

DB2<sup>®</sup> ユニバーサル・データベース



# DB2 テキスト・エクステンダー 管理およびプログラミング

バージョン 7



DB2<sup>®</sup> ユニバーサル・データベース



# DB2 テキスト・エクステンダー 管理およびプログラミング

バージョン 7

**ご注意!**

本書、および本書がサポートする製品をご使用になる前に、349ページの『特記事項』にある一般的な情報を必ずお読みください。

本書は、DB2 ユニバーサル・データベース (DB2 UDB) バージョン 7 の DB2 テキスト・エクステンダー (5645-DB2) および新版において特に断りのない限り、それ以降のすべてのリリースおよびモディフィケーションに適用されます。

この版は、SD88-7282-00 の改訂版です。

本書において、日本では発表されていない IBM 製品 (機械およびプログラム)、プログラミング、およびサービスについて言及または説明する場合があります。しかし、このことは、弊社がこのような IBM 製品、プログラミング、またはサービスを、日本で発表する意図であることを必ずしも示すものではありません。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

原典：	SC26-9930-00 DB2 <sup>®</sup> Universal Database DB2 Text Extender Administration and Programming Version 7
発行：	日本アイ・ビー・エム株式会社
担当：	ナショナル・ランゲージ・サポート

第1刷 2000.6

この文書では、平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体\*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注\* 平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、  
平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 1995, 2000. All rights reserved.

Translation: © Copyright IBM Japan 2000

# 目次

本書について . . . . .	vii	DB2 テキスト・エクステンダー・サーバーの 設定および保守 . . . . .	23
本書の対象読者 . . . . .	vii	DB2 テキスト・エクステンダー・インスタ ンスの作成 . . . . .	23
本書の使用方法 . . . . .	vii	DB2 テキスト・エクステンダー・サーバ ーの開始および停止 . . . . .	24
構文図の読み方 . . . . .	viii	索引および使用可能なデータベースのバッ クアップおよび復元 . . . . .	25
改訂の要約 . . . . .	xi	障害のトレース . . . . .	26
<b>第1部 ガイド . . . . .</b>	<b>1</b>	<b>第3章 はじめに . . . . .</b>	<b>29</b>
<b>第1章 DB2 テキスト・エクステンダーの概説 3</b>	<b>3</b>	テキストを検索可能にする簡単な例 . . . . .	29
DB2 クライアント / サーバー環境における		索引付けおよび検索のためのサンプル・ウィ ザード . . . . .	31
DB2 テキスト・エクステンダー . . . . .	4	<b>第4章 必要に応じた検索の計画 . . . . .</b>	<b>33</b>
区分データベース環境での DB2 テキスト・エ クステンダー . . . . .	7	テキスト文書を索引付けする理由 . . . . .	33
<b>第2章 インストール、設定、および保守 . . . 9</b>	<b>9</b>	サポートされる文書形式 . . . . .	35
AIX、Sun-Solaris、および HP-UX システムへ のインストール . . . . .	9	HTML 文書に関する特別な考慮事項 . . . . .	36
インストール要件 . . . . .	9	XML 文書に関する特別な考慮事項 . . . . .	37
ステップ 1 (AIX 用): 製品コンポーネン トのインストール . . . . .	9	サポートされていない文書形式の使用 言語 . . . . .	37
ステップ 1 (Sun-Solaris 用): 製品コンポー ネントのインストール . . . . .	11	CCSID . . . . .	39
ステップ 1 (HP-UX 用): 製品コンポーネン トのインストール . . . . .	11	EBCDIC . . . . .	39
ステップ 2: DB2 テキスト・エクステンダ ーのインスタンスを確立する . . . . .	14	ASCII . . . . .	40
ステップ 3: DB2 テキスト・エクステンダ ー・サーバーの開始 . . . . .	14	DBCS . . . . .	41
Windows 95、Windows 98、Windows NT、お よび Windows 2000 へのインストール . . . . .	15	UNICODE . . . . .	42
サーバー要件 . . . . .	15	テキストの保管および使用可能化の際に起 きるコード・ページに関する問題の回避 . . . . .	42
クライアント要件 . . . . .	15	検索のタイプ . . . . .	45
DB2 エクステンダーのインストール . . . . .	15	基本語形検索 . . . . .	46
インスタンスの作成および管理 . . . . .	18	一致語句検索 . . . . .	48
インストール検査のためにサンプル・デー タベースを準備する . . . . .	18	ファジー検索または DBCS 文書の検索 . . . . .	48
構成 . . . . .	19	索引タイプの変更 . . . . .	49
環境変数 . . . . .	19	表に対する 1 つまたは複数のテキスト索引の 作成 . . . . .	49
テキスト構成設定値 . . . . .	21	索引サイズの計算 . . . . .	50
テキスト構成の変更 . . . . .	22	索引の更新 . . . . .	51
		マルチノード環境での索引の保守 . . . . .	52
		構造化文書 (セクション・サポート) での作業 属性セクション . . . . .	53

フラット・ファイルおよび HTML 文書 . . . . .	56	任意の順序での用語の検索 . . . . .	95
XML 文書 . . . . .	58	同じ文または段落中の用語の検索 . . . . .	95
辞書、停止ワード・リスト、省略後リスト、 および言語パラメーター . . . . .	59	構造化文書のセクションでの用語の検索 . . . . .	96
停止ワードと省略語ファイルの変更 . . . . .	60	用語の同義語の検索 . . . . .	96
<b>第5章 テキストを検索可能にする . . . . .</b>	<b>63</b>	基本語形検索の実行 . . . . .	97
テキストを検索可能にする前の準備 . . . . .	63	ブール演算子 NOT を使用した検索 . . . . .	98
DB2 テキスト・エクステンダー・コマンド行 プロセッサの開始 . . . . .	65	ファジー検索 . . . . .	98
コマンド行プロセッサのヘルプ . . . . .	66	語句の境界の考慮 . . . . .	99
データベースへの接続 . . . . .	67	発音の似た語の検索 . . . . .	99
データベースの使用可能化 . . . . .	68	シソーラス検索 . . . . .	99
テキスト表の使用可能化 (任意選択) . . . . .	69	フリー・テキストと混成検索 . . . . .	100
例 . . . . .	70	直前の検索の詳細化 . . . . .	101
テキスト列の使用可能化 . . . . .	72	ハンドル内の情報の設定と抽出 . . . . .	103
ハンドル列の追加 . . . . .	73	新規テキストを挿入する際のテキスト情報 の設定 . . . . .	103
文書情報の設定 . . . . .	74	ハンドルからの情報の抽出 . . . . .	104
ログ表の作成 . . . . .	74	ハンドル内の情報の変更 . . . . .	104
索引の作成 . . . . .	75	検索パフォーマンスの向上 . . . . .	105
例 . . . . .	75	<b>第7章 管理 . . . . .</b>	<b>107</b>
ラージ表でのテキスト列の使用可能化 . . . . .	76	テキスト索引の保守 . . . . .	107
サポートされないデータ・タイプのテキス ト列の使用可能化 . . . . .	77	索引の更新 . . . . .	107
データ・タイプ DATALINK を含む列の使 用可能化 . . . . .	78	外部ファイルの索引の更新 . . . . .	108
外部テキスト・ファイルの使用可能化 . . . . .	79	索引の設定の変更 . . . . .	108
例 . . . . .	80	索引状況のリセット . . . . .	109
セッションの終了 . . . . .	81	索引事象の削除 . . . . .	110
<b>第6章 検索方法 . . . . .</b>	<b>83</b>	索引の再編成 . . . . .	110
検索関数の構文例の情報源 . . . . .	84	有用な情報の入手 . . . . .	111
DB2TX.SAMPLE サンプル表 . . . . .	84	使用可能状況情報の表示 . . . . .	111
外部ファイルのハンドル . . . . .	88	環境変数の設定値の表示 . . . . .	112
現行の関数パスの設定 . . . . .	89	テキスト構成設定値の表示 . . . . .	113
テキストの検索 . . . . .	89	索引状況の表示 . . . . .	113
照会の作成 . . . . .	90	エラー事象の表示 . . . . .	115
検出された一致数の検索と入手 . . . . .	91	索引設定の表示 . . . . .	116
検出テキスト文書のランクの検索と入手 . . . . .	91	列のテキスト設定の表示 . . . . .	117
検索引き数の指定 . . . . .	92	DB2 テキスト・エクステンダー・カタログ表 示の処理 . . . . .	118
複数の用語の検索 . . . . .	92	テキスト準備処理の逆処理 . . . . .	120
ブール演算子 AND および OR を使用した 検索 . . . . .	92	テキスト列を使用不可にする . . . . .	120
用語の変化形の検索 . . . . .	93	テキスト・ファイルを使用不可にする . . . . .	121
用語の一部 (マスキング文字) の検索 . . . . .	93	テキスト表を使用不可にする . . . . .	121
マスキング文字をすでに含む用語の検索 . . . . .	95	データベースを使用不可にする . . . . .	122
		<b>第8章 検索およびブラウズのための API 関 数の使用法 . . . . .</b>	<b>123</b>
		アプリケーションのセットアップ . . . . .	123

UNIX アプリケーションのリンク	123
OS/2 または Windows アプリケーションのリンク	124
API 関数の概説	124
テキストの検索	126
検索結果表の入手 (DesGetSearchResultTable)	126
テキストのブラウズ	127
ブラウズ情報の入手 (DesGetBrowseInfo)	127
ブラウズ・セッションの開始 (DesStartBrowseSession)	127
文書のオープン (DesOpenDocument)	128
一致情報の入手 (DesGetMatches)	128
文書のクローズ (DesCloseDocument)	129
ブラウズ・セッションの終了 (DesEndBrowseSession)	129
ブラウズ情報の解放 (DesFreeBrowseInfo)	129

## 第2部 参照情報 . . . . . 131

第9章 クライアント向けのテキスト作成コマンドと管理コマンド . . . . .	133
CHANGE INDEX SETTINGS . . . . .	135
CHANGE TEXT CONFIGURATION . . . . .	137
CONNECT . . . . .	140
DELETE INDEX EVENTS . . . . .	142
DISABLE DATABASE . . . . .	143
DISABLE TEXT COLUMN . . . . .	144
DISABLE TEXT FILES . . . . .	145
DISABLE TEXT TABLE . . . . .	146
ENABLE DATABASE . . . . .	147
ENABLE TEXT COLUMN . . . . .	148
ENABLE TEXT FILES . . . . .	157
ENABLE TEXT TABLE . . . . .	160
GET ENVIRONMENT . . . . .	165
GET INDEX SETTINGS . . . . .	166
GET INDEX STATUS . . . . .	168
GET STATUS . . . . .	169
GET TEXT CONFIGURATION . . . . .	170
GET TEXT INFO . . . . .	171
QUIT . . . . .	172
REORGANIZE INDEX . . . . .	173
RESET INDEX STATUS . . . . .	174
UPDATE INDEX . . . . .	175

## 第10章 サーバー向け管理コマンド . . . . . 177

TXICRT . . . . .	179
TXIDROP . . . . .	181
TXILIST (UNIX のみ) . . . . .	182
TXIPCLEAN (UNIX のみ) . . . . .	183
TXNADD . . . . .	184
TXNCHECK . . . . .	185
TXNDROP . . . . .	186
TXSAMPLE . . . . .	187
TXSTART . . . . .	188
TXSTATUS . . . . .	189
TXSTOP . . . . .	190
TXTHESC . . . . .	191
TXTHESN . . . . .	193
TXTRACE . . . . .	195
TXVERIFY . . . . .	200
TXWIZARD . . . . .	201

## 第11章 検索関数 . . . . . 203

DB2 テキスト・エクステンダー特殊タイプ	203
DB2 テキスト・エクステンダー関数の要約	204
CCSID . . . . .	206
CONTAINS . . . . .	207
FILE . . . . .	208
FORMAT . . . . .	209
HANDLE . . . . .	210
HANDLE_LIST . . . . .	211
INIT_TEXT_HANDLE . . . . .	212
LANGUAGE . . . . .	214
NO_OF_DOCUMENTS . . . . .	215
NO_OF_MATCHES . . . . .	216
RANK . . . . .	217
REFINE . . . . .	218
SEARCH_RESULT . . . . .	219

## 第12章 検索引き数の構文 . . . . . 221

検索引き数 . . . . .	223
-----------------	-----

## 第13章 検索およびブラウズのための API 関数 . . . . . 235

DesCloseDocument . . . . .	237
DesEndBrowseSession . . . . .	238
DesFreeBrowseInfo . . . . .	239
DesGetBrowseInfo . . . . .	240
DesGetMatches . . . . .	244
DesGetSearchResultTable . . . . .	250
DesOpenDocument . . . . .	255
DesStartBrowseSession . . . . .	258

<b>第14章 サンプル API プログラム</b> . . . . .	<b>261</b>	テキスト分析構成ファイル . . . . .	294
		共通オプション . . . . .	295
<b>第15章 基本語形索引と一致語句索引の言語 的な処理</b> . . . . .	<b>263</b>	<b>第17章 戻りコード</b> . . . . .	<b>299</b>
索引作成時の言語的な処理 . . . . .	263	<b>第18章 メッセージ</b> . . . . .	<b>309</b>
基本テキスト分析 . . . . .	264	DB2 テキスト・エクステンダーの関数により 戻される SQL の状態 . . . . .	309
用語の基本形への変形 (lemmatization) . . . . .	269	DB2 テキスト・エクステンダーからのメッセ ージ . . . . .	313
停止ワードのフィルター処理 . . . . .	270	<b>第19章 検索エンジンの理由コード</b> . . . . .	<b>327</b>
分解 (複合語の分割) . . . . .	270	<b>第20章 エラー事象理由コード</b> . . . . .	<b>331</b>
検索のための言語的な処理 . . . . .	271	<hr/>	
同義語 . . . . .	272	<b>第3部 付録および後付け</b> . . . . .	<b>347</b>
シソーラスの拡張 . . . . .	272	特記事項 . . . . .	349
発音の拡張 . . . . .	273	商標 . . . . .	352
文字と語のマスキング . . . . .	273	用語集 . . . . .	355
ブラウズのための言語的な処理 . . . . .	273	索引 . . . . .	361
ステージ 1: 正規化と用語拡張 . . . . .	273	<b>IBM と連絡をとる</b> . . . . .	<b>373</b>
ステージ 2: 拡張突き合わせ . . . . .	274	製品情報 . . . . .	373
シソーラスの概念 . . . . .	275		
用語 . . . . .	277		
関係 . . . . .	277		
Ngram シソーラスの関係 . . . . .	279		
シソーラスの作成 . . . . .	280		
Ngram シソーラスの作成 . . . . .	285		
<b>第16章 構成ファイル</b> . . . . .	<b>289</b>		
クライアント構成ファイル . . . . .	289		
サーバー構成ファイル . . . . .	291		



---

## 本書について

本書は、テキスト・データを検索するために DB2(R) データベースを準備および保守するにあたって、DB2 テキスト・エクステンダーを使用する方法を説明します。さらに、DB2 テキスト・エクステンダーに備わっている SQL 関数およびアプリケーション・プログラミング・インターフェース (API) を使用して、それらのタイプのデータにアクセスし、操作する方法を示します。DB2 テキスト・エクステンダーの機能をプログラムの SQL ステートメントに組み込んだり、API を組み込んだりすることにより、強力で用途の広いテキスト検索プログラムを作成することができます。

本書で「DB2」に言及している場合、DB2 UDB を指しています。

---

## 本書の対象読者

本書は、DB2 管理の概念、ツール、および技法に精通している DB2 管理者を対象としています。

また、SQL と、DB2 アプリケーション・プログラムに使用できる 1 つ以上のプログラミング言語に精通した、DB2 アプリケーション・プログラマーも対象としています。

本書は、DB2 テキスト・エクステンダーを使用して作業するユーザーを対象としています。DB2 イメージ、オーディオ、およびビデオ・エクステンダーを使って作業するユーザーは、DB2 イメージ、オーディオ、およびビデオ・エクステンダー 管理およびプログラミングの手引き を参照してください。

---

## 本書の使用方法

本書は以下のように構成されています。

### 『第 1 部: ガイド』

この部分では、DB2 テキスト・エクステンダーについて概説すると共に、インストール方法および設定方法、そして計画を立てる際の考慮事項について説明します。さらに、テキストを検索できるように、DB2 データベースを準備して保守する方法についても説明します。

## 本書について

DB2 テキスト・エクステンダーについてあまりご存じではなく、テキストを検索するために DB2 テキスト・エクステンダー関数および API の使用方法を知りたい場合は、この部分をお読みください。

### 『第 2 部: 参照』

この部分では、DB2 テキスト・エクステンダーの関数、API、コマンド、およびメッセージやコードなどの診断情報についての参照情報を示します。

DB2 テキスト・エクステンダーの概念およびタスクを理解しているものの、特定の DB2 テキスト・エクステンダー関数、API、コマンド、メッセージ、またはコードについての情報が必要な場合、この部分をお読みください。

---

## 構文図の読み方

本書では、構文は次のように定義された構造を用いて記述されています。

- 構文図は、左から右、上から下に、線に沿って読みます。

▶— 記号は、ステートメントの開始を示します。

—▶ 記号は、構文図が次の行に続くことを示します。

▶— 記号は、ステートメントが前の行から続いていることを示します。

—▶ 記号は、ステートメントの終わりを示します。

- 必須項目は、水平線 (メインパス) の線上に示されます。

▶—必須項目—▶

- オプション項目は、メインパスの下に示されます。

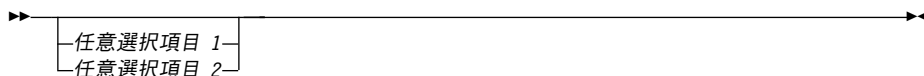
▶—  
└任意選択項目└▶

- 2 つ以上の項目からの選択が可能な場合は、複数の項目が縦に並んで記載されます。

項目から 1 つを選択しなければならない場合、スタック内の項目の 1 つがメインパスの線上に示されます。

▶—  
└必須選択項目 1└  
└必須選択項目 2└▶

項目から選択しなくてもよい場合には、併記された項目全体が主線の下に記載されます。



併記された項目の上を矢印が折り返している場合、併記された項目から 2 つ以上を選択することが可能であることを示します。



- キーワードは大文字で表記されています。キーワードの記載は、表記どおりでなければなりません。変数は小文字で表記されています (たとえば、`srcpath`)。変数は、構文中でのユーザー指定の名前または値を表します。
- 句読記号、括弧、算術演算子、または同様の他の記号が示されている場合、それらを構文の一部として入力しなければなりません。

## 本書について

---

## 改訂の要約

本書 (リリース 7.1 用) における変更には、以下の項目が含まれます。

- **特殊なタイプのセクションの検索**

サポートされるタイプは、日付 / 時刻および整数です。これにより、たとえば Publishing date セクションに 12/13/99 より後の日付を持つ文書を検索することが可能になります。

- **データ・リンク・サポート**

データ・リンクのデータ・タイプを持つ列を介して参照される文書を、索引付けすることができます。

- **サポートされる CCSID の拡大**

Ngram 索引タイプでの CCSID に関する制限が除かれました。

- **併用索引タイプのサポートの中止**

代わりに、基本語形索引タイプまたは正規化された一致語句索引タイプを使用してください。

- **search\_result 構文の変更**

search\_result 関数の構文が変更されました。以前の構文も、逆方向の互換性のために引き続きサポートされます。

- **サポートされる言語の拡大**

サポートされる言語および CCSID のリストが拡張されました。

- **構成機能の拡張**

トークン化機能を構成して、サーバー構成をカスタマイズできます (たとえば、しきい値の再編成など)。

- **UNIX 用の新しいシステム終結処置コマンド**

DB2 テキスト・エクステンダー・サーバーが正常に開始しない場合のシステム終結処置を簡単にするために、UNIX プラットフォーム用の新しいコマンド TXIPCLEAN が備わっています。

## 変更の要約

---

# 第1部 ガイド





---

## 第1章 DB2 テキスト・エクステンダーの概説

DB2 テキスト・エクステンダーは、DB2 エクステンダー・ファミリーの 1 つです。プログラマーはこれを使って、アプリケーションにテキスト文書用の SQL 照会を組み込むことができます。

他のエクステンダーを使用すれば、画像データ、ビデオ・データ、および音声データを検索することができ、XML 文書および場所データを処理することができます。

DB2 テキスト・エクステンダーは、構造化されていないテキスト文書をデータベースに保管できるという DB2 の機能を利用して、SQL 照会に全テキスト検索の機能を追加します。

DB2 ユーザーおよびアプリケーション・プログラマーは、DB2 テキスト・エクステンダーを使用して、このような文書を短時間に、多様かつ高機能な方法で検索することができます。DB2 テキスト・エクステンダーの利点は、何千もの大量のテキスト文書を高速で検索し、直接要求したものだけでなくその変形語や同義語も検出できるということです。

ユーザーは、DB2 データベース内に格納されているテキスト文書だけでなく、ファイルに格納されたテキスト文書も検索することができます。

DB2 テキスト・エクステンダーの中心となるのが、263ページの『第15章 基本語形索引と一致語句索引の言語的な処理』で説明されている高性能な言語検索技術です。これによりアプリケーションは、テキスト文書にアクセスし、さまざまな方法で検索を行うことができます。以下をアプリケーションで実行できます。

- 特定のテキストまたはワードの同義語が含まれている文書の検索、あるいは同じ文や段落中などに近接する検索用語が含まれている文書の検索。
- 語および文字の前、真ん中、および後の部分をマスクしたワイルドカード検索。
- さまざまな文書形式の多言語の文書検索。
- 検索語と似たつづりの語の「ファジー」検索。これは、つづりを誤った語でも検索できるので便利です。
- 検索引き数が自然言語で表されるフリー・テキスト検索。
- 検索項目と類似した発音の語の検索。

## 概説

テキスト検索と業務データ照会を結合できます。たとえば、アプリケーション中の SQL 照会をコーディングして、ある期間内に特定の作成者により作成された、特定の語または句を含むテキスト文書を検索することができます。また、DB2 テキスト・エクステンダー・プログラミング・インターフェースを使用すれば、アプリケーション・ユーザーが文書をブラウズすることもできます。

DB2 の SELECT 照会に全テキスト検索を組み込むことによって、高機能の検索を行うことができます。次の SQL ステートメントはこの一例です。

```
SELECT * FROM MyTextTable
  WHERE version = '2'
  AND DB2TX.CONTAINS (
    DB2BOOKS_HANDLE,
    "authorization"
    IN SAME PARAGRAPH AS "table"
    AND SYNONYM FORM OF "delete") = 1
```

