューションを見つけるには、IBM サポート・ポータルを参照してください。Technical Support サイトには、最新の DB2 資料、TechNotes、プログラム診断依頼書 (APAR またはバグ修正)、フィックスパック、およびその他のリソースへのリンクが用意されています。この知識ベースを活用して、問題に対する有効なソリューションを探し出すことができます。

IBM サポート・ポータル ([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)) にアクセスしてください。

---

## ご利用条件

これらの資料は、以下の条件に同意していただける場合に限りご使用いただけます。

**適用度:** これらのご利用条件は、IBM Web サイトのあらゆるご利用条件に追加で適用されるものです。

**個人使用:** これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布 (頒布、送信を含む) または表示 (上映を含む) することはできません。

**商業的使用:** これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

**権利:** ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM は、これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態を提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。

**IBM の商標:** IBM、IBM ロゴおよび [ibm.com](http://ibm.com) は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。



---

## 付録 B. 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。IBM 以外の製品に関する情報は、本書の最初の発行時点で入手可能な情報に基づいており、変更される場合があります。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510  
東京都中央区日本橋箱崎町19番21号  
日本アイ・ビー・エム株式会社  
法務・知的財産  
知的財産権ライセンス渉外

**以下の保証は、国または地域の法律に沿わない場合は、適用されません。** IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。



本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Canada Limited  
U59/3600  
3600 Steeles Avenue East  
Markham, Ontario L3R 9Z7  
CANADA

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができませんが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、



利便性もしくは機能性があることをほのめかしたり、保証することはできません。サンプル・プログラムは、現存するままの状態を提供されるものであり、いかなる種類の保証も提供されません。IBM は、これらのサンプル・プログラムの使用から生ずるいかなる損害に対しても責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生した創作物には、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。© Copyright IBM Corp. \_年を入れる\_. All rights reserved.

## 商標

IBM、IBM ロゴおよび [ibm.com](http://ibm.com) は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

以下は、それぞれ各社の商標または登録商標です。

- Linux は、Linus Torvalds の米国およびその他の国における商標です。
- Java™ およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。
- UNIX は The Open Group の米国およびその他の国における登録商標です。
- インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Celeron、Intel SpeedStep、Itanium、Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。
- Microsoft、Windows、Windows NT、および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。



# 索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

## [ア行]

### アンインストール

#### Net Search Extender

概要 39

db2nse\_deinstall コマンド 254

UNIX 39

Windows 39

### インスタンス・サービス

開始 83

概要 10

更新サービス 89

停止 83

ロッキング・サービス 85

### インストール

#### Net Search Extender

検査 (UNIX) 37

検査 (Windows) 37

システム要件 25

ディレクトリー名 33

パーティション化された DB2 サーバー 27

ファイル名 33

db2nsels コマンド 255

db2nse\_install コマンド 253

Outside-In フィルター・ソフトウェア 35

UNIX 29

Windows 31

### インフォメーション・カタログ

概要 283

ビュー

索引レベル情報 286

データベース・レベル情報 283

テキスト索引の表 290

## [カ行]

### 外部結合

テキスト検索 153

### 外部保管データ

概要 10

### 関数

Net Search Extender 検索 265

### 検索

XML 文書処理 177

### 検索関数

概要 265

### 検索関数 (続き)

詳細 133

ストアド・プロシージャー 145, 275

パフォーマンスに関する考慮事項 155

例 143

CONTAINS 265

DB2EXT.HIGHLIGHT 270

DB2EXT.TEXTSEARCH 267

NUMBEROFMATCHES 266

SCORE 266

SQL スカラー 135

SQL 表値 147

### 検索パラメーター 260

### 検索指数

概要 257

構文 257

### 検索指数キーワード

THESAURUS 260

### 高可用性

Net Search Extender 41

### 更新

DB2 インフォメーション・センター 319, 321

### 構造化文書

詳細 183

文書モデル 183

Net Search Extender 175

### コマンド

CONTROL 207

START 209

STOP 211

### ご利用条件

資料 324

### コンパイル

シソーラス 163

## [サ行]

### サーバー

障害のトレース 303

### 再編成

テキスト索引 113

### 索引

イベントのクリア 119

ニックネーム 11

### シソーラス

エラー・メッセージ 168

概要 161

関係の定義 163

検索構文 260

構造 161

コンパイル 163

## シソーラス (続き)

- 作成 163
- サポート 165
- サポートされているコード化文字セット ID (CCSID) 167
- 事前定義された関係 162
- 定義ファイルのコンパイル 165
- 定義ファイルの作成 164
- THESAURUS キーワード 260