DB2TX.CONTAINS は、いくつかある DB2 テキスト・エクステンダー検索関数のうちの 1 つです。DB2BOOKS\_HANDLE は、検索するテキスト文書を含んだ DB2BOOKS 列を参照するハンドル列の名前です。上記ステートメントのそれ以外の部分は、table、ならびに delete または delete の任意の同義語と同じ段落に現れる authorization を探す検索引き数の例です。

---

## DB2 クライアント / サーバー環境における DB2 テキスト・エクステンダー

5ページの図1 は、DB2 テキスト・エクステンダーを DB2 クライアント / サーバー環境にどのように組み込むのかを示しています。

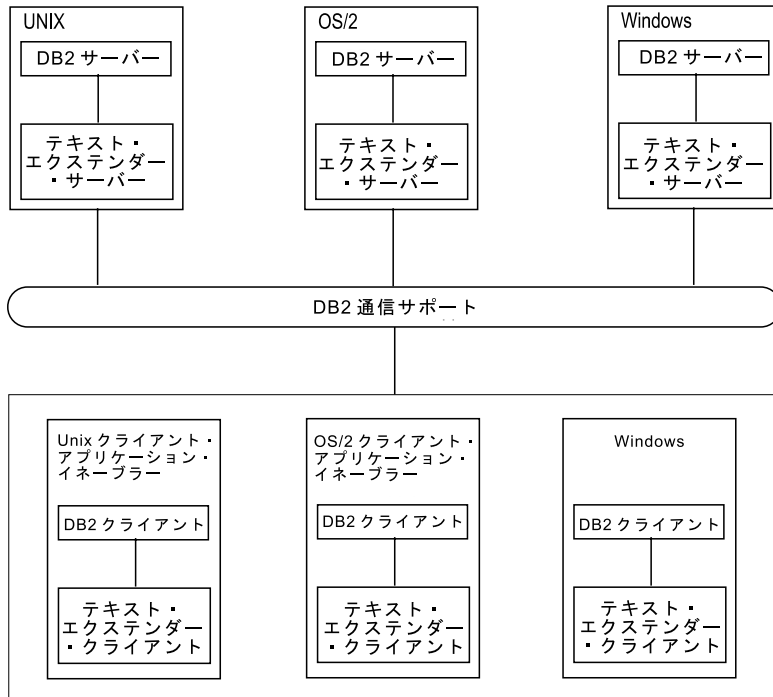


図1. DB2 クライアント / サーバー環境への DB2 テキスト・エクステンダーの統合

クライアント用の DB2 通信サポート・プロトコル (TCP/IP または NETBIOS など) のリストについては、該当するプラットフォーム用の DB2 概説およびインストール を参照してください。

DB2 テキスト・エクステンダーの主なソフトウェア部品は、DB2 サーバーと同じマシンにインストールされます。1 つの DB2 サーバー・インスタンスと共にインストールできる DB2 テキスト・エクステンダー・サーバーは 1 つだけです。

DB2 テキスト・エクステンダーのインストール内容は必要に応じて変更ことができ、以下のものを組み込むことができます。

- 図1 に示されている、1 つまたは複数の DB2 テキスト・エクステンダー・サーバー (UNIX には AIX(R)、SUN-Solaris、および HP-UX ワークステーションが含まれる)
- 1 つまたは複数のリモート DB2 テキスト・エクステンダー・サーバーにアクセスする AIX、SUN-Solaris、Windows NT、および Windows 2000 クライアント

- ローカルなサーバーを含み、リモート・サーバーへのアクセス権を持つ AIX クライアント

図2 は、代表的な DB2 テキスト・エクステンダー構成を示しています。クライアントから DB2 テキスト・エクステンダーを実行するには、まず 1 台の DB2 クライアントといくつかの DB2 テキスト・エクステンダー・ユーティリティをインストールしておかなければなりません。これらのユーティリティは、厳密にはクライアントではありませんが、DB2 テキスト・エクステンダー「クライアント」の構成要素となります。クライアントは、DB2 クライアント接続を介してサーバーと通信します。

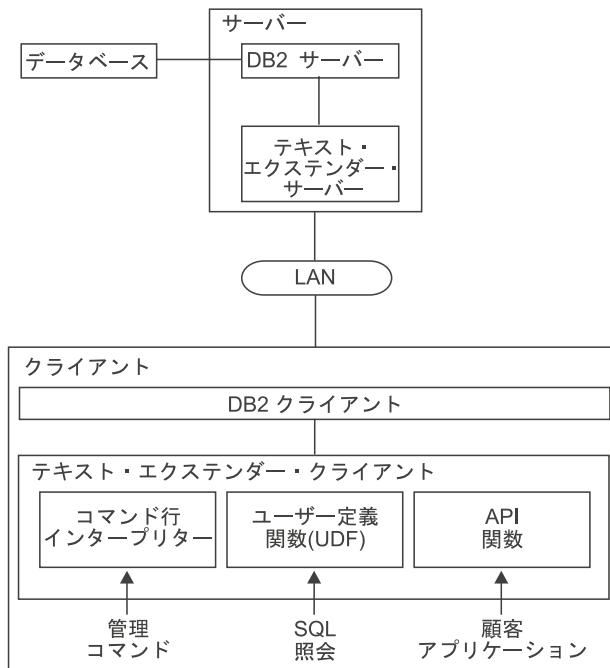


図2. DB2 テキスト・エクステンダー構成

DB2 テキスト・エクステンダーの主な構成要素は以下のとおりです。

- コマンド行インタープリター。** コマンドを使用して、検索されるテキストを列中に作成したり、テキスト索引を保守したりすることができます。
- SQL 関数。** これを SQL 照会に組み込んで、テキストの検索を行ったり、テキスト中に検索項目が出てくる回数などを検出したりすることができます。分かりやすくするために、この図では SQL 関数をクライアント上に示しています。それらは SQL 照会の一部として使用できるからです。実際には、SQL 関数はサーバー・インスタレーションの一部であり、サーバーで実

行されます。しかし、これらの関数は、DB2 テキスト・エクステンダー・クライアントをインストールしなくても任意の DB2 クライアントから使用できます。

- **アプリケーション・プログラミング・インターフェース (API)**。これは、テキスト検索を行ったり検索結果を表示するために C プログラムで呼び出すことのできる関数から成ります。

#### ヒント

DB2 テキスト・エクステンダー・クライアント・ユーティリティーには、テキスト準備関数、管理関数、および API が備わっています。(これらの関数はサーバー上で使用できます。) これらの関数を使用するには、DB2 テキスト・エクステンダー・クライアントをインストールしなければなりません。DB2 テキスト・エクステンダー・サーバーだけでテキスト準備および管理を行う場合は、DB2 テキスト・エクステンダー・クライアント・ユーティリティーは、API 関数を使用するクライアントだけにインストールすれば十分です。

クライアント側で、DB2 SQL ステートメントを使用する検索機能しか必要でない場合は、DB2 テキスト・エクステンダー・クライアントをインストールする必要はありません。すべての通信は DB2 によって処理され、DB2 テキスト・エクステンダー検索エンジンは、サーバーでのみ稼働します。

---

## 区分データベース環境での DB2 テキスト・エクステンダー

8ページの図3 は、DB2 テキスト・エクステンダーを DB2 区分データベース環境にどのように組み込むのかを示しています。

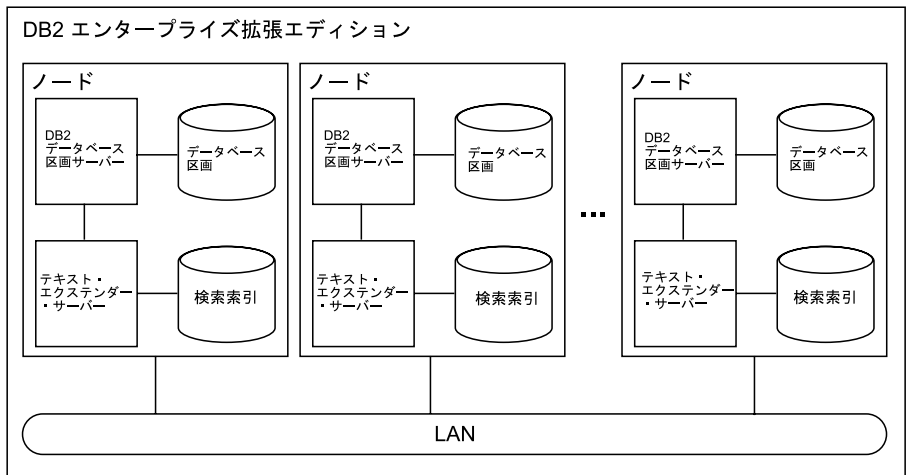


図3. DB2 区分データベース環境への DB2 テキスト・エクステンダーの組み込み

それぞれのデータベース区画サーバーは、ノードと呼ばれますが、別個のマシンに割り当てられることも、単一マシンに入れることもできます。別個のマシンにあるデータベース区画サーバーは、物理ノードと呼ばれます。単一マシンにあるものは、論理ノードと呼ばれます。DB2 テキスト・エクステンダーは、両方の構成をサポートします。

ノードグループは、1 つまたは複数のデータベース区画サーバーの指定されたサブセットです。DB2 テキスト・エクステンダー使用可能テーブルが多重区画ノードグループに割り当てられる場合には、DB2 テキスト・エクステンダーにより作成されるテキスト索引は、それに対応するテーブル区画と同じノードに入れます。これにより、索引付けのときにデータがノード間を移動する必要がなくなります。

DB2 テキスト・エクステンダー・サーバーは、ノードグループのすべての区画で稼働し、対応する区画の索引付けされる行を管理します。ユーザーは、1 つのデータベース区画を介して DB2 テキスト・エクステンダーと対話します。それぞれのデータベース区画は、DB2 テキスト・エクステンダー要求を出すために使用できます。要求は、関係するすべてのデータベース区画に経路指定されます。

---

## 第2章 インストール、設定、および保守

---

### AIX、Sun-Solaris、および HP-UX システムへのインストール

AIX、Sun-Solaris、および HP-UX 用の DB2 テキスト・エクステンダー バージョン 7 は、非区分データベース環境内で実行できます。AIX および Sun-Solaris では、DB2 エンタープライズ拡張エディションに備わっている区分データベースのサポートも利用できます。

以下にインストールのためのステップを示します。

1. 製品コンポーネントをターゲット・マシンにインストールします。
2. DB2 テキスト・エクステンダーのインスタンスを確立します。
3. DB2 テキスト・エクステンダー・サーバーを開始します。

#### インストール要件

DB2 テキスト・エクステンダーをインストールする前に、サーバーおよびクライアント・コンポーネントのための以下のインストール要件に適合していることを確認してください。

##### サーバー要件

サーバー・コンポーネントには、DB2 バージョン 7.1 (またはそれ以降) がインストールされている必要があります。

##### クライアント要件

クライアント・コンポーネントには、DB2 バージョン 7.1 (またはそれ以降) のパーソナル・エディション、DB2 バージョン 7.1 (またはそれ以降) のクライアント・アプリケーション・イネーブラー、または DB2 バージョン 7.1 (またはそれ以降) のソフトウェア開発者キットがインストールされている必要があります。

#### ステップ 1 (AIX 用): 製品コンポーネントのインストール

インストールの際に、smadmin と呼ばれるグループが自動的に作成されます。

1. ターゲット・クライアントまたはサーバー・マシンに root ユーザーとしてログインします。

## UNIX システムへのインストール

2. システム・マネージメント・インターフェース・ツール (SMIT) を使用して指示インターフェースから `installp` を実行するか、または `installp` コマンドを入力します。
3. `smit install_latest` コマンドを入力します。「ソフトウェアのインストールおよび保守ツール (Software Installation and Maintenance Tool)」メニューが開きます。
4. パッケージのロケーションを、「ソフトウェアの装置 / ディレクトリーを入力 (INPUT device / directory for software)」フィールドに入力します。
5. 「実行 (DO)」ボタンをクリックするか、または ENTER キーを押します。これにより、インストール・ディレクトリーが確定します。
6. クライアント・コンポーネントとサーバー・コンポーネントのどちらをインストールするかを、「インストールするソフトウェア (SOFTWARE TO INSTALL)」フィールドで指定します。
7. 「実行 (DO)」ボタンをクリックするか、または ENTER キーを押します。インストール・パラメーターを確定するためのプロンプトが表示されます。確定するには、ENTER キーを押します。
8. クライアントにインストールしたい辞書がある場合、それを「インストールするソフトウェア (SOFTWARE TO INSTALL)」フィールドに指定します。辞書を指定しない場合は、米国英語の辞書がインストールされます。

表1 には、DB2 テキスト・エクステンダーの辞書がリストされています。辞書をクライアントにインストールするには、「インストールするソフトウェア (SOFTWARE TO INSTALL)」フィールドに `db2tx_07_01.dic` と入力します。

表1. DB2 テキスト・エクステンダーの辞書

辞書	言語
<code>db2tx_07_01.dic.Ar_AA</code>	アラビア語
<code>db2tx_07_01.dic.Da_DK</code>	デンマーク語
<code>db2tx_07_01.dic.Nl_NL</code>	オランダ語
<code>db2tx_07_01.dic.En_GB</code>	英語 (英国)
<code>db2tx_07_01.dic.En_US</code>	英語 (米国)
<code>db2tx_07_01.dic.Fi_FI</code>	フィンランド語
<code>db2tx_07_01.dic.Fr_FR</code>	フランス語
<code>db2tx_07_01.dic.Fr_CA</code>	フランス語 (カナダ)
<code>db2tx_07_01.dic.De_DE</code>	ドイツ語
<code>db2tx_07_01.dic.De_CH</code>	ドイツ語 (スイス)



表 1. DB2 テキスト・エクステンダーの辞書 (続き)

辞書	言語
db2tx_07_01.dic.Iw_IL	ヘブライ語
db2tx_07_01.dic.Is_IS	アイスランド語
db2tx_07_01.dic.It_IT	イタリア語
db2tx_07_01.dic.No_NO	ノルウェー語
db2tx_07_01.dic.Pt_PT	ポルトガル語
db2tx_07_01.dic.Ru_RU	ロシア語
db2tx_07_01.dic.Es_ES	スペイン語およびカタロニア語
db2tx_07_01.dic.Sv_SE	スウェーデン語
db2tx_07_01.dic.Th_TH	タイ語

9. ログアウトします。

### ステップ 1 (Sun-Solaris 用): 製品コンポーネントのインストール

インストールの際に、smadmin と呼ばれるグループが自動的に作成されます。

1. ターゲット・クライアントまたはサーバー・マシンに root ユーザーとしてログインします。
2. 以下のように、tar を使用してフィックスパック・ファイルからインストール可能パッケージを取り出します。

```
tar -xvf IBMdb2tx.tar
```

3. pkgadd を使用して、それらのファイルをディレクトリーからターゲット・マシンに転送します。これを行うには、以下の 2 つの方法のいずれかを選択できます。
  - DB2 テキスト・エクステンダーのクライアントおよびサーバーを米国英語辞書と共にインストールする。

```
pkgadd -d . db2txclnt db2txsrv db2txcom db2txmsgs db2txmsc db2txdius
```

- DB2 テキスト・エクステンダーのクライアントまたはサーバーのどちらをインストールしたいのか、およびどの辞書をインストールしたいのかを指定する。

```
./db2tx_install
```

4. ログアウトします。

### ステップ 1 (HP-UX 用): 製品コンポーネントのインストール

インストールの際に、smadmin と呼ばれるグループが自動的に作成されます。

## UNIX システムへのインストール

1. ターゲット・クライアントまたはサーバー・マシンに root ユーザーとしてログインします。
2. 以下のように、tar を使用してフィックスパック・ファイルからインストール可能パッケージを取り出します。

```
tar -xvf b2tx.tar
```

3. swinstall プログラムを実行します。

```
swinstall
```

これにより、「ソフトウェアの選択 (Software Selection)」ウィンドウおよび「ソースの指定 (Specify Source)」ウィンドウが開きます。

4. 必要であれば、「ソース・ホスト名 (Source Host Name)」を変更します。
5. 「ソース保管パス (Source Depot Path)」フィールドの値として、IBMdb2tx を使用している tar ディレクトリーを入力します。
6. 「ソフトウェア・ビューの変更 (Change Software View)」フィールドで、「製品 (Products)」を選択します。
7. 「OK」を選択して「ソフトウェアの選択 (Software Selection)」ウィンドウに戻ります。そこには、インストール可能なソフトウェアがリストされています。
8. ここで、以下の 1 つを実行してください。
  - すべての辞書、および提供されているすべての言語の文書と共に DB2 テキスト・エクステンダーをインストールするには、「DB2TX 製品 (DB2TX Product)」を強調表示します。
  - DB2 テキスト・エクステンダーの一部をインストールするには、以下のように行います。
    - a. 「アクション (Actions)」メニューから、「項目のオープン (Open Item)」を選択します。
    - b. インストールしたいファイル・セットごとに、それを強調表示してから「アクション (Actions)」メニューの「インストールのためのマークを選択 (Choose Mark for Install)」を選択します。以下のようにして、DB2 テキスト・エクステンダー・サーバーまたはクライアントのどちらをインストールするかを選択できます。

```
Text Extender Server - DB2TX.db2txsrv
Text Extender Client - DB2TX.db2txc1nt
```
    - c. インストールしたい辞書がある場合、それを選択します。13ページの表2 には、DB2 テキスト・エクステンダーの辞書がリストされています。

表 2. DB2 テキスト・エクステンダーの辞書

辞書	言語
db2txdiaa	アラビア語
db2txdidk	デンマーク語
db2txdinl	オランダ語
db2txdigb	英語 (英国)
db2txdius	英語 (米国)
db2txdifi	フィンランド語
db2txdifr	フランス語
db2txdica	フランス語 (カナダ)
db2txdide	ドイツ語
db2txdich	ドイツ語 (スイス)
db2txdiil	ヘブライ語
db2txdiis	アイスランド語
db2txdiit	イタリア語
db2txdino	ノルウェー語
db2txdipt	ポルトガル語
db2txdies	ロシア語
db2txdiru	スペイン語およびカタロニア語
db2txdise	スウェーデン語
db2txdith	タイ語

9. 「アクション (Actions)」メニューから「インストール (Install)」を選択して、製品のインストールを開始し、「インストール分析 (Install Analysis)」ウィンドウをオープンします。
10. 「状況 (Status)」フィールドに「作動可能 (Ready)」メッセージが表示されたら、「インストール分析 (Install Analysis)」ウィンドウで「OK」ボタンを選択します。
11. 「確認 (Confirmation)」ウィンドウで「はい (Yes)」ボタンを選択して、DB2 (HP-UX 版) のソフトウェア製品をインストールしたいことを確認します。
12. 「インストール (Install)」ウィンドウを表示すると、ソフトウェアがインストールされる際のデータ処理を読み取ることができます。「状況 (Status)」フィールドに「作動可能 (Ready)」と表示されて、「注釈 (Note)」ウィンドウが開けば、インストールは完了です。

## UNIX システムへのインストール

swinstall プログラムはファイル・セットをロードして、ファイル・セット用の制御スクリプトを実行します。

HP-UX オペレーティング・システムには、swinstall についての詳細なヘルプが備わっています。ヘルプを表示するには、以下のようにタイプします。

```
man swinstall
```

13. ログアウトします。

### ステップ 2: DB2 テキスト・エクステンダーのインスタンスを確立する

1. root ユーザーとして活動状態であることを確認してください。
2. AIX では、以下のように入力します。

```
cd /usr/lpp/db2ext_07_01/instance
```

Solaris および HP-UX では、以下のように入力します。

```
cd /opt/IBMDB2TX/V7.1/instance
```

3. 既存のインスタンスがある場合、以下のように入力します。

```
./txiupdt instanceid db2instanceid
```

これにより、インスタンスが新規のファイルによって更新されます。

既存のインスタンスがない場合、以下のように入力して DB2 テキスト・エクステンダーのインスタンスを新規に作成します。

```
./txicrt instanceid db2instanceid
```

ここで、*instanceid* は既存の DB2 インスタンスのユーザー ID、または DB2 インスタンスの ID に関連付けるユーザー ID です。

txicrt を実行すると、`/u/instanceid/db2tx` ディレクトリーが作成されます。`/u/instanceid/db2tx` の下に追加のファイルまたはディレクトリーを作成しないでください。それらのファイルは、インスタンスが削除されるときに失われてしまいます。

区分データベース環境を使用する場合、追加のパラメーターが必要です。詳細については、179ページの『TXICRT』を参照してください。

4. ログアウトします。

### ステップ 3: DB2 テキスト・エクステンダー・サーバーの開始

1. *instanceid* としてログインします。
2. DB2 を停止してから、再始動します。

3. DB2 テキスト・エクステンダー・サーバーを開始するには、以下のように入力します。

```
txstart
```

4. ログアウトします。

---

## Windows 95、Windows 98、Windows NT、および Windows 2000 へのインストール

DB2 テキスト・エクステンダーは、DB2 エクステンダーのパッケージの一部としてインストールされます。

### サーバー要件

DB2 エクステンダー用のサーバー・コンポーネントは、Windows NT または Windows 2000 サーバーにインストールできます。

非区分データベース環境で実行するサーバー・コンポーネントには、DB2 バージョン 7.1 (またはそれ以降) がインストールされている必要があります。

区分データベース環境で実行するサーバー・コンポーネントには、DB2 エンタープライズ拡張エディション バージョン 7.1 (またはそれ以降) がインストールされている必要があります。

### クライアント要件

DB2 エクステンダー用のクライアント・コンポーネントは、Windows 95、Windows 98、Windows NT、または Windows 2000 にインストールできます。

クライアント・コンポーネントには、DB2 バージョン 7.1 (またはそれ以降) のパーソナル・エディション、DB2 バージョン 7.1 (またはそれ以降) のクライアント・アプリケーション・イネーブラー、または DB2 バージョン 7.1 (またはそれ以降) のソフトウェア開発者キット (Windows 版) がインストールされている必要があります。

### DB2 エクステンダーのインストール

DB2 エクステンダーを Windows にインストールするには、以下のように行います。

1. CD-ROM をクライアント・ワークステーション上の CD-ROM ドライブに挿入します。

## Windows システムへのインストール

2. CD 自動開始機能が使用可能になっている場合、DB2 エクステンダー LaunchPad が自動的に開始します。そうでない場合、タスクバー上の「スタート」ボタンをクリックします。「ファイル名を指定して実行」を選択します。開いたフィールドに、

```
d:¥db2ext¥winnt95¥setup
```

とタイプします。ここで、“d” は CD-ROM ドライブの文字です。

3. LaunchPad から、DB2 エクステンダー ReadMe ファイルを表示できます。インストールの前提条件を表示して読むか、または製品のインストールを実行します。
4. 左側の「インストール」ボタンを選択すると、インストールが開始します。
5. 最初のウィンドウで、インストールする DB2 エクステンダーのバージョンを、「EEE のクライアント / サーバー (Client/Server for EEE)」、「非 EEE のクライアント / サーバー (Client/Server for non-EEE)」、または「クライアントのみ (Client only)」から選択します。
6. 次のウィンドウで、エクステンダーをインストールするウィンドウを選択します。
7. その後、インストールのタイプを「通常 (Typical)」、「カスタム (Custom)」、または「コンパクト (Compact)」から選択します。