## シソーラス検索

- 構文 260
- THESAURUS キーワード 260

## シナリオ

- Net Search Extender
  - 概要 157
  - ストアド・プロシージャ検索 159
  - SQL スカラー検索 157
  - SQL 表値関数 160

## 障害の検出 303

## 障害のトレース 303

## 資料

- 印刷 316
- 概要 315
- 使用に関するご利用条件 324
- PDF ファイル 316

## ストアド・プロシージャ検索

- 概要 7
- 関数 275
- DB2EXT.TEXTSEARCH コマンド 275

## ストップワード 172

# [夕行]

## チュートリアル

- トラブルシューティング 324
- 問題判別 324
- リスト 323
- pureXML 323

## データベース・パーティション

- Net Search Extender 61

## テキスト検索エンジン

- 文字正規化 173
- ワードの一部としての特殊文字の使用 173

## テキスト索引

- 更新 113
- 再編成 113
- 索引イベントのクリア 119
- 作成

- 概要 91, 97
- サポートされないデータ・タイプ 100
- ストアド・プロシージャ検索 102
- データベースの使用可能化 93
- データベースを使用不可にする 95
- ニックネーム 101
- バイナリー・データ・タイプ 99
- 範囲パーティション表 107

## 状況の表示 127

## テキスト索引 (続き)

- 詳細 111
- データベースのドロップ 305
- ドロップ 125
- バックアップ 129
- ビュー 105
- 表のドロップ 305
- 変更 117
- 保守 91
- リストア 129
- /tmp ディレクトリからのファイルの除去 131

## トークン化 171

## 特記事項 327

## トラブルシューティング

- オンライン情報 324
- チュートリアル 324

# [八行]

## パーティション表

- 概要 12
- パーティション・データベース
  - Net Search Extender 10

## パフォーマンス

- 索引
  - 推奨 109
  - 低下の原因 109

## ビュー

- db2ext.textindexes 127

## 表値関数

- 概要 8

## 文書フォーマット

- Outside-In フィルター・ソフトウェア 73

## 文書モデル

- 概要 184
- 参照情報 199
- テキスト・フィールド
  - 概要 185
  - 制限 202
- デフォルト 186
- プレーン・テキスト
  - 定義 187
  - 例 184