「**通常 (Typical)**」インストールには、DB2 エクステンダーのクライアント / サーバー、サンプル・ファイル、SDK、米国英語のオンライン文書、および米国英語の DB2 テキスト・エクステンダー辞書が含まれます。

「**コンパクト (Compact)**」インストールでは、クライアント / サーバーおよび米国英語の DB2 テキスト・エクステンダー辞書だけをインストールします。

「**カスタム (Custom)**」インストールでは、以下のコンポーネントから選択することができます。

- DB2 エクステンダーを区分データベース環境 (DB2 エンタープライズ拡張エディション) にインストールする場合、「**DB2 エクステンダー・サーバー (EEE) (DB2 Extenders Server (EEE))**」を選択します。これにより、「**DB2 エクステンダー・クライアント (DB2 Extenders Client)**」コンポーネントもインストールされます。
- DB2 エクステンダーを非区分データベース環境にインストールする場合、「**DB2 エクステンダー・サーバー (非 EEE) (DB2 Extenders Server (non-EEE))**」を選択します。これにより、「**DB2 エクステンダー・クライアント (DB2 Extenders Client)**」コンポーネントもインストールされます。

- DB2 エクステンダーのクライアント環境をインストールする場合、「**DB2 エクステンダー・クライアント (DB2 Extenders Client)**」を選択します。
- DB2 エクステンダーのアプリケーションを開発するためにヘッダー・ファイルおよびライブラリーをインストールしたい場合、「**DB2 エクステンダー SDK (DB2 Extenders SDK)**」を選択します。
- サンプル・プログラムおよびサンプル・メディア・ファイルをインストールしたい場合、「**サンプル・ファイル (Sample Files)**」を選択します。
- エクステンダー資料をソフトコピー形式でインストールしたい場合、「**オンライン文書 (Online Documentations)**」を選択します。この項目を選択した後で、「説明 (Description)」セクションの「**変更 (Change)**」ボタンをクリックすると、使用可能な言語に特定の資料を表示することができます。そのリストから、インストールしたいものを選択してください。
- DB2 テキスト・エクステンダー用に選択した言語のための言語サポートが必要な場合、「**テキスト・エクステンダー辞書 (Text Extender Dictionaries)**」を選択します。この項目を選択した後で、「説明 (Description)」セクションの「**変更 (Change)**」ボタンをクリックすると、使用可能な言語に特定の辞書を表示することができます。そのリストから、インストールしたいものを選択してください。米国英語は常に必須です。

選択したコンポーネントのために十分なディスク・スペースがない場合、「**戻る (Back)**」を押してインストール・ディレクトリーを再選択してください。そうでない場合は、「**次へ (Next)**」を押します。

8. エクステンダー・サーバー (EEE) を区分データベース環境にインストールする場合、インスタンスを所有するマシン上で操作しているのであれば、TCP/IP ポートの範囲を指定するようにプロンプトが出されます。現在のDB2 インスタンスに関連したすべてのノードに同じ共用名が存在しなければなりません。

インスタンスを所有しないマシン上に DB2 エクステンダーをインストールする場合、インスタンスを所有するマシン名を入力するようにプロンプトが出されます。TCP/IP ポート範囲およびデータ経路情報は、インスタンスを所有するマシンから入手できます。

9. 「**次へ (Next)**」ボタンをクリックして、選択したコンポーネントをインストールします。

インストール・プログラムが開始すると、それはワークステーションがDB2 エクステンダーのソフトウェア要件に適合するかを調べます。

## Windows システムへのインストール

10. ワークステーションをシャットダウンしてから、再始動してください。

### インスタンスの作成および管理

DB2 エクステンダー・サーバーの複数のインスタンスを作成して使用することができます。Windows 用の DB2 サーバーに複数のインスタンスを作成した場合、複数のインスタンスを作成してください。DB2 エクステンダー・サーバーの各インスタンスは、Windows 用の DB2 サーバーのインスタンスに関連しています。

Windows 用の DB2 エクステンダーを非区分データベース環境にインストールするとき、DB2 という名前の省略時の DB2 エクステンダー・インスタンスが作成されます。

Windows 用の DB2 エクステンダーを区分データベース環境にインストールするとき、DB2MPP という名前の省略時の DB2 エクステンダー・インスタンスが作成されます。区分データベース環境では、DB2 テキスト・エクステンダーのために作成される省略時のインスタンスはありません。インスタンスを作成するには、コマンド `txicrt` を使用します。詳細については、23ページの『DB2 テキスト・エクステンダー・インスタンスの作成』を参照してください。

### インストール検査のためにサンプル・データベースを準備する

DB2 テキスト・エクステンダーは、サンプル・データベースを作成するユーティリティを備えています。これらは、インストール直後に DB2 テキスト・エクステンダーの検索機能をテストするため、またインストール後、随時に汎用テストを行うために迅速にテキストを作成するのに便利です。

これらのユーティリティは、以下のディレクトリーに入っています。

- OS/2(R)、Windows NT、および Windows 2000 用: `DMBMMPATH¥samples`  
ここで、`DMBMMPATH` はインストール・パスです。
- AIX、HP-UX、および SUN-Solaris 用: `$HOME/db2tx/samples`

サンプル・データベースを作成するためには、サーバーまたはクライアントのワークステーションで次のようにします。

1. オペレーティング・システムのプロンプトに応じて、以下のコマンドを実行する。

```
TXVERIFY database-name [user-id] [password]
```

このコマンドは、サンプル・データベースを作成し、使用可能にします。



- オペレーティング・システムのプロンプトに応じて、以下のコマンドを実行する。

```
TXSAMPLE database-name [user-id] [password]
```

クライアントのワークステーションから処理している場合に限り、ユーザー ID とパスワードが必要です。このコマンドは、以下の処理を行います。

指定されたデータベースに接続する。

表 db2tx.sample を作成する。

サンプルの英語文書をインポートして、表を埋める。

次の索引タイプを指定して、表内のテキスト列を使用可能にする。

一致語句

正規化された一致語句

基本語形

基本語形 (セクション指定可能)

Ngram

Ngram (大文字小文字の区別あり)

テキスト索引が作成されるのを待つ。

---

## 構成

このセクションでは、DB2 テキスト・エクステンダー環境変数と構成情報について説明します。これらを使用して、DB2 テキスト・エクステンダーに必要なほとんどのパラメーターに対して省略時値を指定します。

### 環境変数

環境変数は、環境パラメーターの省略時値を設定する変数です。環境変数の現行設定値を表示するには、112ページの『環境変数の設定値の表示』で説明されている GET ENVIRONMENT コマンドを使用してください。

#### **DB2INSTANCE**

DB2 インスタンス名。

#### **DB2DBDFT**

省略時のデータベース名。データベース名が指定されないときに想定される DB2 データベースの名前です。

#### **DB2TX\_INSTOWNER**

DB2 テキスト・エクステンダー・インスタンス名。これは、インスタンスを所有するユーザーのログイン名です。これが必要なのは、DB2 テキスト・エクステンダー・クライアントが UNIX サーバーに接続している場合です。

### DB2TX\_INSTOWNERHOMEDIR

インスタンス所有者のホーム・ディレクトリー (UNIX のみ)。

**Windows 環境の場合、** DB2 テキスト・エクステンダー環境変数はシステム・プロパティ・ページでシステム変数として指定されます。これらの環境変数はインストール時にユーザーのシステム設定値に追加されます。

**OS/2 環境の場合、** DB2 テキスト・エクステンダー環境変数はインストール時に CONFIG.SYS ファイルに追加されます。

**UNIX 環境では、** DB2DBDFT 以外の DB2 テキスト・エクステンダー環境変数は、以下のプロファイル・シェル・スクリプト内に指定されます。

C シェルの場合、db2txcshrc

Korn シェルおよび Bourne シェルの場合、db2txprofile

インスタンスを作成するとき、ファイル db2txprofile または db2txcshrc が .profile にコピーされます。

インスタンスを作成するとき、db2txprofile がインスタンス所有者の .profile および db2profile に追加されます。シェル・スクリプトは \$DB2TX\_INSTOWNERHOMEDIR/db2tx に保管されます。

以下の環境変数も必要です。

**LANG** 言語の設定値を指定します。

**NLSPATH** DB2 テキスト・エクステンダーのメッセージ・ファイルへのパスを指定します。

**DB2ENVLIST** DB2 から DB2 テキスト・エクステンダーの関数に渡される環境変数のリストです。

**RAHUSER** DB2 コマンド db2a11 で使用されるインスタンス所有者の名前を指定します。DB2 EEE 環境だけで使用されます。

インスタンス所有者の場合、これらの環境変数はインスタンスの作成時に自動的に設定されます。インスタンス所有者でない場合は、これらの環境変数を手操作で設定せずに、インスタンス所有者の db2tx プロファイルを組み込んでください。

**ヒント**

環境変数を変更するときは常に、その変更を有効にするために、DB2 および DB2 テキスト・エクステンダーを停止してから再始動しなければなりません。

**テキスト構成設定値**

各データベースには、以下を含むテキスト構成設定値があります。

- テキスト特性
- 索引特性
- 処理特性

データベースを DB2 テキスト・エクステンダーに対して使用可能にするとき、これらの設定値が設定されます。ENABLE DATABASE コマンドは、コマンドで指定する設定値、またはここに説明する初期設定値のどちらかを使用します。これらの省略時設定値を表示または変更することができます。113ページの『テキスト構成設定値の表示』と 22ページの『テキスト構成の変更』を参照してください。

**テキスト特性**

33ページの『第4章 必要に応じた検索の計画』では、DB2 テキスト・エクステンダーでサポートされる文書形式、言語、および CCSID について説明します。これらの省略時値は、さまざまなコマンドで必要となります。

**FORMAT**            初期設定: TDS  
**LANGUAGE**        初期設定: データベースに対して設定される言語  
**CCSID**             初期設定: データベースに対して設定される CCSID

**索引特性**

**DIRECTORY**        索引を保管するためのディレクトリー。  
                           UNIX システムの初期設定:  
                           *DB2TX\_INSTOWNERHOMEDIR/db2tx/indexes*  
                           OS/2、Windows NT、および Windows 2000 の初期設定:  
                           *DMBMMPATH¥instance¥instance-name¥db2tx¥indexes*。ここで、  
                           *DMBMMPATH* は、DB2 テキスト・エクステンダーがインストール

## 構成

ールされているパスです。Windows クライアントの場合、サーバー・マシン上にあるディレクトリーの名前に **DIRECTORY** を設定する必要があります。

**INDEXTYPE** 使用される索引タイプ。詳しくは 45ページの『検索のタイプ』を参照してください。

初期設定: NGRAM

### UPDATEFREQ

周期的な索引更新の頻度。詳しくは 51ページの『索引の更新』を参照してください。

初期設定: NONE

## 処理特性

### UPDATEINDEX

最初の索引更新の時期を、使用可能化ステップ中にすぐに行う、更新頻度の設定に基づいて後に行う (NOUPDATE)、または明示的な UPDATE INDEX コマンドの結果として行うの、いずれかに決定する設定。

初期設定: UPDATE

### COMMITCOUNT

いくつかの挿入ステートメントまたは更新ステートメントが出されると、DB2 テキスト・エクステンダーが DB2 コミット用ステートメントを出すかを定めるための設定です。76ページの『ラージ表でのテキスト列の使用可能化』を参照してください。

初期設定: 0

## テキスト構成の変更

DB2 テキスト・エクステンダーが初めてインストールされるときに、省略時値がテキスト構成に設定されます。現行のテキスト構成値を表示するには、113ページの『テキスト構成設定値の表示』を参照してください。

索引の作成時に省略時値として使用されるテキスト構成を変更するには、次のように入力します。

```
db2tx CHANGE TEXT CFG USING settings
```

例:

今後の索引のための省略時索引タイプと省略時ディレクトリーを変更するには次のようにします。

```
db2tx CHANGE TEXT CONFIGURATION USING
      INDEXTYPE    precise
      INDEXTYPE    normalized
      DIRECTORY    /myfs/indexes
```

100 個以上のテキスト文書が待ち行列にある場合に、月曜から金曜の 12:00 または 15:00 に索引が更新されるように、索引の更新頻度の省略時値を変更するには、次のようにします。

```
db2tx CHANGE TEXT CONFIGURATION USING
      UPDATEFREQ min(100) d(1,2,3,4,5) h(12,15) m(00)
```

索引の周期的な更新を停止するには次のようにします。

```
db2tx CHANGE TEXT CONFIGURATION USING
      UPDATEFREQ none
```

---

## DB2 テキスト・エクステンダー・サーバーの設定および保守

新規にインストールした DB2 テキスト・エクステンダーに関する次のステップは、DB2 テキスト・エクステンダー・サーバーを DB2 テキスト・エクステンダー・クライアントが使用できるように設定することです。通常これは、DB2 テキスト・エクステンダー・サーバー管理者によって行われ、以下の作業が含まれます。

1. DB2 テキスト・エクステンダー・インスタンスの作成
2. DB2 テキスト・エクステンダー・サーバーの開始

DB2 テキスト・エクステンダー・サーバーを保守するための他のタスクは、以下のとおりです。

1. DB2 テキスト・エクステンダー索引および使用可能なデータベースのバックアップおよび復元
2. 障害のトレース

コマンド構文については、177ページの『第10章 サーバー向け管理コマンド』を参照してください。

### DB2 テキスト・エクステンダー・インスタンスの作成

DB2 テキスト・エクステンダーを使った作業を開始する前に、DB2 テキスト・エクステンダー・インスタンスを作成する必要があります。ユーザーが作

## サーバーの設定および保守

成するインスタンスはいずれも分離した環境を提供し、ここで索引を別々のディレクトリーに保管して保守することができます。

**UNIX 単一ノード環境にインスタンスを作成するには、** 次のように入力します。

```
txicrt instance-name db2-instance-name
```

ここで、*instance-name* はインスタンス所有者のユーザー ID で、*db2-instance-name* はそれに関連付けられる DB2 インスタンスの名前です。

**Windows NT および Windows 2000** では、インストール中に省略時のインスタンスが作成されます。追加のインスタンスを作成するには、次のように入力します。

```
txicrt db2-instance-name
```

複数の環境でインスタンスを作成するには、179ページの『TXICRT』を参照してください。

**UNIX システムでインスタンスを除去するには、** 以下のように入力します。

```
txidrop instance-name db2-instance-name
```

**Windows NT および Windows 2000** でインスタンスを除去するには、以下のように入力します。

```
txidrop db2-instance-name
```

どのインスタンスを処理するかを決めるには、以下のようにして適切な環境変数を設定します。

```
set DB2INSTANCE = instance-name
```

その後、DB2 および DB2 テキスト・エクステンダーを停止してから、再始動します。

## DB2 テキスト・エクステンダー・サーバーの開始および停止

**DB2 テキスト・エクステンダー・サーバーを開始するには、** インスタンス所有者のユーザー ID でログオンした後 (UNIX システムのみ) で、次のように入力します。

```
txstart
```

**DB2 テキスト・エクステンダー・サーバーの状況を表示するには、** 以下のように入力します。

```
txstatus
```

**DB2 テキスト・エクステンダー・サーバーを停止するには、以下のように入力します。**

```
txstop
```

#### UNIX ユーザー向けのヒント

DB2 テキスト・エクステンダー・サーバーを開始することに問題がある場合、共用資源をクリーンアップしてみてください。183ページの『TXIPCLEAN (UNIX のみ)』を参照してください。

### 索引および使用可能なデータベースのバックアップおよび復元

使用可能にされたデータベースおよび DB2 テキスト・エクステンダーが作成したテキスト索引をバックアップおよび復元することができます。

バックアップするには、

1. どの表が DB2 テキスト・エクステンダーにより使用可能にされたかを調べます。次のように入力します。

```
db2tx GET STATUS
```

2. データベースにより使用される索引ディレクトリーの名前を見つけます。次のように入力します。

```
db2tx GET INDEX SETTINGS table-name
```

3. DB2 テキスト・エクステンダー・サーバーを停止します。次のように入力します。

```
TXSTOP
```

4. 索引ディレクトリーおよびそれらのサブディレクトリー `index` および `work` をバックアップします。
5. ファイル `desmastr.dat` をバックアップします。これは以下の場所にあります。

```
instance_owner_home_directory/db2tx/txins000 (UNIX)
```

または

```
DMBMPATH¥db2tx¥txins000 (OS/2、Windows NT、および Windows 2000)
```

区分データベースには、いくつかのインスタンス・ディレクトリー `txins000`、`txins001`、... があり、個々のディレクトリーでファイルをバックアップするよりも、.../db2tx ディレクトリー全体をバックアップする方が便利であることに注意してください。

6. DB2 テキスト・エクステンダー・サーバーを再始動します。

## サーバーの設定および保守

TXSTART

復元するには、

1. DB2 テキスト・エクステンダー・サーバーを停止します。

TXSTOP

2. 既存の desmastr.dat ファイルを保管します。
3. desmastr.dat ファイルのバックアップ・コピーを復元します。
4. 索引ディレクトリーのバックアップ・コピーを前と同じパスに復元します。
5. DB2 テキスト・エクステンダー・サーバーを再始動します。

TXSTART

### 障害のトレース

IBM 担当員にエラーを報告する必要がある場合は、エラー箇所を突き止めるのに使用できる情報をファイルに書き込むために、トレース機能に切り替えるよう要求される場合があります。このトレース機能を使用するのは、IBM サポート・センターの担当員またはお客様の技術サポート担当者からの指示がある場合だけにしてください。

トレース機能に切り替えると、システム・パフォーマンスに影響するので、この機能はエラー条件が発生した場合だけ使用するようになります。

トレース機能をオンに切り替えるには、次のように入力します。

```
txtrace on [options]
```

構文、事象のリスト、および構成要素は、195ページの『TXTRACE』に記載してあります。また、これ以外のオプションも記載してあります。

トレース・レコードの ID に基づいて、それらの各レコードの受け入れまたは排除を行う「マスク」を指定することによって、トレース機能にフィルターをかけることができます。省略時には、すべてのトレースが行われます。

マスクは、2.2-6.1,3.\* のように、ピリオドで区切られた 4 つの部分からなります。ここで、

- 2** DB2 UDB テキスト・エクステンダーを示します。
- 2-6** 事象 ID が 2 から 6 までの項目だけを含めます。
- 1,3** 構成要素 1 と 3 によって報告される事象だけを含めます。
- \*** 構成要素のすべての関数を含めます。



特定の重大度より低いシステム・エラーは排除することができますし、トレース・バッファがいっぱいになったら、最初または最後いずれのレコードを保持するかを指定できます。

エラーを複製して、そのトレース情報を 2 進数形式でダンプ・ファイルに書き込むには、次のように入力します。

```
txtrace dump dump-filename
```

形式化されたダンプ・ファイルを生成するには、次のように入力します。

```
txtrace format dump-filename formatted-filename
```

また、トレース機能がオンであるときに、共用メモリーから定様式ファイルへ直接トレース情報を次のように入力することもできます。

```
txtrace format > formatted-file
```

トレース情報をファイルに書き込んだら、トレース機能をオフにします。

```
txtrace off
```



---

## 第3章 はじめに

この章は、テキストを検索可能にするための基本的な点を理解するのに役立ちます。この章では、稼働中の DB2 テキスト・エクステンダー・システムで作業を行っていることを前提としています。つまり、システムのインストールと構成が済んでおり、DB2 テキスト・エクステンダー・インスタンスの作成と開始も済んでいるものとします。

### ヒント

この章で解説するのは、テキストを検索可能にするための基本的な点だけです。自分のテキストを検索用に準備する前に、63ページの『テキストを検索可能にする前の準備』をお読みください。

テキストを検索可能にする方法を理解するためには、以下の 2 つの方法があります。

- 簡単な例を通して、コマンドを手入力しながら一通りの手順を実行してみる方法
- サンプルの索引付けおよび検索ウィザードを実行する方法

---

### テキストを検索可能にする簡単な例

1. **Windows** システムの場合、**DB2** コマンド行プロセッサを実行する  
オペレーティング・システムのプロンプトで、以下を入力します。  
`=>db2 cmd`
2. **DB2** テキスト・エクステンダーのコマンド行プロセッサを開始する  
UNIX システムの場合、オペレーティング・システムのプロンプトで以下を入力します。  
`=>db2tx`  
  
Windows システムの場合、DB2 プロンプトで以下を入力します。  
`db2=>db2tx`
3. **データベースに接続する**  
検索可能にしたいテキストを含むデータベースを選択します。データベースに接続するには、以下を入力します。

## はじめに

```
db2tx=>CONNECT TO mydatabase
```

### 4. データベースをテキスト検索に使用できるようにする

接続されたデータベースを使用可能にするには次のコマンドを入力します。

```
db2tx=>ENABLE DATABASE
```

### 5. テキスト表をテキスト検索に使用できるようにする (任意選択)

テキスト表を使用可能にする必要があるのは、表全体のために単一の索引を作成したい場合だけです。

#### 索引を 1 つにするか、それとも複数にするか

49ページの『表に対する 1 つまたは複数のテキスト索引の作成』では、表を検索可能にするにあたり、表全体に 1 つのテキスト索引を作成する方法と、複数の索引 (各テキスト行に 1 つずつ) を作成する方法があることが説明されています。

- 表全体のために 1 つの索引を作成するには、この時点で `ENABLE TEXT TABLE` を実行して空の索引を作成してから、`ENABLE TEXT COLUMN` (次のステップを参照) を複数回 (テキスト列ごとに 1 回) 実行して、その 1 つの索引にデータを入れます。
- テキスト列ごとに別個の索引を作成する場合は、表全体のための索引を作成するステップを省略し (つまり `ENABLE TEXT TABLE` の実行を省略する)、`ENABLE TEXT COLUMN` を複数回実行することにより、テキスト列ごとに 1 つの索引を作成して、データを入れます。

この例では、表全体のために単一の索引を作成するのではなく、以下の手順に従って、テキスト列ごとに 1 つの索引を作成します。

### 6. テキスト列をテキスト検索で使用できるようにする

以下のコマンドを入力して、DB2 テキスト・エクステンダーが表 `db2tx.sample` 内のテキスト列 `mycolumn` を検索できるようすると共に、このコマンドが作成するハンドル列に名前 `myhandle` を割り当てます。

```
db2tx=>ENABLE TEXT COLUMN db2tx.sample mycolumn HANDLE myhandle
```

このコマンドは、テキスト索引を作成します。索引付けされる文書のタイプおよび索引特性には、省略時値が使用されます。

### 7. 作成する索引の状況を調べる

次のように入力します。

```
db2tx=>GET INDEX STATUS db2tx.sample HANDLE myhandle
```

8. **DB2 テキスト・エクステンダーのコマンド行プロセッサを終了する**  
次のように入力します。

```
db2tx=>QUIT
```

9. **DB2 のコマンド行プロセッサを開始する**

オペレーティング・システムのプロンプトで、以下を入力します。

```
=>db2
```

10. **テキストを検索する**

これで文書を検索できるようになりました。以下の **SELECT** コマンドを実行すると、索引付けしたテキスト内の **searchterm** のすべてのオカレンスが見つかります。

```
db2=>SELECT COUNT(*)
        FROM sample
        WHERE DB2TX.CONTAINS (myhandle,'"searchterm"') = 1
```

---

## 索引付けおよび検索のためのサンプル・ウィザード

選択した表または列に対して索引を作成する方法、およびそれらの索引に対して検索を実行する方法の例として、索引付けおよび検索ウィザードが提供されています。このサンプル・ウィザードは、DB2 テキスト・エクステンダー検索機能により表の選択された列に対して検索を実行するために必要なステップをガイドします。

サンプル・ウィザードを開始するには、次のように入力します。

```
txwizard
```

サンプル・ウィザードで提供される Java ソース・コードを使用すれば、テキストの準備と検索を行う独自のアプリケーションを作成したり、Java 環境で DB2 テキスト・エクステンダーの機能を使用し呼び出す方法について理解したりすることができます。

AIX では、ウィザードの Java ソース・コードは、次のディレクトリーに保管されています。

```
/usr/lpp/db2tx_06_01/samples/wizard
```

Windows NT および Windows 2000 では、次のディレクトリーに保管されています。

```
%DMBMPATH%\samples\wizard
```

はじめに

---

## 第4章 必要に応じた検索の計画

63ページの『第5章 テキストを検索可能にする』で説明されているステップを開始する前に、以下の点を調べる必要があります。

- 使用する文書の形式とコード・ページ、および使用している言語
- コード・ページの問題を回避する方法
- 必要な検索機能の種類
- 必要なディスク・スペース
- テキスト索引とは何か、および 1 つの DB2 表に共通の索引を 1 つ設ける方法と、表列ごとに別個の索引を設ける方法のどちらを採用するか

この章では、上記の情報を収集する理由と方法を説明します。

索引のタイプは、基本語形、一致語句、および Ngram から選択します。どの索引タイプを選択するかは重要なことです。たとえば、索引のタイプとして基本語形を選択すると、検索項目の変化形や同義語を検索できます。また、索引のタイプによって索引付けのパフォーマンスや索引のサイズが異なってきます。1 つのテキスト列について、索引タイプの異なる索引をいくつか作成すれば、複数の索引タイプの検索機能を使用することもできます。

---

### テキスト文書を索引付けする理由

高速情報検索システムでは、テキスト文書全体を順次走査するわけではありません。これでは非常に時間がかかります。そこで、あらかじめ作成したテキスト索引で操作を行います。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から成るものと考えことができ、各用語は、その用語を含む文書に関する情報と一緒に保管されます。

テキスト索引には関係のある情報だけが含まれ、「and」、「of」、および「which」などの重要でない語は索引付けされません。(Ngram 索引では停止ワードのフィルター処理は行われません。) DB2 テキスト・エクステンダーは、停止ワードと呼ばれる語のリストを使用して、このような語が索引付けされないようにします。検索システムは、要求された用語を索引中で検索し、その用語を含んだテキスト文書を検出します。

### ヒント

停止ワードのリストを変更する必要がある場合は、インストール時に一度だけ行うようにしてください。

停止ワードのリストは、ユーザーが変更可能なファイル内に言語ごとに保管されます (60ページの『停止ワードと省略語ファイルの変更』を参照してください) が、ファイルはシステム全体で 1 つしかないので、変更は、DB2 テキスト・エクステンダーを最初にセットアップするときに一度だけ行うようにしてください。後でそのファイルを変更しても、既存の索引はその変更を反映しません。

たとえば、「Now」という週刊誌名を含む文書があるものとします。この語を停止ワードから除去すると、これに索引が付けられ、後の検索で検出可能になります。ただし、停止ワードを除去する前に作成されたどの索引にも「now」という語は含まれず、その検索は失敗に終わります。

停止ワードを変更することにし、その変更をすべての索引に反映させたい場合は、全索引を作成し直す必要があります。

索引付けは 2 段階のプロセスで行われます。第 1 ステップでは、索引付けを行うテキスト文書がログ表に記録されます。これは、列中でテキスト文書の挿入、更新、または削除を行うたびに DB2 トリガーによって自動的に実行されます。

第 2 ステップでは、ログ表にリストされたテキスト文書を索引付けします。これは定期的に行うことができます。列中で挿入または変更が行われた文書の用語は、索引に追加されます。列から削除された文書の用語は、索引から除去されます。



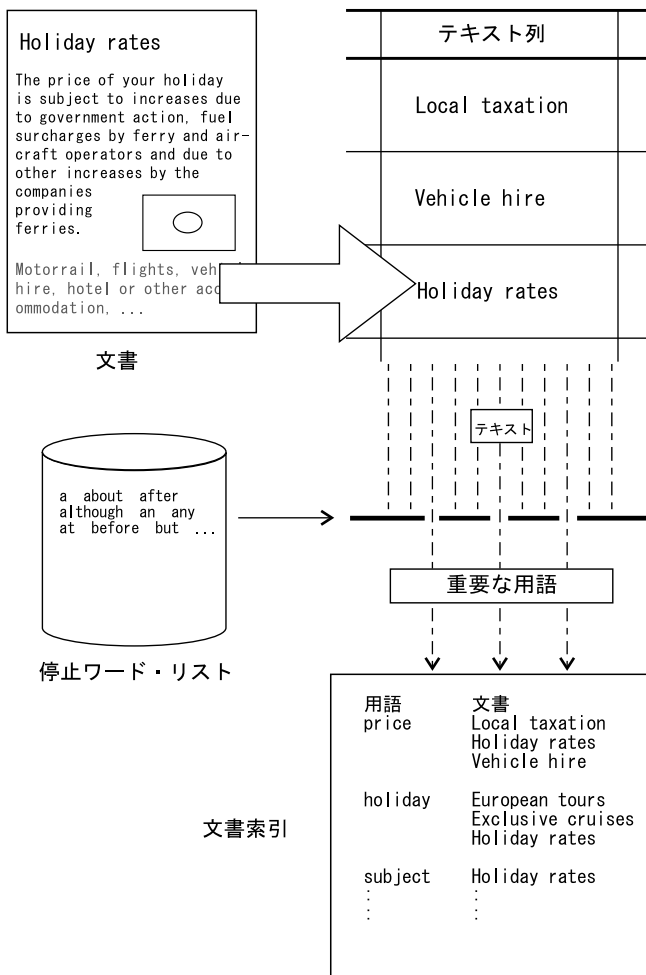


図4. 重要な用語だけの索引付け

## サポートされる文書形式

DB2 テキスト・エクステンダーは、検索する予定のテキスト文書の形式（またはタイプ）を知る必要があります（HTML や ASCII など）。この情報は、テキスト文書の索引付けの際に必要です。

サポートされるテキスト文書形式は以下のとおりです。

<b>HTML</b>	ハイパーテキスト・マークアップ言語
<b>XML</b>	拡張マークアップ言語
<b>ASCII_SECTIONS</b>	セクションを含む構造化 ASCII

## 文書形式

<b>TDS</b>	フラットな ASCII
<b>AMI</b>	AmiPro アーキテクチャー バージョン 4
<b>FFT</b>	IBM 最終形式テキスト: 文書コンテンツ・アーキテクチャー
<b>MSWORD</b>	Microsoft Word、バージョン 5.0 および 5.5
<b>RFT</b>	IBM 変更可能テキスト: 文書コンテンツ・アーキテクチャー
<b>RTF</b>	Microsoft Rich Text Format (RTF)、バージョン 1
<b>WP5</b>	WordPerfect (OS/2 および Windows)、バージョン 5.0、5.1、および 5.2

### HTML 文書に関する特別な考慮事項

HTML 文書のウムラウトや発音区別文字の扱いは、文書のコード・ページに依存しています。

- コード・ページが 37、273、277、278、280、284、297、437、500、819、850、858、860、863、865、871、923、924、および 1252 の場合、以下が適用されます。
  - ウムラウトや特殊文字を表す場合に、実体表記が使用されます。たとえば、`&auml;` が ä を表すために使用されます。
  - コード・ページ 819 (ASCII) または 500 (EBCDIC) でコード・ポイントを持つ文字だけが有効です。
  - 文書のコード・ページが 819 または 500 でない場合、ä のような言語固有の文字が含まれる文書では、索引付けの時に言語認識エラーが起きません。
  - 文書を Ngram 索引に追加する場合、索引は、コード・ページ 819、500、または UTF8 を使用して作成されたものでなければなりません。
- 他のすべての 1 バイト文字セットのコード・ページの場合、以下が適用されます。
  - 実体は解決されません。
  - 特殊文字は言語固有のコード・ポイントで書かなければなりません。

## XML 文書に関する特別な考慮事項

XML 文書には、エンコード・ストリングが含まれていなければなりません。これは構文解析の時に使用されます。以下のエンコードがサポートされています。

```
UTF8, utf8, utf-8, ibm-1208, utf_8,
UTF16 BigEndian, utf-16be, utf16
UTF16 LittleEndian, utf-16le
LATIN_1, latin1, latin-1, ascii, ibm-819, iso-8859-1, Latin-1
ibm-912, iso-8859-2
ibm-913, iso-8859-3
ibm-914, iso-8859-4
ibm-915, iso-8859-5
ibm-1089, iso-8859-6
ibm-813, iso-8859-7
ibm-916, iso-8859-8i
ibm-920, iso-8859-9
ibm-0037, ebcdic-cp-us, ebcdic-cp-ca, ebcdic-cp-nl, ebcdic-cp-dk, ebcdic-cp-no,
    ebcdic-cp-fi, ebcdic-cp-se, ebcdic-cp-it, ebcdic-cp-es, ebcdic-cp-gb
ibm-297, ebcdic-cp-fr, ebcdic-cp-ar1, ebcdic-cp-he, ebcdic-cp-ch, ebcdic-cp-roece,
    ebcdic-cp-yu, ebcdic-cp-is, ebcdic-cp-ar2
ibm-954, euc-jp eucjis
ibm-943, shift_jis, sjis, shiftjis, shift-jis
ibm-950, big-5, big5
ibm-949, iso2022kr, euc-kr
ibm-878, koi8-r
```

XML 文書の場合、省略時のコード・ページは無視されます。

XML 文書を Ngram 索引に追加する場合、索引は、コード・ページ UTF8 を使用して作成されたものでなければなりません。索引にセクションのサポートがある場合、索引付けの際、葉レベルの識別子だけが使用されます。ネストされたセクション内のテキストの解決はサポートされていません。

## サポートされていない文書形式の使用

サポートなしの文書形式には、数値 ID を指定してください。有効な値は、1 ~ 100 です。この値は、ユーザー出口からソース形式に渡され、オリジナル形式から TDS に変換します。

索引付けの実行時に、サポートされていないタイプの文書が見つかると、DB2 テキスト・エクステンダーは出口を提供します。出口では、その文書をディスクに書き込み、ユーザーが提供したプログラムを呼び出して、テキストを抽出し、サポートされるいずれかの書式にします。ユーザー出口は、サーバー構成ファイル (DESSRV.INI) およびクライアント構成ファイル (DESCL.INI) の両方に登録する必要があります。[DOCUMENTFORMAT] セクション内の USEREXIT オプションを、ユーザー出口の名前に更新します。

ユーザー出口を使用可能にするには、以下の ASCII ファイルを編集します。

## 文書形式

Windows NT および Windows 2000 の場合:  
%DMBMPATH%¥instance¥%DB2INSTANCE%¥db2tx¥desc1.ini  
%DMBMPATH%¥instance¥%DB2INSTANCE%¥db2tx¥txinsnnn¥dessrv.ini

UNIX の場合:  
\$DB2TX\_INSTOWNERHOMEDIR/db2tx/desc1.ini  
\$DB2TX\_INSTOWNERHOMEDIR/db2tx/txinsnnn/dessrv.ini

以下のステートメントを追加します。

```
[DOCUMENTFORMAT]  
USEREXIT=name_of_executable
```

ここで、<name\_of\_executable> はユーザー出口の名前です。完全修飾ファイル名を指定するか、または、ユーザー出口が PATH ステートメント内のディレクトリーに保管されている場合には、ファイル名のみ指定することもできます。

ユーザー出口を呼び出すには、以下の構文を使用します。

```
<name_of_user_exit> -sourcefile <sourcefilename>  
                    -targetfile <targetfilename>  
                    -sourceccsid <sourceccsid>  
                    -targetccsid <targetccsid>  
                    -sourceformat <sourceformat>  
                    -targetformat <targetformat>
```

### sourcefilename

ユーザー出口プログラムによって変換されるファイル。ファイル名は完全修飾名であり、クライアント・プロファイルまたはサーバー・インスタンスのいずれかで指定した作業ディレクトリーに置かれます。

### targetfilename

ユーザー出口の出力を収めるファイル。この後、DB2 テキスト・エクステンダーはこのファイルを使って処理を実行します。ファイル名は完全修飾名であり、クライアント・プロファイルまたはサーバー・インスタンスのいずれかで指定した作業ディレクトリーを指します。クライアント・プロファイル内の項目は API 呼び出しの EhwGetMatches で、またサーバー・インスタンス内の項目は API 呼び出しの EhwUpdateIndex で使用されます。

### sourceccsid

ソース・ファイルのコード・ページ。これは省略時のコード・ページです。

### targetccsid

DB2 テキスト・エクステンダーが予期するコード・ページ。コード・ページは 850 です。

**sourceformat**

ソース・ファイルの形式。これは省略時の形式です。

**targetformat**

DB2 テキスト・エクステンダーが予期するファイルの形式。現行では、フラット・ファイル形式 (TDS)、またはセクションが使用可能な索引の場合は ASCIISECTION だけがサポートされています。

ユーザー出口は以下の値を戻すことができなければなりません。

- 0 形式の変換が正常に行われたことを示します。
- >0 形式の変換が正常に行われなかったことを示します。索引付けの時、エラー・メッセージは文書のエラー表に書き出されます。エラー・メッセージを表示するには、 `desmsgix` コマンドを使用します。

**言語**

DB2 テキスト・エクステンダーは、実行する言語処理に正しい辞書が使用されるように、文書がどの言語で書かれているかを知る必要もあります。59ページの表5 は、テキスト列または外部文書を使用可能にするとき、指定できる言語パラメーターのリストを示しています。

**CCSID****ヒント**

テキスト列を使用可能にする際に CCSID を指定する場合は、その前に42ページの『テキストの保管および使用可能化の際に起きるコード・ページに関する問題の回避』をお読みください。

CCSID が以下の表のいずれかである文書は、索引付けすることができます。

**注:** CCSID 861、865、および 4946 は DB2 ではサポートされていません。これらの CCSID の文書を索引付けするには、2 進数データ・タイプ (BLOB または FOR BIT DATA) で文書を列に保管してください。

**EBCDIC**

- 37 米国、カナダ英語
- 273 オーストリア、ドイツ語

## CCSID

277	デンマーク語、ノルウェー語
278	フィンランド語、スウェーデン語
280	イタリア語
284	スペイン語 (ラテン・アメリカ)
285	英国英語
297	フランス語
420	アラビア語
424	ヘブライ語
437	米国英語
500	国際ラテン -1
871	アイスランド語
875	ギリシャ語
1025	ロシア語
1112	ラトビア語
1122	エストニア語
1123	ウクライナ語

## ASCII

813 AIX、HP、SUN	ギリシャ語
819 AIX、HP、SUN	ラテン -1
850 AIX、OS/2	ラテン -1
855 OS/2	ブルガリア語
860 OS/2	ポルトガル語
861 注を参照	アイスランド語
862 OS/2	ヘブライ語
864 OS/2	アラビア語
863 OS/2	カナダ英語
865 注を参照	デンマーク語、ノルウェー語
866 OS/2	ロシア語
869 OS/2	ギリシャ語

915 AIX、OS/2、HP	ロシア語
916 AIX	ヘブライ語
921 AIX、OS/2、WIN	ラトビア語
922 AIX、OS/2、WIN	エストニア語
1064 AIX	アラビア語
1089 AIX、HP	アラビア語
1124 AIX	ウクライナ語
1125 OS/2	ウクライナ語
1129	ベトナム語
1130	ベトナム語
1131	ベトナム語
1250 WIN	クロアチア語、ペロルシア語
1251 WIN	ロシア語
1252 WIN	ラテン -1
1253 WIN	チェコ語
1255 WIN	ヘブライ語
1256 WIN	アラビア語
1257 WIN	ギリシャ語
1258 WIN	ベトナム語

## DBCS

932 AIX、OS/2	日本語、SBCS/DBCS の結合
942 OS/2	日本語、SBCS/DBCS の結合
943 OS/2、WIN	日本語、SBCS/DBCS の結合
5039 HP	日本語、SBCS/DBCS の結合
954 AIX、HP、SUN	日本語
949 OS/2	韓国語
970 AIX、HP、SUN	韓国語
1363 WIN	韓国語
948 OS/2	中国語 (繁体字)、SBCS/DBCS の結合

## CCSID

### 950 AIX、HP、OS/2、SUN、WIN

中国語 (繁体字)、SBCS/DBCS の結合

### 964 AIX、HP、SUN

中国語 (繁体字)、SBCS/DBCS の結合

### 1381 OS/2、WIN

中国語 (簡体字)、SBCS/DBCS の結合

### 1383 AIX、HP、SUN

中国語 (簡体字)、SBCS/DBCS の結合

### 1386 AIX、OS/2、WIN

中国語 (簡体字)、SBCS/DBCS の結合

### 4946 注を参照

ラテン -1 (CP850)

### 5039 HP

日本語

## UNICODE

### 1208

UTF8

### 13488

UCS2

## テキストの保管および使用可能化の際に起きるコード・ページに関する問題の回避

以下の各項目にはコード・ページの設定値があります。

- アクティブ・アプリケーション環境
- 各文書
- 各 DB2 データベース
- 各 DB2 テキスト・エクステンダーの索引

VARCHAR や CLOB などの文字データ・タイプを持つ DB2 データベース列に文書を保管する場合、DB2 は、各文書のコード・ページはアクティブ・アプリケーション環境と同じと考えて、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。データベースのコード・ページは、すでにアクティブ・アプリケーション環境のコード・ページと同じになっているか (この場合、変換は行われません)、データベースの作成時に指定した、アプリケーションとは異なるコード・ページ (この場合、変換が行われます) のいずれかです。

BLOB や FOR BIT DATA などのバイナリー・データ・タイプを持つ DB2 データベースの列にデータを保管する場合、DB2 はデータを変換せず、文書は元の CCSID を保持します。



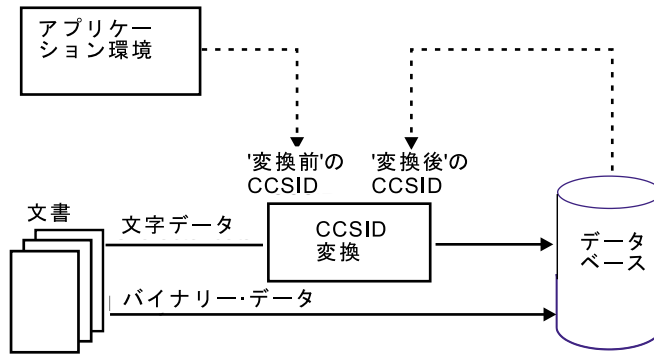


図 5. DB2 がデータベース文書の CCSID を設定する方法

テキスト列を DB2 テキスト・エクステンダーが使用できるようにする、つまり ENABLE TEXT COLUMN コマンドを使用して検索用の索引を作成すると、索引のコード・ページは、データベースのコード・ページ (省略時値) か、CHANGE TEXT CONFIGURATION コマンドを使用して設定できる現行の省略時値か、または ENABLE TEXT COLUMN コマンドで指定したコード・ページのいずれかに設定されます。

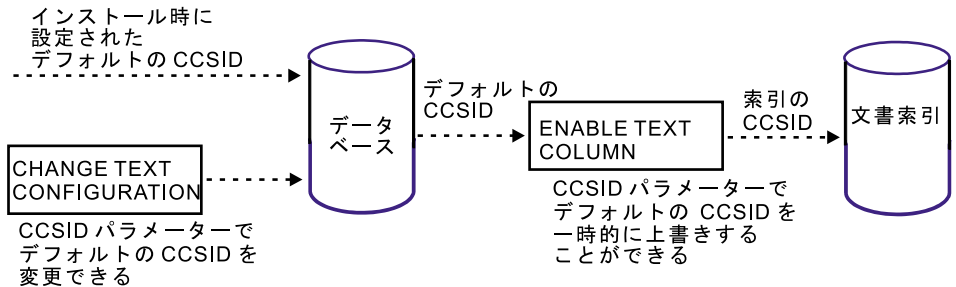


図 6. DB2 テキスト・エクステンダーがテキスト索引の CCSID を設定する方法

検索の際には、データベースの CCSID を使用して検索文字列の CCSID が解釈されます。

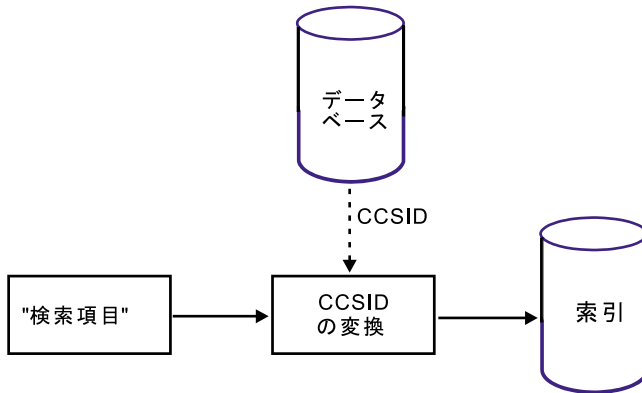


図7. 検索でデータベースの CCSID が使用される

以下はコード・ページの設定を調べる方法です。

- データベースのコード・ページを調べるには、以下の DB2 コマンドを使用します。

```
db2 get db cfg
```

- 省略時の索引コード・ページを調べるには、以下の DB2 テキスト・エクステンダー・コマンドを使用します。

```
db2tx get text cfg
```