## 文書属性

- 概要 186
- 制限 202
- 文書タイプ定義 199
- Outside-In フィルター・テクノロジー 195
- XML 191

## ヘルプ

- SQL ステートメント 319
- 保全性処理に基づく増分更新 13

## [マ行]

### マイグレーション

#### Net Search Extender

ツール 49

32 ビットから 64 ビット 49

### メモリー

#### 所要量

AIX 63

Linux 65

Solaris 64

Windows 63

ストアード・プロシージャ 63

文字正規化 173

### 戻りコード

インストール 307

### 問題判別

チュートリアル 324

利用できる情報 324

## [ヤ行]

### ユーザー・ロール

データベース管理者 75

テキスト表所有者 75

DB2 インスタンス所有者 75

Net Search Extender 75

## [ラ行]

列トランスフォーメーション関数 9

### レプリケーション

キャプチャー表 227

レプリケーション・キャプチャー表 227

### ロッキング・サービス

詳細 85

スナップショットの表示 87

## [ワ行]

ワードの一部としての特殊文字の使用 173

## A

ACTIVATE CACHE コマンド 220

ALTER INDEX コマンド 221

## C

checknseindex コマンド

詳細 123

CLEAR EVENTS コマンド 225

CONTAINS 関数 265

CONTROL コマンド 207

COPYRIGHT コマンド 252

CREATE INDEX コマンド 227

ctemigridx ツール 53

## D

DB2 インフォメーション・センター

更新 319, 321

バージョン 319

DB2EXTHL コマンド 217

DB2EXTTH コマンド 245

DB2EXT.HIGHLIGHT 関数 270

DB2EXT.TEXTSEARCH 関数 267

DB2EXT.TEXTSEARCH コマンド 275

db2nseis コマンド 255

db2nse\_deinstall コマンド 254

db2nse\_install コマンド 253

db2text コマンド 19

DEACTIVATE CACHE コマンド 242

DISABLE DATABASE コマンド 215

DROP INDEX Net Search Extender コマンド 244

## E

ENABLE DATABASE コマンド 213

## G

GPP 文書モデル

詳細 187

文書タイプ定義 199

要素パラメーター 187

GPP 文書の索引付け 188

## H

HELP コマンド

DB2 Net Search Extender 251

HTML 文書モデル DTD 199

## N

Net Search Extender

アップグレード 45

インスタンス・サービス 10, 81

インストール 23

外部結合 153

概要 3, 17

拡張テキスト保持ステージング・インフラストラクチャー  
77

管理表

概要 10

ビュー 10

計画 57

コード・ページ 69

## Net Search Extender (続き)

- 高可用性災害時リカバリー 41
- 索引ストレージ 59
- 索引妥当性検査ユーティリティ 123
- 索引ファイル破損
  - 防止 71
- 索引マイグレーション 53
- 索引名 67
- ストップワード 172
- 増分更新 77
- 属性値 186
- データベース
  - 使用可能化 93
  - 使用不可にする 95
- ディレクトリーのロケーション 59
- テキスト検索エンジン 171
- テキスト索引の妥当性検査 121
- トークン化 171
- 表名 67
- ヒント 309
- 文書フォーマット 69
- 別個の fenced ユーザーのサポート 14
- 文字正規化 173
- リソース 61
- 理由コード 293
- 列名 67
- ワードの一部としての特異文字 173
- XML サポート 12

## Net Search Extender のビュー

- イベント情報 290
- 索引レベル情報
  - db2ext.indexconfiguration 288
  - db2ext.textconfiguration 286
  - db2ext.textindexes 286, 287
  - db2ext.textindexformats 289
  - db2ext.textinformats 286
- データベース・レベル情報
  - db2ext.dbdefaults 283
- ログ表 291

## Net Search Extender 用の管理コマンド

- インスタンス所有者
  - サマリー 207
  - CONTROL 207
  - START 209
  - STOP 211
- データベース管理者
  - サマリー 213
  - DB2EXTHL 217
  - DISABLE DATABASE 215
  - ENABLE DATABASE 213
- テキスト表所有者
  - サマリー 219
  - ACTIVATE CACHE 220
  - ALTER INDEX 221
  - CLEAR EVENTS 225
  - COPYRIGHT 252

## Net Search Extender 用の管理コマンド (続き)

- テキスト表所有者 (続き)
  - CREATE INDEX 227
  - DB2EXTTH 245
  - DEACTIVATE CACHE 242
  - DROP INDEX 244
  - HELP 251
  - RESET PENDING 246
  - UPDATE INDEX 247
- NUMBEROFMATCHES 関数 266

## O

### Outside-In フィルター

- 索引付け 197
- 詳細 195
- 属性値 203

## R

- RESET PENDING コマンド 246

## S

### SCORE 関数

- 詳細 266
- パラメーター 266

### SQL

- 表値関数
  - 強調表示 147

### SQL 検索引数

- 同じ段落内での用語 140
- 同じ文内での用語 140
- 固定順序での用語 140
- シソーラス検索 141
- 数値属性検索 141
- セクション内の用語 140
- 任意の順序での用語 137
- ブール演算子 AND 137
- ブール演算子 NOT 138
- ブール演算子 OR 137
- ファジー検索 138
- フリー・テキスト検索 142
- マスキングでのエスケープ文字 139
- ワイルドカード文字マスキング 139

### SQL スカラー検索関数

- 概要 6
- 複数列の検索 151
- CONTAINS 135
- NUMBEROFMATCHES 136
- SCORE 136
- SQL 検索引数 137

### SQL ステートメント

- ヘルプ
  - 表示 319

START Net Search Extender コマンド 209

STOP コマンド 211

## U

UPDATE INDEX Net Search Extender コマンド 247

## W

Windows

システム・エラー 279

## X

XML 分解

Net Search Extender 186

XML 文書処理

カスタマイズした XML 文書モデル 178

セクション検索 177

デフォルト XML 文書モデル 177

XQuery サポート 179

XML 文書モデル

詳細 191

文書タイプ定義 199

XPath 式のセマンティクス 200









Printed in Japan

SA88-4691-01



日本アイ・ビー・エム株式会社  
〒103-8510 東京都中央区日本橋箱崎町19-21

Spine information:

IBM DB2 10.1 for Linux, UNIX, and Windows

Net Search Extender 管理およびユーザーズ・ガイド