コード・ページの問題を回避するには、これらのコード・ページの設定を正確に調整することが重要です。

- **例 1:** 環境 850、文書 850、データベース 850、索引 850

**正常。** DB2 は、文書と環境のコード・ページが同じであると正しく判断します。データベースと環境のコード・ページが同じであるため、DB2 は変換を行わず、コード・ページ 850 で文書をデータベースに保管します。後でテキスト列を使用可能にするとき、省略時のコード・ページの設定 (データベースのコード・ページ 850) を採用すると、文書は正確に、850 のコード・ページの索引に索引付けされます。

- **例 2:** 環境 1252、文書 1252、データベース 850、索引 850

**正常。** DB2 は文書と環境のコード・ページが同じであると正しく判断します。そして、文書をデータベースに保管するときに、1252 から 850 へのコード・ページ変換を行います。後でテキスト列を使用可能にするとき、省略時のコード・ページの設定である 850 (データベースのコード・ページ) を採用すると、文書は正確に、850 のコード・ページの索引に索引付けされます。

- **例 3:** 環境 1252、文書 850、データベース 850、(索引は任意)

エラー。DB2 は、文書 (コード・ページ 850) と環境 (コード・ページ 1252) のコード・ページが同じであると間違っ判断します。そして、文書をデータベースに保管するときに、1252 から 850 への誤ったコード・ページ変換を行います。

- **例 4:** 環境 1252、文書 850、データベース 1252、索引 850

**正常。** DB2 は、文書と環境のコード・ページが同じであると間違っ判断をしますが、環境とデータベースのコード・ページが同じであるため、DB2 は変換を行わず、コード・ページ 850 で文書をデータベースに保管します。ただし、後にテキスト列を使用可能にするとき、文書のコード・ページを 850 に指定して、文書が 850 のコード・ページの索引に正しく索引付けされるようにしなければなりません。

- **例 5:** 環境 1252、文書 1252、データベース 850、索引 1252

**エラーになる可能性あり。** DB2 は、文書と環境のコード・ページが 1252 であると正しく判断します。そして、文書をデータベースに保管するときに、コード・ページを 850 に変換します。コード・ページの索引が 850 であるテキスト列を使用可能にするときに、文書の元のコード・ページである 1252 を指定すると、エラーが起きる可能性があります。正しい処置は、省略時のコード・ページ設定の 850 (データベースのコード・ページ) を採用するというものです。

---

## 検索のタイプ

検索するテキストを含む列に対して、以下のいずれかの検索タイプおよびさまざまなオプションを割り当てることができます。すなわち、基本語形、一致語句、および *Ngram* です。このような列を作成して DB2 テキスト・エクステンダーで使用するには、その前にまずどの索引タイプを作成するかを決めておかなければなりません。各索引タイプが言語処理にどのような影響を与えるかについて詳しくは 263 ページの『第15章 基本語形索引と一致語句索引の言語的な処理』を参照してください。

### 要約

- **基本語形の変化形を検索する場合は、基本語形索引を使用します。**  
語の変化形を、正規化と語幹処理、および辞書の使用に基づいて検索します。ディスク・スペースが一番少なくて済みます。
- **完全一致検索を行う場合は、一致語句索引を使用します。**  
入力した通りの用語を検索します。索引付けと検索がより速く行えます。より多くのディスク・スペースを必要とします。正規化する場合、検索は大文字小文字を区別します。
- **文字の変化形を検索する場合は、Ngram 索引を使用します。**  
つづりの間違って語も検出します。CASE\_ENABLED を指定して大文字小文字の区別をする検索を行う場合、索引はより多くの容量を使用し、検索により時間がかかることがあります。
- **DBCS 文書の検索を行う場合は、Ngram 索引を使用します。**  
DBCS 文書の唯一の選択です。ただし、タイプ TDS の SBCS 文書でも使用することができます。

DB2 テキスト・エクステンダーは、多様な検索オプションを備えていますが、そのすべてをすべての索引タイプで使用できるわけではありません。どの索引タイプを使用するかを決める前に、229ページの表8 および229ページの表9 を参照してください。

## 基本語形検索

基本語形索引の場合、各文書テキストを分析して索引付けを行う際に、言語的な処理が適用されます。つまり、語は基本形に直されてから索引に保管されます。たとえば、「mice」は mouse として索引に保管されます。

基本語形索引に対して照会を行う場合も、これと同じ言語的な処理が検索項目に適用されてからテキスト索引が検索されます。このため「mice」を検索する場合は、基本形の mouse に直されてから検索が開始されます。264ページの表18に、基本語形索引を用いた索引付けの場合に、用語がどのように抽出されるのかを要約しています。

この索引タイプの利点は、検索項目としてどのような変化形を指定しても、索引付けされたテキスト文書のどれかに出てくる他の変化形と一致するということです。mouse という検索項目は、文書用語の

「mouse」、「mice」、「MICE」(大文字)などと一致します。同様に Mice という検索項目も、同じ文書用語と一致します。

この索引タイプは、最もディスク容量が少なくて済みます。ただし、索引付けと検索にかかる時間が一致語句索引の場合より長くなります。

使用できる言語的な処理のタイプは文書の言語によって異なります。以下にタイプをリストします。

- 語および文の分割
- 文の開始処理
- ハイフンの除去
- 用語の標準形への正規化。これには大文字は使われず、「ü」などのアクセント記号がついた文字はアクセント記号のない書式に変更されます。たとえば、ドイツ語の「Tür」(ドア)は、tuer として索引付けされます。
- 用語の基本形への変形。たとえば、「bought」は buy、「mice」は mouse として索引付けされます。

### ヒント

語の断片(ワイルドカード文字でマスクされた語)は、基本形に変形できません。したがって、swu% を検索すると、「swum」という語は索引の中で基本形に変形されるので検出されません。この語を検出するには、swi% により検索しなければなりません。

つづりが正確でない語の変形は、基本形に直すことができません。ドイツ語の röstete を例に考えてみましょう。この語は rösten という動詞の基本形に正しく索引付けされます。検索項目 röstete または rösteten は、基本形の rösten に正しく正規化され、この用語が索引で見つかります。rostete という検索項目では、rösten に正規化することができず、この用語は索引で見つかりません。

- 語の分解。これによりドイツ語の「Wetterbericht」(天気予報)などの複合語は、wetterbericht だけでなく wetter および bericht としても索引付けされます。
- 停止ワードのフィルター処理。これは、関係のない用語を索引付けしません。「A report about all animals」は、report および animal として索引付けされます。
- 品詞のフィルター処理。これは停止ワードのフィルター処理に似ており、名詞、動詞、形容詞だけが索引付けされます。「I drive my car quickly」は、

## 検索のタイプ

drive および car として索引付けされます。停止ワードとして「I」と「my」の語が除去され、さらに副詞「quickly」が品詞のフィルター処理によって除去されます。

### 一致語句検索

一致語句索引では、テキスト文書内の用語は文書に出てくるとおりに索引付けされます。たとえば、mouse という検索項目で「mouse」を検出できますが、「mice」や「Mouse」は検出できません。一致語句索引での検索では大文字と小文字が区別されます。

照会でも同じ処理が照会用語に適用され、照会用語は索引中の検出される用語と比較されます。すなわち、検出される用語は検索用語と全く同じになります。マスキング文字を使用すれば検索範囲を広げることができます。たとえば、experiment% という検索項目で「experimental」や「experimented」などを検索できます。

264ページの表19 に、一致語句索引を使用する索引付けの場合に、どのように用語が文書テキストから抽出されるか、いくつか例を挙げてあります。

この索引タイプの利点は、検索が正確であることと、索引付けと検索の速度が速いことです。あらゆる用語の変化形ごと、つまりつづりが違ふとそのたびに索引付けされるので、基本語形索引の場合よりも多くのディスク・スペースが必要です。

一致語句索引の場合に索引のテキスト文書に使用される言語的な処理は以下のとおりです。

語および文の分割

停止ワードのフィルター処理

### ファジー検索または DBCS 文書の検索

Ngram 索引は、文字セットを解析することによって、テキストを分析します。この分析は辞書に基づいて行われるものではありません。

テキストに DBCS 文字が含まれている場合は、Ngram 索引を使用する必要があります。これ以外の索引タイプは、DBCS 文字をサポートしません。

この索引タイプは、「ファジー」検索をサポートします。すなわち、指定した検索項目に類似した文字ストリングを検出することができます。たとえば、Extender を検索すると、つづりを誤った語 Extendrns も検出されます。また、要求する類似性のレベルを指定することもできます。

**注:** ファジー検索を使用する場合でも、先頭の 3 文字は一致していなければなりません。

Ngram 索引で大文字小文字の区別をする検索を行うためには、その照会で PRECISE FORM OF キーワードを指定するだけでは十分ではありません。これは、Ngram 索引では通常、索引付き文字が大文字か小文字かを区別しないためです。しかし、Ngram 索引の作成時に、CASE\_ENABLED オプションを指定することによって、その索引で大文字小文字が区別されるようにできます。さらに、照会時に、PRECISE FORM OF キーワードを指定します。

CASE\_ENABLED オプションを使用すると、索引用に必要なスペースが増え、検索により時間がかかることがあります。

Ngram 索引でサポートされる CCSID のリストについては、39ページの『CCSID』を参照してください。Ngram 索引は、ネイティブの CCSID のリストをサポートします。他のすべての CCSID の場合、データはこの CCSID から UTF8 にマップされます。

Ngram 索引タイプは DBCS 文書の索引付けのために使用するよう考えられていますが、SBCS 文書に対して使用することもできます。ただし、サポートする文書は TDS 文書だけです。

また、すべての構文検索オプションがサポートされるわけではないことにも注意してください。221ページの『第12章 検索引き数の構文』に記載されている規則と制限の要約を参照してください。

## 索引タイプの変更

使用中の索引タイプが適切でない判断したならば、まずテキスト列またはテキスト表を使用不可にして索引を削除し、その後テキスト列またはテキスト表を再び使用可能にして索引を再作成します。

---

## 表に対する 1 つまたは複数のテキスト索引の作成

63ページの『第5章 テキストを検索可能にする』では、テキストを検索するための表の作成方法について説明します。しかしその前に、表中のすべての索引付きテキスト列に対して共通なテキスト索引を 1 つ作成するか、または索引付きテキスト列ごとに 1 つずつ、つまり複数のテキスト索引を作成するのかを決めておかなければなりません。テキスト列ごとに別個の索引を持つ表は、複数索引表と呼ばれます。

## 表に対する 1 つまたは複数のテキスト索引の作成

### ヒント

外部ファイルを索引付けする予定である場合 (79ページの『外部テキスト・ファイルの使用可能化』を参照)、関連する表は複数索引表でなければなりません。

複数索引を使用すると以下の利点があります。

- テキスト列ごとに異なる索引タイプを作成できる  
テキスト列の索引を更新する周期や索引の保管先のディレクトリーなどの、テキスト列に関連した特性に柔軟性が増します。これらの特性については 148ページの『ENABLE TEXT COLUMN』を参照してください。
- 列の索引付けを別々に行える  
索引付けの活動に多くの時間と資源を費やす場合があります。複数索引表を作成すると、列の索引付けを様々な時点で行うので、この活動にある程度余計に時間がかかることがあります。

複数索引表を作成して柔軟性を高める必要がない場合は、共通索引を使用して DB2 テキスト・エクステンダーでの保守を簡単にすることができます。テキスト表を使用可能にするときに設定する索引パラメーターが、そのテキスト列すべてに対する省略時値として使用されます。また、列を使用不可にする必要がある場合も、テキスト表を使用不可にするためのコマンドを 1 つ使用するだけで済みます。

---

## 索引サイズの計算

索引に必要なディスク・スペースは、索引付けされるデータのサイズおよびタイプと索引タイプによって決まります。ワード・プロセッサによって作成されたテキスト文書は、多くの内容に制御文字を含んでいるため、スペースは少なく済みます。目安としては、基本語形索引の場合、索引付けされる文書のサイズの 0.7 倍のディスク・スペースを予約し、さらにこの 2 倍のスペースを、索引を再編成するための一時スペースとして予約します。Ngram 索引の場合は、ディスク・スペースの約 2 倍が必要になります。

大量のスペースを要する索引がいくつかあり、索引の更新または検索中にそれらの索引に同時にアクセスする場合には特に、それらを別個のディスク装置に格納する必要があります。



## 索引の更新

データベースにテキスト文書を追加するか、データベース内の既存の文書を変更する場合は、その文書を索引付けして、索引の内容とデータベースの内容を同期化する必要があります。テキスト文書をデータベースから削除する場合は、その用語を索引から削除する必要があります。

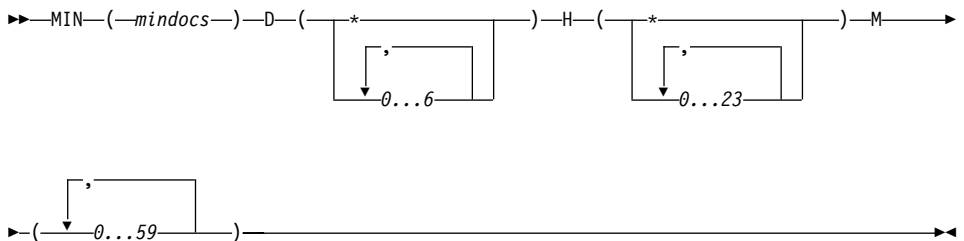
文書が新規のものか、変更を加えられているか、削除されたかに関する情報は、トリガーによって、自動的にログ表に保管されます。ログ表にリストされた文書は、次に索引が更新されたときに索引付けされます。

UPDATE INDEX コマンドを使用すると、要求した時すぐに索引を更新できます。しかし一般には、環境変数 DB2TXUPDATEFREQ に指定した間隔で索引を自動的に更新するのが普通です。この環境変数には省略時値の設定が指定されています。省略時値の設定は、ENABLE TEXT COLUMN または ENABLE TEXT TABLE コマンドを使用して索引を作成するとオーバーライドすることができます。既存の索引に関する更新頻度は、CHANGE INDEX SETTINGS コマンドを使用して変更することができます。

索引更新の頻度は、更新を行う時点と、待ち行列に入れなければならないテキスト文書の最小数とに基づいて指定します。指定された日時にログ表内に十分な数の文書がない場合、索引は更新されません。

テキスト文書の索引付けは時間がかかり資源も浪費するタスクですので、周期的な索引付けを慎重に計画してください。索引付けにかかる時間は、前回の索引更新以後追加または変更されたテキスト文書の数、文書のサイズ、およびプロセッサの処理能力など多くの要因により異なります。

## 構文

**MIN mindocs**

索引を更新する前に待ち行列になければならないテキスト文書の最小数。

**D** 索引を更新する曜日。

## 索引の更新

- \* 毎日
- 0** 日曜日
- 1** 月曜日
- 2** 火曜日
- 3** 水曜日
- 4** 木曜日
- 5** 金曜日
- 6** 土曜日
- H** 指定された日の、索引を更新する時
  - \* 一時間ごと
  - 0...23** 時の指定範囲
- M** 指定された時の、索引を更新する分
  - 0...59** 分の指定範囲

例: `min(100) d(1,2,3,4,5) h(12,15) m(0)`

この例では、月曜日から金曜日の 12:00 または 15:00 に最低 100 個のテキスト文書が待ち行列にあれば、索引は更新されます。

以下のように、複数の頻度の指定を結合することができます。

例: `min(1) d(*) h(22) m(0) ; min(100) d(1,2,3,4,5) h(12,15) m(0)`

この場合、前の例と同様に索引の更新は月曜日から金曜日の 12:00 と 15:00 にスケジュールされていますが、さらに、ログ表にテキスト文書が 1 つしかない場合でも、毎日 22:00 に索引は更新されます。

---

## マルチノード環境での索引の保守

使用可能にしているテキスト列がマルチノード・ノードグループの一部である表に属している場合には、指定する索引ディレクトリーは、すべての物理ノードで使用できなければなりません。テキスト構成で指定される省略時ディレクトリーを使用する場合には、ノードグループのすべてのノードでパスを使用できることを確認してください。これが適当でない場合には、それぞれのノードについて特定のパスを `ENABLE TEXT COLUMN` コマンドに指定することができます。

DB2 テキスト・エクステンダーに使用可能にされた表を含むノードグループのノード構成を変更するには、表を索引付けし直さなければなりません。ノード構成が変更されたかどうかは、コマンド TXNCHECK を使用してチェックできません。

#### ヒント

DB2 UDB エンタープライズ拡張エディションを使用している場合で、複数のノードに表区画がある場合には、ノード間のシステム時刻が異なっている場合があります (たとえば、ノードの 1 つはサマー・タイムであるなど)。問題を解決するために、ノード間のシステム時刻設定の相違が、DB2 の MAX\_TIME\_DIFF 設定の値の範囲内であることを確認してください。Windows NT および Windows 2000 では、NET TIME コマンドを使用してシステム時刻設定を同期化してください。

## 構造化文書 (セクション・サポート) での作業

セクション・サポートにより、構造化された文書の中の特定のセクション (表題、著者、説明など) に索引を付けて、検索することができます。使用できる文書形式は XML または HTML 形式、または HTML に似たタグを使用したフラット・ファイル文書です。マークアップのタグとそれに対応するセクション名は、文書モデル に定義します。文書モデルには、文書内のどのセクションに索引を付けて検索できるようにするかを定義します。セクション名は、そのセクションに対する照会で使用される記述的な名前です。

セクション・サポートを使用可能にするには、文書を含むテキスト列を使用可能にするときに INDEXPROPERTY SECTIONS\_ENABLED を指定する必要があります。

文書モデル・ファイル には、サーバー・インスタンスの定義済み文書モデルがすべてリストされています。サーバー・インスタンスを作成すると、サンプル・モデル・ファイル DESMODEL.INI がサーバー・インスタンス・サブディレクトリーに自動的に作成されます。ファイルのコード・ページは ASCII です。

文書モデルの情報は、索引ディレクトリーにコピーされます。索引を作成した後にはサーバー・インスタンスの文書モデルを変更しても、作成した索引のセクション・サポートには影響ありません。

## 構造化文書 (セクション・サポート)

セクション・サポート付きの索引を検索する (たとえば、著者のセクションから McDaniel を検索する) 方法は以下のとおりです。セクション (この場合は Author) の前には必ずモデル名が付きます。

```
db2 "select count (*)
     from db2tx.htmltable
     where db2tx.contains(myhandle,'MODEL myhtmlmodel SECTION (author) "Schmidt") = 1
```

### 属性セクション

セクションにはさまざまなタイプがあります。平文のセクションにタイプはありません。宣言されたタイプのセクションは、属性セクションと呼ばれます。サポートされる属性タイプは以下のとおりです。

- DATE
- TIME
- FLOAT
- INTEGER

属性セクションを含む文書を検索するときには、値の範囲を使用することができます。セクションは、適切なセクション・タイプでなければなりません。以下の例の場合、照会がサポートされるためには、セクション ABC のタイプが TIME でなければなりません。

```
retrieve all documents containing values between
20:00:00 and 22:00:00 in the section ABC
```

1 つのセクションに複数の属性値を含めることができます。ブランクまたは改行で値を区切ってください。属性セクションをネストすることはできません。

### サポートされる日付形式

文書モデル内のサポートされるセクションのリストに日付 (作成日付や出版日付など) を含めることができます。サポートされる日付形式は以下のとおりです。

表 3. サポートされる日付形式

形式名	形式	例
ISO	YYYY-MM-DD	1986-12-25 100-12-25 10123-2-17
	DD-MM-YYYY	01-10-1999 5-8-1990

表3. サポートされる日付形式 (続き)

形式名	形式	例
IBM USA 標準規格 (USA)	MM/DD/YYYY	08/18/1999 5/13/1977
	YYYY/MM/DD	1999/08/18
IBM 欧州標準規格 (EUR)	DD.MM.YYYY	20.10.1999 1.5.1980

特定の日付に出版された文書をセクション・サポート付きの索引を使って探す場合、検索は次のようになります。

```
dessrch -s SERVER -x TESTIX -section sample/PublishingDate -term 1999-09-09
```

### サポートされる時刻形式

サポートされる時刻形式は以下のとおりです。

表4. サポートされる時刻形式

形式名	形式	例
ISO	hh:mm:ss	19:00:23
	hh:mm	19:00
	hh.mm.ss	19.00.23
	hh.mm	19.00
	hh	19
	hhmm	1900
	hhmmss	190013

時間の先頭のゼロは省略できます。以下の範囲外の値は無効です。

時間	分	秒
0-23	0-59	0-59

### サポートされる浮動形式

サポートされる浮動形式は以下のとおりです。

- 1.175494351e-38F から 3.402823466e+38F までの範囲の連続した数字
- e または E を使用した科学的記数法

浮動の値を区切るには、ブランクまたは改行を使用します。

## 構造化文書 (セクション・サポート)

以下は、浮動値の正しい形式、および誤った形式の例です。

正しい形式	誤った形式
1000	1 000
1	1,000
1.0e3	e3
1.0E3	
1E3	
.1E4	
0.1E4	

### サポートされる整数形式

-2147483648 から 2147483647 までの範囲の整数がサポートされます。整数を区切るには、ブランクを使用します。つまり、INTEGER タイプのセクションで、ストリング「42 234」は、「42234」ではなく、「42」と「234」になります。

以下は、整数値の正しい形式、および誤った形式の例です。

正しい形式	誤った形式
1000	1.000 1,000 1 000
1000000	100.000 100,00 100 000

#### 注

言語固有の区切り記号および通貨形式はサポートされていません。

## フラット・ファイルおよび HTML 文書

フラット・ファイルの場合、セクションは、<title> や <subject> などの HTML に似たタグを使用してマークアップされます。以下は、マークアップされたセクションを含む文書の例です。

```
<title> IBM Dictionary of Computing
<author> McDaniel, George
<subject> Computers, Reference, ....
```

以下は、フラット・ファイルまたは HTML 文書用の文書モデル・ファイルの例です。モデル名およびセクション名に含めることができるのは、A-Z、a-z、および 0-9 だけです。モデル名には常に大文字小文字の区別があります。セクション名は、大文字小文字を区別するようにすることもしないようにすることもできます。モデルの作成時に設定を指定します。

```
;文書モデルのリスト
;モデルは常に 'modelname' とモデルの名前で始まる
[MODELS]
modelname=sample
modelname=sample2
modelname=sample3
```

```
; 'sample' 文書モデル定義
; 左辺 - セクション名識別子
; 右辺 - セクション名タグ
[sample]
Title = title
Author = author
Subject = subject
Abstract = abstract
Content = content
PublishingDate=date
```

```
[sample2]
Title = title
Author = author
Subject = subject
[sample3]
Title = title
Author = author
Abstract = abstract
Docnum = docnum
```

文書中に現われるマークアップされたセクションが、文書モデルで定義されていない場合、そのセクションの内容は、定義されている直前のセクションに含められて、索引付けと検索がなされます。たとえば、ある文書に以下のマークアップされたセクションが含まれているとします。

```
<title> IBM Dictionary of Computing
<subject> Computers, Reference, ....
<author> McDaniel, George
<abstract> Contains up-to-the-minute coverage of information processing systems,
communication products and facilities, personal computers, and office systems, as
well as the full range of IBM hardware and software products.
```

また、文書モデル book が以下のように定義されているとします。

```
[MODELS]
modelname=book
[book]
Title = title
Author = author
Abstract = abstract
```

## 構造化文書 (セクション・サポート)

book 文書モデルには、<subject> セクションが含まれていません。この文書が索引付けされる時、 subject セクションの内容は title セクションの内容と一緒に索引付けされることになります。 subject セクションの内容が、title セクション内の検索でも使用可能になります。

索引の作成時にモデルのリストを指定する場合、省略時のモデルになるのはリストの先頭のモデルです。 desmodix コマンドを使用して省略時のモデルを変更することができます。

## XML 文書

セクション処理の可能な索引を使用する場合、 XML 文書は正しく構造化されていなければならず、ルート・エレメントが含まれていなければなりません。ルート・エレメントの名前は、定義済みのモデル名のいずれかと同じでなければなりません。大文字小文字も一致していなければなりません。文書モデル・ファイル内のモデルの記述は、文書の DTD (文書タイプ定義) ファイルで定義された文書モデルのサブセットになる必要があります。

モデルの記述はルート・エレメントから始める必要があります。セクションとして使用したい XML エレメントごとに、その完全な階層をモデルの記述に含める必要があります。セクションのタイプが日付の場合、このセクションは文書モデル・ツリーの中で、葉となる必要があります。属性セクションのネストはサポートされていません。

以下は、XML 文書のモデルの説明の例です。

```
; 文書モデルのリスト
[MODELS]
modelname = LETTER
; XML モデル定義のサンプル
; 左辺 = パス全体をエンコードしたセクション識別子。
; 右辺 = セクション名のタグ。ツリーをたどって指定の
;       ノードに至るまでの、パスの各エレメントに
;       タグを指定する。タグの区切り文字は /。
[LETTER]
LETTER = LETTER
LETTER/date = LETTER/DATE
LETTER/address = LETTER/ADDRESS
LETTER/address/City = LETTER/ADDRESS/CITY
LETTER/Content = LETTER/CONTENT
LETTER/Content/Greetings = LETTER/CONTENT/GREETINGS
```

以下は、XML 文書の例です。モデルで定義されないセクションがどのように索引付けされるかも示されています。

```
<?xml version="1.0"?>
<!DOCTYPE LETTER SYSTEM "letter.dtd">
```



```

<LETTER>
  <HEADER>このタグは定義の中で省略されています。
              このテキストは LETTER というセクションに追加されます。
  </HEADER>
  <DATE>
      01.01.2000  03.02.2000
  </DATE>
  <ADDRESS>
      テキストは LETTER/address というセクションに追加されます。
      <CITY>
          テキストは LETTER/address/City というセクションに追加されます。
      </CITY>
  </ADDRESS>
  <CONTENT>
      テキストは LETTER/Content というセクションに追加されます。

      <NOSECTION>テキストは LETTER/Content というセクションに追加されます。
          これは、NOSECTION が定義されていないためです。
      </NOSECTION>
      <GREETINGS>
          テキストは LETTER/Content/Greetings というセクションに追加されます。
      </GREETINGS>
  </CONTENT>
</LETTER>

```

## 辞書、停止ワード・リスト、省略後リスト、および言語パラメーター

表5 は、サポートされる言語と、辞書、停止ワード・リスト、および省略語リストとして提供されるファイルの名前を示しています。辞書ファイルは 2 進数形式で、変更できません。停止ワード・ファイルおよび省略語ファイル (もしあれば) は、フラット・ファイル形式で、変更できます。これらのファイルのいずれかを変更する場合には、必ずその言語のコード・ページを使用してください。

この表は、また、テキスト列または外部文書を使用可能にするときに指定する必要がある言語パラメーターも示しています。言語パラメーターは、文書がどの言語で書かれているかを DB2 テキスト・エクステンダーに知らせ、こうして言語処理が必要になったときに正しい辞書が使用されるようになります。

表5. さまざまな言語で使用される言語機能

言語	ファイル名	言語パラメーター	コード・ページ
アラビア語	arabic	ARABIC	864
ブラジル・ポルトガル語	brazil	BRAZILIAN	850
カナダ・フランス語	canadien	CAN_FRENCH	850
カタロニア語	catala	CATALAN	850
デンマーク語	dansk	DANISH	850

## 辞書、停止ワード・リスト、および省略語リスト

表 5. さまざまな言語で使用される言語機能 (続き)

言語	ファイル名	言語パラメーター	コード・ページ
オランダ語	nederlnd	DUTCH	850
フィンランド語	suomi	FINNISH	850
フランス語	francais	FRENCH	850
ドイツ語	deutsch	GERMAN	850
ヘブライ語	hebrew	HEBREW	862
アイスランド語	islensk	ICELANDIC	850
イタリア語	italiano	ITALIAN	850
ノルウェー語 (ボクモール)	norbook	BM_NORWEGIAN	850
ノルウェー語 (ニューノルスク)	norntn	NN_NORWEGIAN	850
ポルトガル語	portugal	PORTUGUESE	850
ロシア語	russian	RUSSIAN	866
スペイン語	espana	SPANISH	850
スウェーデン語	svensk	SWEDISH	850
スイス・ドイツ語	dschweiz	SWISS_GERMAN	850
タイ語	thai	THAI	850
英国英語	uk	UK_ENGLISH	850
米国英語	us	US_ENGLISH	850

ファイルは、拡張子によって識別されます。

内容	拡張子
辞書	DIC
停止ワード・リスト	STW
省略語リスト	ABR

### 停止ワードと省略語ファイルの変更

言語ごとに、1 つの停止ワード・ファイルと省略語ファイルが存在します。これらのファイルを編集する意味を知るためには 33ページの『テキスト文書を索引付けする理由』を参照してください。

### ヒント

これらのファイルのいずれかの編集を始める前に、バックアップ・コピーをとってください。

停止ワード・ファイルおよび省略語ファイルは、OS/2 と Windows システムでは以下に格納されています。

```
drive:%dmb%db2tx%dict
```

AIX、HP-UX および SUN-Solaris システムでは、以下に格納されています。

```
DB2TX_INSTOWNER /db2tx/dicts
```

使用しているエディターで、これらのファイルを編集します。これらは CCSID 850 を使用しているため、編集を始める前に、ユーザーのアプリケーションの CCSID も必ず 850 に設定します。

索引付けしたい語と省略語を除去します。索引付けしたくない語を加えます。

## 変更の要約

---

## 第5章 テキストを検索可能にする

29ページの『第3章 はじめに』を読むと、DB2 テキスト・エクステンダーによってテキストの検索を可能にすることに精通できるようになります。その章では簡単な例を使って説明しました。この章では、テキストを検索可能にすることについてより詳細に説明されています。また、開始する前に考慮すべき局面すべてが説明されています。

テキストを検索可能にするためのステップは以下のとおりです。

1. 綿密に準備する
2. DB2 テキスト・エクステンダー・コマンド行プロセッサを開始する
3. データベースに接続する
4. データベースをテキストの検索で使用できるようにする
5. テキスト表をテキストの検索で使用できるようにする (テキスト列につき 1 つの索引を作成している場合は必要ありません)
6. テキスト列をテキストの検索で使用できるようにする

---

### テキストを検索可能にする前の準備

#### ヒント

このセクションを注意深くお読みください。テキストを検索可能にする前に知っておく必要のある選択事項がリストされています。

- **UNIX ユーザーの場合、プロファイルを設定アップする**

自分が DB2 テキスト・エクステンダーのインスタンス所有者でない場合には、19ページの『構成』を参照して、プロファイルの設定アップ方法を調べてください。

- **表全体に 1 つの索引を作成するか？**

テキスト表全体について 1 つの索引を作成するか、それともテキスト列ごとに別々の索引を作成するのかを決める必要があります。49ページの『表に対する 1 つまたは複数のテキスト索引の作成』は決定に役立ちます。

- **文書について理解する**

## テキストを検索可能にする前の準備

文書を検索可能にするときに、その CCSID、言語、およびテキストの形式を指定する必要があります。詳細については、33ページの『第4章 必要に応じた検索の計画』を参照してください。

- **必要なテキスト索引のタイプを決める**

必要とする索引のタイプは、行いたい検索の種類（一致語句、ファジーなど）、および文書が SBCS か DBCS かによって決まります。33ページの『第4章 必要に応じた検索の計画』により多くの情報が記されています。

- **索引を保管する場所を決める**

文書を検索可能にすると、DB2 テキスト・エクステンダーはテキスト索引を作成します。索引を保管するディレクトリーを指定する必要があります。十分なディスク・スペースがあることを確認してください（50ページの『索引サイズの計算』を参照してください）。

- **省略時のデータベース名をチェックする**

DB2 環境変数 DB2DBDFT 内の省略時のデータベース名（19ページの『環境変数』を参照）は、テキストを検索可能にするときにデータベース名を指定しない場合に、DB2 テキスト・エクステンダーが使用する名前になります。

- **テキスト構成をセットアップする**

テキスト構成は、索引 CCSID の省略時値の設定、文書の言語、文書の形式、索引タイプ、索引の更新の頻度、表スペース名、および索引ディレクトリーを決定します。

テキストを使用可能にするときにこれらの設定を上書きすることができますが、前もって省略時に正しく設定しておくことより便利です。DB2 テキスト・エクステンダーがインストールされたときのこの表の初期テキスト構成設定値については、21ページの『テキスト構成設定値』で説明しています。インストールの設定を変更して、独自の省略時値をセットアップするには、137ページの『CHANGE TEXT CONFIGURATION』を利用してください。

- **セクション・サポートをセットアップする**

検索をドキュメントの特定のセクション内でだけ行う必要がある場合、53ページの『構造化文書（セクション・サポート）での作業』を読み、文書モデル・ファイル内のモデルを指定する方法を学習してください。

- **停止ワードおよび省略語リストを変更する**

33ページの『テキスト文書を索引付けする理由』および 60ページの『停止ワードと省略語ファイルの変更』を読み、停止ワード・リストと省略語リストの概念を理解して、索引付けをする前に、それらを変更するかどうかを決めます。

63ページの『テキストを検索可能にする前の準備』の説明にしたがって情報を収集し、決定したら、テキストを検索可能にする準備が整ったことになりません。

## DB2 テキスト・エクステンダー・コマンド行プロセッサの開始

### 要約

実行時	任意選択。個々のセッションの開始時
コマンド	db2tx
権限	すべて

オペレーティング・システムのプロンプトで DB2 テキスト・エクステンダー・クライアント・コマンドを入力することができます。これらのコマンドは DB2 コマンドに似ていますが、db2 で始まるのではなく、db2tx で始まります。

すべての DB2 テキスト・エクステンダー・コマンドの接頭部に db2tx をおく代わりに、DB2 テキスト・エクステンダー・コマンド行プロセッサを開始します。これには、各コマンドの後にデータベースの接続が切断されず、接続を保つことができるという利点があります。(コマンド行プロセッサがない場合、db2tx コマンドを実行するたびに自動的に再接続されますが、接続したいデータベースの名前が DB2DBDFT 環境変数に設定されているか確認する必要があります。)

### 1. (Windows のみ): DB2 コマンド行プロセッサを開始する

オペレーティング・システムのプロンプトで以下を入力します。

```
=>db2 cmd
```

### 2. DB2 テキスト・エクステンダー・コマンド行プロセッサを開始する

UNIX システムの場合、オペレーティング・システムのプロンプトで以下を入力します。

```
=>db2tx
```

Windows システムの場合、DB2 プロンプトで以下を入力します。

```
db2=>db2tx
```

db2tx プロンプトが表示されます。

```
db2tx=>
```

## DB2 テキスト・エクステンダー・コマンド行プロセッサの開始

そして、後続のコマンドはすべて DB2 テキスト・エクステンダー・コマンドとして解釈されます。

このモードを終了するには、以下を入力します。

```
db2tx=>quit
```

Windows 95 および Windows 98 では、上記の db2tx コマンドを使用するか、または「スタート」ボタンをクリックし、「プログラム」、「DB2 Client for Windows 95/98 (DB2 クライアント (Windows 95/98 版))」を順に選び、最後に「DB2 Command Window (DB2 コマンド・ウィンドウ)」をクリックします。

このステップを終えたら、db2tx を DB2 テキスト・エクステンダー・コマンドの前に付けて、このコマンドをオペレーティング・システム・プロンプトから直接出すことができます。オペレーティング・システム・プロンプトから出すコマンドの例を以下に示します。

```
=>db2tx enable database
```

### ヒント

クライアントから作業する場合は、DB2 テキスト・エクステンダー・コマンド環境である必要があり、ユーザー ID とパスワードを指定した完全書式の接続ステートメントを使用する必要があります。

```
db2tx => connect to database-name user user-id using password
```

## コマンド行プロセッサのヘルプ

コマンドのリストを表示するには次のように入力します。

```
db2tx ?
```

個々のコマンドの構文を表示するには次のように入力します。

```
db2tx ? command
```

たとえば次のようにします。

```
db2tx ? CHANGE TEXT CONFIGURATION
```



## データベースへの接続

要約	
実行時	任意選択。 DB2DBDFT 環境変数で指定された省略時以外のデータベースを使用してセッションを開始する場合。
コマンド	CONNECT
権限	データベースに関する CONNECT

DB2 テキスト・エクステンダー・セッションでさらにコマンドを発行するには、データベースに接続していなければなりません。データベースに接続しないでコマンドを発行すると、DB2 テキスト・エクステンダーは自動的に DB2DBDFT 環境変数で指定された省略時のデータベースへの接続を実行します。DB2 テキスト・エクステンダーの CONNECT TO コマンドを使用すれば、データベースに明示的に接続することができます。

```
db2tx CONNECT TO sample-database
```

次のように表示されます (AIX 上の場合)。

```
>-----Database Connection Information -----<
Database Product      = DB2/6000 05.02.0000
User                  = user-name
Database alias        = sample-database
```

**注:** コマンド・プロセッサの外で、db2tx を前に付けた DB2 テキスト・エクステンダー・コマンドを続けて出すと、現行のデータベース接続は失われ、省略時のデータベースに新しく接続されます。

OS/2 クライアントを使用していて、システムの起動後初めてデータベースに接続する場合には、ユーザー ID とパスワードを入力するようにユーザー・プロファイル管理 (UPM) からプロンプト指示されます。すでにログオンしている場合は、UPM は、このセッションからユーザー ID とパスワードを入力して、サーバー上のリモート・データベースに接続を試みます。

データベースの使用可能化

要約	
実行時	検索するテキスト列が含まれたデータベースごとに 1 度ずつ
コマンド	db2tx=>ENABLE DATABASE
権限	SYSADM または DBADM

このコマンドにはパラメーターはありません。このコマンドは、接続しているデータベースを DB2 テキスト・エクステンダーで使用できるよう準備します。

また、このコマンドは DB2 に DB2 テキスト・エクステンダーの関数および DB2 テキスト・エクステンダーの特殊タイプを宣言します。これらは SQL 関数であり、後でテキスト検索を行う際に使用します。これらについての説明は 203 ページの『第11章 検索関数』に記載しています。これらの宣言はすべての将来セッションに適用されます。

使用可能なテキスト列を追跡する、TEXTINDEXES というカタログ表示が作成されます。118 ページの『DB2 テキスト・エクステンダー・カタログ表示の処理』を参照してください。

このコマンドは、データベースのテキスト構成情報を作成します。これには、索引、テキスト、および処理特性の省略時値が含まれます。これらについては、21 ページの『テキスト構成設定値』で説明しています。

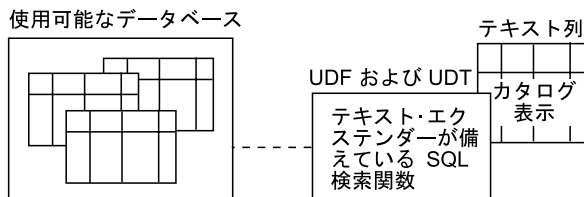


図8. データベースの使用可能化

データベースは、いったん使用可能になると使用不能にするまでは使用可能のままです。ENABLE DATABASE による変更を逆転させる場合は、122 ページの『データベースを使用不可にする』を参照してください。

**ヒント**

環境変数 `DB2TX_INSTOWNER` が使用される場合には、これはデータベースが使用可能になる前のインスタンス所有者の名前に設定されなければなりません。これは UNIX ユーザーにとっては特に重要です。UNIX では、この変数が省略時値に設定されているからです。

後で使用可能なデータベースを除去することにした場合には、最初にこれを使用不可にして、宣言された DB2 テキスト・エクステンダー関数、カタログ表示などを除去するようしなければなりません。

**テキスト表の使用可能化 (任意選択)****要約**

<b>実行時</b>	任意選択。表中のすべてのテキスト列に関する共通索引を作成する場合のみこのコマンドを使用します。49ページの『表に対する 1 つまたは複数のテキスト索引の作成』を参照してください。
<b>コマンド</b>	<code>db2tx=&gt;ENABLE TEXT TABLE ...</code> (例を参照)
<b>権限</b>	表に関する ALTER、SELECT、UPDATE

このステップでは、表中のすべてのテキスト列について共通索引を 1 つ作成するか、またはテキスト列ごとに別々の索引を付けて複数の索引を作成するのかを決めます。

共通索引を作成する場合は、まず `ENABLE TEXT TABLE` を実行し、次にテキスト列ごとに `ENABLE TEXT COLUMN` を実行します。別々の索引を作成する場合は、`ENABLE TEXT TABLE` はスキップし、テキスト列ごとに `ENABLE TEXT COLUMN` だけを実行します。これは図9 と 72ページの図10に記載してあります。

このステップの間に DB2 テキスト・エクステンダーにより空のテキスト索引が作成されますが、これは後で使用可能にするすべてのテキスト列に共通なも

## テキスト表の使用可能化 (任意選択)

のです。索引のタイプ、索引を更新する周期、および索引を保管するディレクトリーを指定します。指定しないパラメーターの省略時値は、テキスト構成設定値から得られます。

### ヒント

ほとんどのテキスト列について、索引タイプなどの設定を同じにする場合は、テキスト構成情報を使用して省略時設定を指定してください。21ページの『テキスト構成設定値』を参照してください。

またこのステップでは、表中の追加、変更、または削除が行われた文書を記録するための空のログ表も作成されます。更新されたログ表を保持するためにトリガーが作成されます。

ENABLE TEXT TABLE は、DB2 テキスト・エクステンダー用に使用可能にしたテキスト列がすでに含まれている表には実行できません。

ENABLE TEXT TABLE によって作成された索引を削除するには、121ページの『テキスト表を使用不可にする』を参照してください。

### ヒント

使用可能なテキスト表を除去することにした場合には、最初にこれを使用不可にして、索引、ログ表などを除去するようしなければなりません。

## 例

次の例を実行すると、DB2TX.MYTABLE テキスト表が使用可能になります。

```
db2tx ENABLE TEXT TABLE db2tx.mytable
```

索引特性の省略時値は、テキスト構成設定値から得られます。

次の例では、テキスト表用に作成される共通索引の特性が明示的に設定されず。UNIX サーバーの場合

```
db2tx ENABLE TEXT TABLE db2tx.mytable
      INDEXTYPE linguistic
      UPDATEFREQ min(100) d(1,2,3,4,5) h(12,15) m(00)
      DIRECTORY DB2TX_INSTOWNERHOMEDIR/db2tx/indexes
```

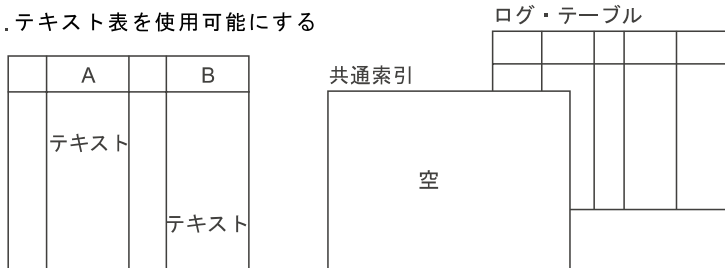
OS/2 または Windows NT または Windows 2000 サーバーの場合

## テキスト表の使用可能化 (任意選択)

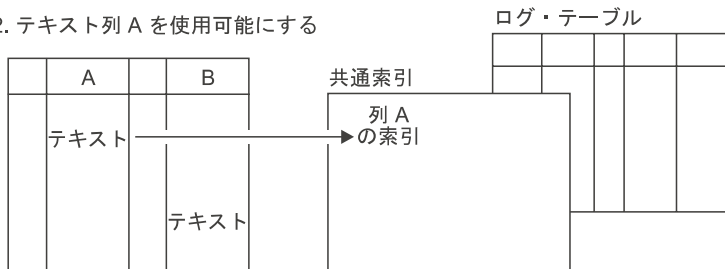
```
db2tx ENABLE TEXT TABLE db2tx.mytable
      INDEXTYPE linguistic
      UPDATEFREQ min(100) d(1,2,3,4,5) h(12,15) m(00)
      DIRECTORY d:¥dmb¥db2tx¥indexes
```

この例では、索引タイプと索引ディレクトリーを設定し、最少 100 のテキスト文書が待ち行列にある場合に月曜から金曜の 12:00 または 15:00 に索引が更新されるように索引の更新頻度を設定します。

### 1. テキスト表を使用可能にする



### 2. テキスト列 A を使用可能にする



### 3. テキスト列 B を使用可能にする

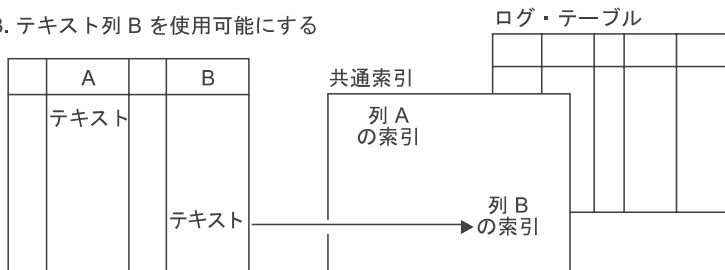
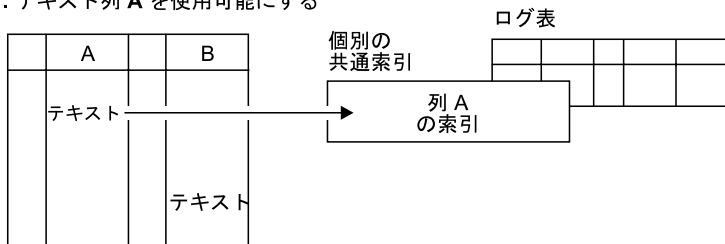


図9. 表中のすべてのテキスト列に関する共通索引の作成

## テキスト列の使用可能化

### 1. テキスト列 A を使用可能にする



### 2. テキスト列 B を使用可能にする

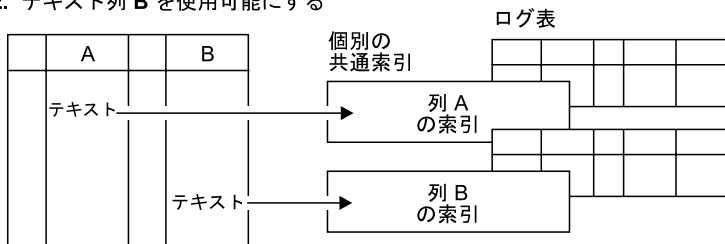


図 10. テキスト列ごとに別個の索引の作成

## テキスト列の使用可能化

### 要約

実行時	検索するテキストが含まれる列ごとに 1 度ずつ
コマンド	db2tx=>ENABLE TEXT COLUMN ... (例を参照)
権限	表に関する ALTER、SELECT、UPDATE

### ヒント

ほとんどのテキスト列について、索引タイプなどの設定を同じにする場合は、テキスト構成情報を使用して省略時設定を指定してください。21ページの『テキスト構成設定値』を参照してください。

ENABLE TEXT COLUMN によって加えられた変更を逆転させるには、DISABLE TEXT COLUMN コマンドを使用します。表中の使用可能なテキスト列をすべて使用不能にする場合は、DISABLE TEXT TABLE コマンドを使用します。

テキスト列を使用可能にする際には、表へのハンドル列の追加、文書情報（書式、言語、CCSID）の設定、ログ表の作成、および索引の作成を行います。

## ハンドル列の追加

このステップでは、DB2 テキスト・エクステンダーは表に 60 バイト VARCHAR ハンドル列を追加します。この列には、使用可能化されたテキスト列に関連するハンドルが含まれています。ハンドルには関連したテキスト列および関連した外部ファイル中のテキストに関する情報が含まれています。この情報には、固有の文書 ID、文書の言語、書式、CCSID、および索引名が含まれています。

### ヒント

このバージョンの DB2 テキスト・エクステンダーでは、これ以前のリリースを使用して作成された索引を処理できません。DB2 テキスト・エクステンダー使用可能データベースおよび表すべてを使用不可にしてから再び使用可能にしなければなりません。

DB2TX.SAMPLE

DOCID	AUTHOR	SUBJECT	DATE	COMMENT
データ	データ	データ	データ	テキスト

図 11. 使用可能にする前の DB2TX.MYTABLE 表の構造

テキスト・ブロックが含まれた列は COMMENT です。この列中のテキストを検索するには、まず DB2 テキスト・エクステンダーで使用するためのデータベースおよび COMMENT 列を作成しておかなければなりません。

こうした準備のステップを経て、DB2TX.MYTABLE 表には、次のようなハンドル用の追加列が含まれるようになります。

DB2TX.SAMPLE

DOCID	AUTHOR	SUBJECT	DATE	COMMENT	COMMENTHANDLE
データ	データ	データ	データ	テキスト	テキスト・ハンドル

図 12. 使用可能にした後の DB2TX.MYTABLE 表の構造

**注:** 以後テキストの検索を行う際は、検索する列としてテキスト列ではなくハンドル列を指定します。

## テキスト列の使用可能化

### 文書情報の設定

このテキスト列に保管する代表的なテキスト文書のタイプ、すなわちそれらの書式 (ASCII など)、言語、および CCSID を指定します。この情報の省略時値をテキスト構成設定値に指定することができます。21ページの『テキスト構成設定値』を参照してください。

### ログ表の作成

このステップで、LOGIXnnnnnn と呼ばれるログ表および視点が作成されます。ここで、IXnnnnnn は索引名です (カタログ表示から使用できます)。省略時の表スペースがテキスト構成で指定されている場合、ログ表はそこに保管されます。指定されていない場合は、DB2 システムの省略時の表スペースに保管されます。パフォーマンスおよびディスク・スペースの使用を最適化するために、ログ表に使用される別の表スペースを指定することができます。

また、トリガーも作成され、これにより、列内に文書の追加や変更が行われるたびに、ログに情報が追加されます。次に索引付けが行われるときに、この情報によって、これらの文書が索引付けされます。

外部ファイルが追加または変更される場合は、これらのトリガーは、その変更を感知しません。このような場合に、トリガーによってログ表に情報が追加されるようにするには、108ページの『外部ファイルの索引の更新』に記載されている例のように、UPDATE ステートメントを使用します。

索引付け中に、索引付けの対象となる文書が待ち行列内に見つからないといったエラーが発生した場合は、エラー事象 と呼ばれる事象がログに追加され、115ページの『エラー事象の表示』で説明するように、それを表示することができます。

#### ヒント

このステップで、ログ用スペースを使い尽くしてしまった場合は、76ページの『ラージ表でのテキスト列の使用可能化』を参照して、可能な解決方法を見つけてください。

区分データベースでは、それぞれの表が表スペースおよびノードグループに割り当てられます。ログ表が、使用可能なユーザー表と同一のノードグループに属する表スペースに割り当てられるようにするのは重要です。DB2 テキスト・エクステンダーは、これを ENABLE コマンドでチェックします。



## 索引の作成

テキスト列ごとに別々の索引を作成する、すなわち `ENABLE TEXT TABLE` ステップをスキップする場合は、DB2 テキスト・エクステンダーはこのステップの間にそれぞれのテキスト列に別々の索引を作成します。索引のタイプ、索引を更新する周期、および索引を保管するディレクトリーを指定します。このようにしないで表全体に 1 つの索引を作成する場合は、すでに `ENABLE TEXT TABLE` の実行と索引パラメーターの指定は行われているので、ここでこれらを繰り返すと無視されます。

指定のテキスト列でテキスト文書の索引付けを即時に開始するかどうか、また次回いつ周期的な索引付けをスケジュールするかを決めるには、`UPDATEINDEX` キーワードを使用します。このキーワードを使用しない場合は、テキスト構成設定値内に指定された値が採用されます。

**テキスト列に対するさまざまなタイプの索引の作成。** テキスト列には複数の索引を作成することができます。これは、たとえば、基本語形および Ngram など異なる索引タイプに列に関連させることによって、同じテキスト列に対して基本語形検索およびファジー検索を実行する場合に有用です。これを行うには、`ENABLE TEXT COLUMN` をもう一度実行し、作成する索引の追加タイプだけでなく、固有のハンドル列名も指定します。

### ヒント

区分データベースを使用しており、ノードグループの構成を変更し表の列の再配布を開始する場合には、テキスト索引とログ表を再作成しなければなりません。つまり、そのノードグループの DB2 テキスト・エクステンダー使用可能な表を使用不可にしてから再び使用可能にしなければなりません。

## 例

次の例では、`DB2TX.MYTABLE` 表中の `COMMENT` テキスト列を使用可能にし、作成されるハンドル列に `COMMENTHANDLE` という名前を割り当てます。

```
db2tx ENABLE TEXT COLUMN db2tx.mytable comment
      HANDLE commenthandle
```

テキスト情報および索引特性の省略時値は、テキスト構成設定値から得られません。

## テキスト列の使用可能化

また、次の例は COMMENT 列中の文書タイプの値を明示的に設定します。索引特性の省略時値は、テキスト構成設定値から得られます。

```
db2tx ENABLE TEXT COLUMN      db2tx.mytable  comment
        HANDLE      commenthandle
        CCSID        819
        LANGUAGE     uk_english
        FORMAT       rft
```

次の例は COMMENT 列のために作成される索引の特性の値を明示的に設定します。この例では索引タイプと索引ディレクトリーを設定し、最少 100 のテキスト文書が待ち行列にある場合に月曜から金曜の 12:00 または 15:00 に索引が更新されるように索引の更新頻度を設定します。テキスト情報の省略時値は、テキスト構成設定値から得られます。

UNIX サーバーの場合

```
db2tx ENABLE TEXT COLUMN      db2tx.mytable  comment
        HANDLE      commenthandle
        INDEXTYPE    linguistic
        UPDATEFREQ   min(100) d(1,2,3,4,5) h(12,15) m(00)
        UPDATEINDEX  UPDATE
        DIRECTORY    DB2TX_INSTOWNERHOMEDIR/db2tx/indexes
```

OS/2 または Windows NT または Windows 2000 サーバーの場合

```
db2tx ENABLE TEXT COLUMN      db2tx.mytable  comment
        HANDLE      commenthandle
        INDEXTYPE    linguistic
        UPDATEFREQ   min(100) d(1,2,3,4,5) h(12,15) m(00)
        UPDATEINDEX  UPDATE
        DIRECTORY    d:¥dmb¥db2tx¥indexes
        TABLESPACE  mytablespace
        STOGROUP     mystoragegroup
```

## ラージ表でのテキスト列の使用可能化

ラージ行長のある表で作業している場合、テキスト列を使用可能にすると、タイプ DB2TEXTH (VARCHAR 60) のハンドル列を追加することになることに注意してください。これは、表が DB2 により定められた最大の列長に近づいている場合には、重要です。

ラージ表でテキスト列を使用可能にする場合は、DB2 REORGANIZE ユーティリティーを使用して、その表を再編成する必要があるかどうかをチェックします。最初にラージ表を使用可能にするときには、以下のステップにより、索引付けを迅速に行うことができます。

1. NOUPDATE オプションを使用して、表を使用可能にする。これによって、ハンドルが作成されるが、文書の索引付けはまだ行われない。

2. DB2 REORGANIZE ユーティリティを使用して、表を再編成する。
3. UPDATE INDEX を実行して、索引を作成する。

テキスト列または外部ファイルを使用可能にすると、DB2 テキスト・エクステンダーは、表にハンドル列を追加し、ハンドル値を初期化します。これによって、DB2 ログ項目が書き込まれます。書き込まれるログ項目の数が例外的に大きいと、DB2 によってログ用スペースが使い尽くされることがあります。

この状態に対処するには次の 2 つの方法がありますが、パフォーマンス上の理由で、最初の方が望ましい方法です。

- DB2 UPDATE DB CFG コマンドを使用して、LOGPRIMARY、LOGSECOND、および LOGFILSIZ のデータベース構成パラメーターを変更することによって、使用可能なログ用スペースを増やします。経験上、以下の値が採用されます。ユーザーのインストール状況に合うようにこれらの値を変更する必要がある場合もあります。

```
LOGSECOND    50
```

LOGPRIMARY と LOGSECOND の合計が 128 以下であることを確認してください。アプリケーション・ヒープ・サイズも増やす必要があります。

```
APPLHEAPSZ   512
```

- 19ページの『構成』で説明されている COMMITCOUNT 構成パラメーターを使用して、DB2 が必ず中間 COMMIT を行うようにします。ユーザーが指定した値は、いくつの挿入ステートメントまたは更新ステートメントが出されると、DB2 テキスト・エクステンダーが DB2 コミット・ステートメントを出すかを示します。これによって、使用可能化ステップにかかる時間は増えますが、ログ表に必要なサイズは削減されます。

## サポートされないデータ・タイプのテキスト列の使用可能化

DB2 テキスト・エクステンダーで使用可能化を行うためには、テキスト列は、CHAR、GRAPHIC、VARGRAPHIC、LONG VARGRAPHIC、DBCLOB、VARCHAR LONG VARCHAR、または CLOB でなければなりません。ユーザー定義特殊タイプ (UDT) のように、これらと異なるタイプの列内の文書の場合は、ユーザー・タイプを入力とし、出力タイプとして CHAR、GRAPHIC、VARGRAPHIC、LONG VARGRAPHIC、DBCLOB、VARCHAR、LONG VARCHAR、または CLOB を提供する関数を用意する必要があります。

この関数の名前を指定するには、ENABLE TEXT COLUMN 中で FUNCTION キーワードを使用します。

例: 圧縮テキストを表に保管するとします。

## テキスト列の使用可能化

1. テキストのユーザー定義特殊タイプ (UDT) を作成します。

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

2. 表を作成し、これにテキストを挿入します。

```
db2 CREATE TABLE MYTEXT (author VARCHAR(50),
                           text COMPRESSED_TEXT)
db2 INSERT ...
```

テキスト列を DB2 テキスト・エクステンダーで使用できるようにするには、以下を実行します。

1. COMPRESSED\_TEXT タイプの値を受け取って、対応する圧縮解除テキストを (たとえば CLOB(10M) 値として) 返すユーザー定義関数 (UDF) を作成し、その UDF に、たとえば UNCOMPRESS という名前を付けます。
2. FUNCTION キーワードを使用してテキスト列を使用可能にし、UNCOMPRESS UDF を識別します。

```
db2tx ENABLE TEXT COLUMN MYTABLE text
      FUNCTION uncompress
      HANDLE handle
      ...
```

## データ・タイプ DATALINK を含む列の使用可能化

DB2 テキスト・エクステンダーは、データ・タイプ DATALINK をサポートします。DB2 のデータ・リンク・テクノロジーでは、DB2 に保管されたデータと外部ファイルに保管されたデータの間の参照保全が提供されています。この参照保全は、データベースの外部に保管されたオブジェクトへの参照を持つ新規の DATALINK データ・タイプを提供することによって行われます。DB2 に保管された DATALINK 値は、ファイルとファイル名を含むデータ・リンク・サーバーの名前を、URL の形式にコード化します。

DATALINK データが、DB2 表ではなく、外部ファイルにある場合は、79ページの『外部テキスト・ファイルの使用可能化』にある説明に従ってください。

DATALINK データ・タイプを含むテキスト列を使用可能にする前に、インストール検査のサンプルを実行することが大切です (18ページの『インストール検査のためにサンプル・データベースを準備する』を参照してください)。

**AIX の例:** 以下のステートメントは、AIX で、データ・タイプ DATALINK を使用してデータ・リンク値をテキスト列に挿入する方法を示しています。

```
db2 "insert into mytable values ('Peter',
                                d1value ('http://dlfs.boeblingen.ibm.com/test/documents/letter1',
                                           'URL', 'this is a letter from Peter'))"
```

この例では、*dlfs.boeblingen.ibm.com* はファイル・サーバー名、*test* はファイル・システム名、*documents/letter1* はパスおよびファイル名を示します。

ファイル・システムが、URL で指定したものと同名前で DB2 サーバーにマウントされるようにしてください。

**Windows NT または Windows 2000 の例:** 以下のステートメントは、Windows で、データ・タイプ DATALINK を使用してデータ・リンク値をテキスト列に挿入する方法を示しています。

```
db2 "insert into mytable ('Peter',
                           dlvalue ('unc:¥dlfs.boeblingen.ibm.com¥cdrive¥documents¥letter1',
                                     'URL', 'this is a letter from Peter'))"
```

この例では、*cdrive* はファイルが保管されるドライブの共用名を示します。

データ・リンク列に保管された文書の索引付けの際に問題が生じる場合、DB2 が提供する API を使用して DB2 サーバーから文書にアクセスできることを確認してください。

---

## 外部テキスト・ファイルの使用可能化

このオプションは、以前のリリースとの互換性を保つためだけに提供されています。DB2 表に保管されていないファイルを索引付けするときは、DATALINK データ・タイプの列を使用する方が良いでしょう。78ページの『データ・タイプ DATALINK を含む列の使用可能化』を参照してください。

要約	
実行時	任意選択。検索される外部ファイルに関連した表ごとに 1 度
コマンド	db2tx=>ENABLE TEXT FILES ... (例を参照)
権限	表に関する ALTER、SELECT、UPDATE

DB2 テキスト・エクステンダーは、DB2 表に保管されている文書だけでなく、ファイルに保管されたテキスト文書も検索することができます。この準備のステップは、外部ファイル内のテキストを検索しようとする場合に必要になります。外部テキスト・ファイルに関連した表を ENABLE TEXT TABLE コマンドによって使用可能にすることはできません。

## 外部テキスト・ファイルの使用可能化

72ページの『テキスト列の使用可能化』で説明する方法と同様に、索引の作成、ログ表の作成、および文書情報の設定を行います。

### ヒント

1. このステップで、ログ用スペースを使い尽くしてしまった場合は、76ページの『ラージ表でのテキスト列の使用可能化』を参照して、可能な解決方法を見つけてください。
2. 使用可能にしている表が区分されており、複数の物理ノードのあるノードグループを使用している場合には、それぞれの表区画が常駐するノードに、表の列で参照する外部ファイルが位置していることを確認してください。

既存の DB2 表にタイプが DB2TEXTFH のハンドル列が追加されます。ハンドル列には、外部ファイルへの参照、索引を含む各ハンドル、および文書情報 (CCSID、書式、および言語) が保持されます。

詳しくは 88ページの『外部ファイルのハンドル』を参照してください。

ユーザーは、テキスト列の使用可能化の場合と同様に、省略時索引特性といった追加のパラメーターを指定することができます。

索引が作成された後は、外部ファイルを移動または削除することができます。ファイルでの検索は依然として実行できます。表に新しい列を挿入し、UPDATE INDEX を使用して新しいファイル参照で索引を更新することができます。

使用可能にしている表が複数の物理ノードのあるノードグループを使用している場合には、表区画が常駐するノードに、表の列で参照する外部ファイルが位置していることを確認してください。

### 例

1. 少なくとも 1 つの列をもつ表 DB2TX.EXTFILE を作成するか既存の表を使用する。
2. ハンドル列 FILEHANDLES を表 DB2TX.EXTFILE に追加する。

```
db2tx ENABLE TEXT FILES db2tx.extfile
                HANDLE   filehandles
                INDEXTYPE linguistic
```

```
UPDATEFREQ min(100) d(1,2,3,4,5) h(12,15) m(00)
UPDATEINDEX NOUPDATE
DIRECTORY ¥any¥db2tx¥indices
```

### 3. ハンドルを初期化する。

- 新しい表の列ごとに次のコマンドを実行します。

```
db2 INSERT INTO db2tx.EXTFILE (FILEHANDLES)
VALUES (db2tx.INIT_TEXT_HANDLE
(850, 'TDS' 'US_ENGLISH',
'd:¥dmb¥samples¥tx')
```

- 既存の表の場合は、外部ファイルの名前を指定して、ハンドル列を更新し、外部ファイル参照を反映します。

```
db2 UPDATE db2tx.EXTFILE
SET FILEHANDLES = db2tx.file(FILEHANDLES, 'd:¥dmb¥samples¥tx')
WHERE DOCID = 'doc1'
```

#### ヒント

外部ファイルを参照するハンドル列の更新には  
INIT\_TEXT\_HANDLE を使用してはなりません。

### 4. 索引を更新する。

```
db2tx UPDATE INDEX db2tx.extfile
HANDLE filehandles
```

## セッションの終了

これで検索するテキスト文書を準備するステップが完了しました。

テキスト列を使用可能にするときに、UPDATEINDEX キーワードとして NOUPDATE を指定すると、DB2 テキスト・エクステンダーは、そのテキストをすぐには索引付けしないで、次の索引付けの周期がくるまで待ちます。今すぐ索引を更新する場合は、107ページの『索引の更新』を参照してください。

文書の索引付けが終了すれば、83ページの『第6章 検索方法』で説明されているような情報検索を開始できます。

#### ヒント

いつ索引付けが終了したかを判断するためには、GET INDEX STATUS を使用します。

## セッションの終了

DB2 テキスト・エクステンダー・コマンド・プロセッサを終了するには以下を入力します。

```
db2tx=>quit
```



---

## 第6章 検索方法

DB2 テキスト・エクステンダーには、SQL 照会へテキスト検索副照会を組み込めるようにする SQL 関数が備わっています。この種の関数は、SQL で一般的に利用できる関数に加えて用意されており、ここでは DB2 テキスト・エクステンダー関数と呼ばれます。

これらの関数の構文の詳細は、203ページの『第11章 検索関数』を参照してください。

検索を行う前に、45ページの『検索のタイプ』を読み、さらに GET INDEX SETTINGS を使用して検索するテキストに関連した索引タイプを見つけ出してください。検索結果は、索引タイプにより異なる場合があります。

この章の例では、索引タイプを基本語形索引と仮定しています。

この章では、以下の項目について説明します。

- サンプル DB2 テキスト・エクステンダー関数
- DB2TX.SAMPLE サンプル表
- 外部ファイルのハンドル
- SQL アクセス権を DB2 テキスト・エクステンダー関数に与える関数パスの設定
- CONTAINS、NO\_OF\_MATCHES、および RANK を使用するテキストの検索
- CONTAINS の例を使用した、DB2 テキスト・エクステンダー関数の検索引き数の指定
- CONTAINS と REFINE を使用した、直前の検索の詳細化
- INIT\_TEXT\_HANDLE、CCSID、FORMAT、および LANGUAGE を使用する、ハンドル内の情報の設定と抽出
- SEARCH\_RESULT を使用した、検索パフォーマンスの向上

検索関数の構文例の情報源

DB2 テキスト・エクステンダーは、txsample.udf と呼ばれるコマンド行プロセッサ入力ファイルを備えています。これには、サンプル表に対して実行される DB2 テキスト・エクステンダー検索関数の例が含まれています。このファイルは、DB2 テキスト・エクステンダーのテキスト準備および検索関数の構文や検索引き数で使用する構文の例を調べるのにご使用ください。

このファイルを実行するには、次のように入力します。

```
db2 -t -v -f txsample.udf
```

DB2TX.SAMPLE サンプル表

検索構文例の多くは、サンプル表 DB2TX.SAMPLE を使用しています。これらの例は、DB2 テキスト・エクステンダーを使用して自分で実行してみることができます。

表 DB2TX.SAMPLE を作成するには、18ページの『インストール検査のためにサンプル・データベースを準備する』を参照してください。

表 DB2TX.SAMPLE から抽出したものを表6 に示してあります。

表6. DB2TX.SAMPLE 表の例からの抜粋

DOCID	AUTHOR	SUBJECT	DATE	COMMENT
doc 5	CHGVMIC1 の RSSHERM	LIBDB2E.A エラー	1995-07-25 -20.13.59	顧客は、「No such file or directory」というエラー・メッセージをLIBDB2E.A に受け取っています。これは、非同期入出力ドライバーに関連するエラー・メッセージではないようです。この顧客は、3.2.5 でベータ 4 を使用しています。顧客に /usr/lpp/db2_02_01/lib のファイルの許可と所有権を私のものと比較してもらいましたが、現在それらは同じものになっています。この顧客の .profile および ENV も問題ないようです。あいにくこの顧客は、インストールを行って (COMMIT を実行して) いました。そのほかに何かが誤っている可能性があります。

表 6. DB2TX.SAMPLE 表の例からの抜粋 (続き)

DOCID	AUTHOR	SUBJECT	DATE	COMMENT
doc 6	SYDVM1 の EDWARDSC	DDCS/2 から の小文字のユー ザー ID お よびパスワー ド	1995-07-25 -20.15.20	再チェック後、大文字小文字の区別に 問題のあったインスタンスが、MVS への DB2/2 ゲートウェイを使用して いました。インスタンスに小文字のユ ーザー ID を渡した場合には、そのゲ ートウェイは使用できませんでした (パスワードは無関係)。大文字のユー ザー ID を実際に入力したときだけ接 続に成功しました。そこで、残念です がこれはあなたの状態の参考にはなら ないと思います。
doc 7	TOROLAB4 の SKY	ODBC およ びストア ード・プロシー ジャー	1995-07-25 -20.42.27	CLI (ODBC) を使用するストアード・ プロシージャーの使用法を説明するサ ンプル・プログラムが 2 セットあり ます。  C ファイル inpsrv2.c (サーバー上にあ る) および C ファイル inpcli2.c (ク ライアント上にある) は、入力用にスト アード・プロシージャーを使用する方 法を説明するサンプルを作成します。 ファイル outsrv2.c および outcli2.c は、出力用にストアード・プロシー ジャーを使用する方法を説明するサ ンプルを作成します。  これらのファイルは、.../sqllib /samples/cli ファイルの一部です。 MAKE ファイルは、これらのファイル を自動的に作成し、サーバーのファイ ルを正しいサブディレクトリーに転送 します。

## DB2TX.SAMPLE サンプル表

表 6. DB2TX.SAMPLE 表の例からの抜粋 (続き)

DOCID	AUTHOR	SUBJECT	DATE	COMMENT
doc 8	TOROLAB2 の ADAMACHE	DB2SYS.DLL アクセス違反	1995-07-25 -21.13.22	<p>前のベータ・バージョンをインストールしてありましたか。もしそうなら、ソフトウェア・インストーラーを使用してそれを削除しましたか。</p> <p>データベース・ディレクトリー (SQLBDBDIR および SQL00001 など) を以前のベータ・ドライバーから削除しましたか。</p>
doc 9	TOROLAB2 の ADAMACHE	CREATE DB = SYS3175: db2eng.dll 内 の db2sysc.exe	1995-07-25 -21.40.09	<p>DB2/2 ベータ・ユーザーの多くは、ソフトウェア・インストーラーを使って前バージョンのベータを削除し、ベータ 5 (現在ではゴールデン・コード) をインストールしてデータベースを作成し、SYS3175: db2eng.dll 内の db2sysc.exe を入手します。</p> <p>これは、ディレクトリー書式がベータ 4 とベータ 5 では変わっているために起こります。DB2/2 インストールの際に、ベータ・ドライバー間で sqlbdir ディレクトリーは移行されません。sqlbdir と sql000x ディレクトリーおよび %sqllib%db2%sqlbdir ディレクトリーがあればすべて除去する必要があります。</p> <p>行うべきことは、ソフトウェア・インストーラーを使用して前バージョンのベータを削除し、sqlbdir、sql000x、および %sqllib%db2%sqlbdir ディレクトリーがあればすべて除去した後で、新しいコードをインストールすることです。</p>

表 6. DB2TX.SAMPLE 表の例からの抜粋 (続き)

DOCID	AUTHOR	SUBJECT	DATE	COMMENT
doc 10	CHGVMIC1 の RSSHERM	DB2/NT - SNA サポート	1995-07-25 -22.10.15	DB2/NT は、CAE クライアントと WIN クライアント両方に対するサーバーとして、および SNA ネットワーク上の DB2/6000 か DB2/MVS (またはその両方) に対するクライアント (ホップ) として稼働可能ですか。そうでなければ、DB2/NT から、DB2/6000 または DB2/MVS (あるいはその両方) への、再度 SNA を経由する DRDA がサポートされていますか。

次に、この表の構造の一部を、最初と最後の列を記述して示します。

## DB2TX.SAMPLE

DOCID	COMMENT
doc 1	顧客は...
doc 2	再チェック後...

図 13. DB2TX.SAMPLE 表の構造

検索されるテキストが入っている列は COMMENT です。しかし、この列の中のテキストを使用して検索を行うには、ENABLE TEXT COLUMN コマンドを実行して、DB2 テキスト・エクステンダーが使用する COMMENT 列を事前に作成しておく必要があります。

こうした準備段階を経て、DB2TX.SAMPLE 表は次のようになります。

## DB2TX.SAMPLE

DOCID	COMMENT	COMMENTHANDLE
doc 1	顧客は...	X'..ハンドル..'
doc 2	再チェック後...	X'..ハンドル..'

図 14. 使用可能にした後の DB2TX.SAMPLE 表

ここで、上記の表にはハンドルに関する列が追加され、個々のテキスト・オブジェクトにそれを表す固有のハンドルが与えられています。

## DB2TX.SAMPLE サンプル表

使用可能なテキスト列に後でテキストを挿入すると、挿入トリガーがそのテキストのハンドルを作成します。

### DB2TX.SAMPLE

DOCID	COMMENT	COMMENTHANDLE
doc 1	顧客は...	X'..ハンドル..'
doc 2	再チェック後...	X'..ハンドル..'

ENABLE TEXT COLUMN  
によって作成されたハンドル

### 挿入行

doc 11	私がインストール...	X'..ハンドル..'
--------	-------------	-------------

挿入トリガーにより  
作成されたハンドル

図 15. トリガーによる挿入行のハンドル作成

ハンドルには、以下の情報が含まれます。

文書 ID

関連する索引の名前と場所

文書情報: CCSID、書式、および言語

DB2 テキスト・エクステンダーが提供する SQL 関数は、ハンドルをパラメーターとして、表の SQL 処理の一環として、テキストの保管、アクセス、検索、および処理を行います。

---

## 外部ファイルのハンドル

DB2 テキスト・エクステンダーは、DB2 表に保管されているテキストだけでなく、どこに保管されたテキスト・ファイルでも検索することができます。79 ページの『外部テキスト・ファイルの使用可能化』で、DB2 表に保管されていないテキスト文書内での検索を可能にする準備ステップを説明してあります。このステップでは、ENABLE TEXT FILES コマンドによって、外部ファイルのハンドル用にタイプ DB2TEXTFH のハンドル列が作成されます。既存の表にハンドル列が追加されます。

たとえば、作成者の名前と作成日付のための列を含む表を作成しておくことができます。

INIT\_TEXT\_HANDLE を使用して、ファイル・ハンドルを初期化します。各ハンドルには、文書 ID、関連した索引の名前と位置、および文書情報 (CCSID、書式、および言語) の他に、外部ファイルへの参照も含まれます。

## 現行の関数パスの設定

```
▶▶—SET—CURRENT FUNCTION PATH—[ = ]—DB2TX, ...—▶▶
```

DB2TX を現行パス名に追加して SQL が DB2 テキスト・エクステンダー関数を検出できるようにするには、SQL ステートメントの SET CURRENT FUNCTION PATH を使用します。これを行わない場合、関数名を明示的に修飾できます。たとえば CONTAINS という関数の場合は DB2TX.CONTAINS と入力します。

この章にある例では、修飾形式のDB2 テキスト・エクステンダー関数を使用しています。例にあるステートメントは、現行関数パスを設定しないでも、例の中で記述されているとおりに入力すれば使用できます。

### ヒント

データベースへの接続時には、現行関数パスを常に忘れずに設定してください。

## テキストの検索

```
▶▶—CONTAINS—(—handle—, —search-argument—)—▶▶
   |
   |—NO_OF_MATCHES—
   |—RANK—
```

```
▶▶—SEARCH_RESULT—(—schema—, —table—, —handle—, —search-argument—)—▶▶
```

このセクションでは、DB2 テキスト・エクステンダーに備えられている SQL 関数を使用して、テキストを含んだ DB2 データベースを検索する方法について説明します。以下の検索方法について説明します。

- 照会の作成
- テキスト文書中で検出された一致数の判別
- 検出されたテキスト文書のランクの入手

SEARCH\_RESULT の使用法は 105ページの『検索パフォーマンスの向上』で説明してあります。

## テキストの検索

この種の DB2 テキスト・エクステンダー関数はそれぞれ、テキスト索引中で検索引き数を検索します。表の中に 100 000 個のテキスト文書があるとするれば、CONTAINS、RANK、または NO\_OF\_MATCHES 関数は 100 000 回呼び出されます。ただし、テキスト索引が 100 000 回検索されることはありません。その代わり、関数の最初の呼び出し時に、検索項目を含む全文書の内部リストが作成されます。2 回目以降の関数の呼び出しでは、関係する文書がそのリスト内にあるかどうかは判別されます。

### ヒント

DB2 テキスト・エクステンダーの関数を使用して表の中を検索する場合、テキスト列ではなく関数に必ずハンドル列を渡してください。テキスト列の中を検索しようとする、SQL は次のようなメッセージで応答し、データ・タイプが誤っていることを示します。

```
No function by the name "CONTAINS" having compatible arguments was found in the function path.
```

ENABLE TEXT TABLE または ENABLE TEXT COLUMN コマンドを出した直後にテキストを検索する場合、そのコマンドにより作成中の索引がまだ存在しないことを示すエラー RC\_SE\_EMPTY\_INDEX が発生することがあります。索引の作成に要する時間は、索引を作成する文書の数や、索引の作成を実行するシステムのパフォーマンスなどの要素によって決まります。作成時間は、数分から数時間の範囲です。それで、深夜など、システムの負荷が軽い時間帯に実行するべきです。

このメッセージが出されたら、後で検索を再試行するか、GET INDEX STATUS を使用して、索引付けエラーが起きていないかをチェックします。

## 照会の実行

次の例は、CONTAINS 関数がハンドル名によって識別される文書内のテキストをどのように検索するかを示しています。テキストが検索引き数を満たす場合は 1 が戻され、満たさない場合は 0 が戻されます。

```
db2=>SELECT DATE, SUBJECT
        FROM DB2TX.SAMPLE
        WHERE DB2TX.CONTAINS (COMMENTHANDLE, "compress") = 1
```

この例では、COMMENTHANDLE 列内のハンドルにより参照されるテキストの中で compress という用語を検索します。COMMENTHANDLE 列の中のハンドルは、COMMENT テキストが索引付けされる位置を指示します。



**ヒント**

表または列中に大文字小文字混在の識別子を作成した場合、これらの名前を二重引用符で囲む必要があることを覚えておいてください。たとえば次のようにします。

```
db2=>SELECT DATE, SUBJECT
      FROM "db2tx.sample"
      WHERE DB2TX.CONTAINS (COMMENTHANDLE, "compress") = 1
```

DB2 の select ステートメントをコマンド行に指定すると、オペレーティング・システムのコマンド解析プログラムにより、コマンド・ストリングから二重引用符などの特殊文字が除去されるので、これらの特殊記号をマスクするためには、¥ 記号を使用する必要があります。たとえば次のようにします。

```
DB2 "SELECT DB2TX.file(COMMENTHANDLE)
     FROM db2tx.sample"
     WHERE DB2TX.CONTAINS (COMMENTHANDLE, '¥"compress¥") = 1
```

**検出された一致数の検索と入手**

各テキスト文書中に検索基準が何回検出されるかを判別するには、NO\_OF\_MATCHES 関数を使用してください。

```
db2=>WITH TEMPTABLE (DATE, SUBJECT, MATCHES)
      AS (SELECT DATE, SUBJECT,
                DB2TX.NO_OF_MATCHES (COMMENTHANDLE, "compress")
          FROM DB2TX.SAMPLE)
      SELECT *
      FROM TEMPTABLE
      WHERE MATCHES > 0
```

NO\_OF\_MATCHES は整数値を返します。

**検出テキスト文書のランクの検索と入手**

RANK は、検出された文書が検索基準を満たす度合いを、他の検出文書と比べて示す絶対値です。この値は、文書内で検出された一致数と、その文書サイズを示します。RANK 関数を使用することによって、検出された文書のランクを入手できます。

次に例を示します。

## テキストの検索

```
db2=>WITH TEMPTABLE (DATE, SUBJECT, RANK)
      AS (SELECT DATE, SUBJECT,
                DB2TX.RANK(COMMENTHANDLE, 'compress'))
      FROM DB2TX.SAMPLE)
SELECT *
FROM TEMPTABLE
WHERE RANK > 0
ORDER BY RANK DESC
```

RANK は、0 ～ 1 の DOUBLE 値を戻します。

---

## 検索引き数の指定

CONTAINS、NO\_OF\_MATCHES、RANK、および HANDLE\_LIST では、検索引き数を使用されます。この節では、CONTAINS 関数を使用して、DB2 テキスト・エクステンダー関数での検索引き数のさまざまな例を示します。

### 複数の用語の検索

検索引き数に複数の用語を指定することができます。複数の検索項目を結合する方法として、次のようにコンマを使ってそれらをつなぐ方法があります。

```
db2=>SELECT DATE, SUBJECT
      FROM DB2TX.SAMPLE
      WHERE DB2TX.CONTAINS (COMMENTHANDLE,
                          '("compress", "compiler", "pack", "zip", "compact")') = 1
```

このような検索引き数の書式では、それらの検索項目を含むテキストならどれでも検出されます。論理項の場合、検索項目は OR 演算子でつながります。

### ブール演算子 AND および OR を使用した検索

(98ページの『ブール演算子 NOT を使用した検索』も参照してください。)

ブール演算子「&」(AND) および「!」(OR) を使用すると、ある検索項目を別の検索項目と結合することができます。たとえば次のようにします。

```
db2=>SELECT DATE, SUBJECT
      FROM DB2TX.SAMPLE
      WHERE DB2TX.CONTAINS (COMMENTHANDLE,
                          '"compress" | "compiler"') = 1
```

ブール演算子を使用して複数の用語を結合することができます。

```
db2=>SELECT DATE, SUBJECT
      FROM DB2TX.SAMPLE
      WHERE DB2TX.CONTAINS (COMMENTHANDLE,
                          '"compress" | "compiler" & "DB2"') = 1
```

複数のブール演算子を使用すると、DB2 テキスト・エクステンダーはそれらを左から右へ評価しますが、論理 AND 演算子 (&) は論理 OR 演算子 (!) よりも優先的にバインドされます。たとえば、次の例で括弧を付けません。

```
"DB2" & "compiler" | "support" & "compress"
```

これは次のように評価されます。

```
("DB2" & "compiler") | ("support" & "compress")
```

したがって、次の例では括弧を付ける必要があります。

```
"DB2" & ("compiler" | "support") & "compress"
```

ブール演算子をコンマ区切り文字を使用して 1 つにつなげた検索項目と結合する場合は、次のようにします。

```
("compress", "compiler") & "DB2"
```

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。

```
("compress" | "compiler") & "DB2"
```

### 用語の変化形の検索

一致語句索引を使用している場合、DB2 テキスト・エクステンダーはユーザーが入力したとおりの用語を検索します。たとえば、media という用語からは「media」を含んだテキストしか検出されません。単数形の「medium」を含むテキストは検出されません。

基本語形索引を使用している場合、DB2 テキスト・エクステンダーは名詞の複数形や動詞の別の時制などの用語の変化形も検索します。

たとえば、drive という用語は、「drive」、「drives」、「driving」、「drove」、および「driven」という語を含むテキストを検出します。

### 用語の一部 (マスキング文字) の検索

マスキング文字 (または「ワイルドカード」文字と呼ぶ) を使用すると、検索をより柔軟に行うことができます。マスキング文字とは、検索項目の前部、中央部、および後部で任意指定される文字のことを指します。この文字を使用すると、検索の結果検出されるテキスト文書の数が増加します。

### ヒント

マスキング文字を使用すると、SYNONYM FORM OF キーワードを使用することはできません。

マスキング文字は、一致語句索引の場合に用語の変化形を検出するのに特に便利です。基本語形索引の場合、マスキング文字を用いて検出される変化形の多くは、マスキング文字を用いなくても検出されます。

語の断片 (ワイルドカード文字でマスクされた語) は、基本形に変形できないことに注意してください。したがって、`pass%` を検索すると、「`passes`」や「`passed`」という語は索引の中で基本形「`pass`」に変形されるので検出されません。これらの語を検出するには、`pass%` により検索しなければなりません。

DB2 テキスト・エクステンダーは、2 つのマスキング文字、パーセント (%) と下線 (\_) を使用します。

- % は、任意の個数の任意の文字を表します。次に示すのは、% をマスキング文字として検索項目の前に使用した例です。

```
db2=>SELECT DATE, SUBJECT
        FROM DB2TX.SAMPLE
        WHERE DB2TX.CONTAINS (COMMENTHANDLE, '%"name"') = 1
```

この検索項目は、「`username`」、「`file_name`」、および「`table-name`」などを含んだテキスト文書を検出します。

% で語全体を表すことも可能です。次の例では、「`graphic function`」および「`query function`」などの句を含んだテキスト文書が検出されます。

```
db2=>SELECT DATE, SUBJECT
        FROM DB2TX.SAMPLE
        WHERE DB2TX.CONTAINS (COMMENTHANDLE, '%" function"') = 1
```

- \_ は、検索用語中の 1 文字を示します。次の例では、「`CLOB`」および「`BLOB`」を含むテキスト文書を検出しています。

```
db2=>SELECT DATE, SUBJECT
        FROM DB2TX.SAMPLE
        WHERE DB2TX.CONTAINS (COMMENTHANDLE, '%"_LOB"') = 1
```

## マスキング文字をすでに含む用語の検索

「%」文字または「\_」文字を含む用語を検索したい場合は、その文字の前にいわゆるエスケープ文字を付けてください。そして、ESCAPE キーワードを使用してそのエスケープ文字を指定します。

たとえば、「10% interest」を検索するには、次のように指定します。

```
db2=>SELECT DATE, SUBJECT
      FROM DB2TX.SAMPLE
      WHERE DB2TX.CONTAINS (COMMENTHANDLE,
                            "10!% interest" ESCAPE "!") = 1
```

この例でのエスケープ文字は "!" です。

## 任意の順序での用語の検索

次の例で示されているように「hard disk」を検索すると、使用している索引タイプにかかわらず、この 2 つの用語が隣接してその順序で並んでいる場合のみ検出されます。

```
db2=>SELECT DATE, SUBJECT
      FROM DB2TX.SAMPLE
      WHERE DB2TX.CONTAINS (COMMENTHANDLE, "hard disk") = 1
```

たとえば、「data disks and hard drives」のように任意の順序で並んだ用語を検索するには、コンマを使って用語を分離してください。

```
db2=>SELECT DATE, SUBJECT
      FROM DB2TX.SAMPLE
      WHERE DB2TX.CONTAINS (COMMENTHANDLE, ("hard", "disk")) = 1
```

## 同じ文または段落中の用語の検索

次に示すのは、指定した複数の検索項目が同じ文中にあるテキスト文書を検出する検索引き数の例です。

```
db2=>SELECT DATE, SUBJECT
      FROM DB2TX.SAMPLE
      WHERE DB2TX.CONTAINS (COMMENTHANDLE,
                            "compress" IN SAME SENTENCE AS "decompress") = 1
```

また、2 つ以上の語を一緒に検索することもできます。次の例では、同じ段落にいくつかの用語が見つかる文書が検索されます。

```
db2=>SELECT DATE, SUBJECT
      FROM DB2TX.SAMPLE
      WHERE DB2TX.CONTAINS (COMMENTHANDLE,
                            "compress" IN SAME PARAGRAPH AS "decompress"
                            AND "encryption") = 1
```

## 検索引き数の指定

### 構造化文書のセクションでの用語の検索

次に示すのは、検索項目 Williams が、構造化文書のセクション play のサブセクション author 中にあるテキスト文書を検出する検索引き数の例です。文書構造は、文書モデル・ファイルで記述されたモデル play によって指定されます。詳細については、53ページの『構造化文書 (セクション・サポート) での作業』を参照してください。

```
db2=>SELECT DATE, SUBJECT
        FROM DB2TX.SAMPLE
        WHERE DB2TX.CONTAINS (COMMENTHANDLE,
                               'MODEL play SECTIONS (play/author) "williams") = 1
```

### 用語の同義語の検索

基本語形索引または併用索引の場合、指定の検索項目だけでなく、類似した意味を持つ語も探すことにより、より柔軟な検索が可能になります。たとえば、「book」という語を検索する場合、その同義語も検索すると便利があります。これを行うには、次のように指定します。

```
db2=>SELECT DATE, SUBJECT
        FROM DB2TX.SAMPLE
        WHERE DB2TX.CONTAINS (COMMENTHANDLE, 'SYNONYM FORM OF "book"') = 1
```

SYNONYM FORM OF を使用すると、用語の同義語が論理 OR 演算子でつながれているとみなされます。つまり、検索引き数が次のように解釈されます。

```
"book" | "article" | "volume" | "manual"
```

同義語は、DB2 テキスト・エクステンダーに備えられている辞書の中にあります。同義語用を使用される省略時の辞書は常に US\_ENGLISH で、テキスト構成設定値で指定された言語ではありません。

別の言語を指定して特定の照会用に辞書を変更することができます。次に例を示します。

```
db2=>SELECT DATE, SUBJECT
        FROM DB2TX.SAMPLE
        WHERE DB2TX.CONTAINS (COMMENTHANDLE,
                               'SYNONYM FORM OF UK_ENGLISH "programme"') = 1
```

#### ヒント

SYNONYM キーワードは、検索項目の中にマスキング文字がある場合や、NOT が検索引き数と共に使用されている場合には使用できません。

## 基本語形検索の実行

DB2 テキスト・エクステンダーには、ユーザー指定の検索項目に基づいた検索を行うための、高性能の言語的な処理があります。言語的な機能は、索引が基本語形索引である場合に適用されます。言語関数についての説明は 263 ページの『第15章 基本語形索引と一致語句索引の言語的な処理』に記載しています。

これはたとえば、「utilities」などの複数形の語を検索して、「utility」を検出するような場合です。この複数形は、検索前に、英語の辞書を使用して、基本形である utility に変形されます。

しかし、英語の辞書には、他の言語の用語の変化形を基本形に戻すための情報がありません。他言語の用語の複数形を検索するには、その言語の辞書を使用する必要があります。

たとえば、GERMAN を指定した場合は、「geflogen」(「飛ぶ」の過去分詞)を検索し、その基本形「fliegen」(飛ぶ)のすべてのバリエーションを検出します。つまり、「geflogen」だけでなく、「fliege」、「fliegt」なども検出されます。

```
db2=>SELECT DATE, SUBJECT
      FROM DB2TX.SAMPLE
      WHERE DB2TX.CONTAINS (COMMENTHANDLE,
                            'STEMMED FORM OF GERMAN "geflogen") = 1
```

### ヒント

米国英語以外の文書で検索を行う際は、省略時値の言語に関係なく 検索引き数で言語を指定してください。

検索項目の変化形ではなく基本形を常に指定する場合は、言語を指定する必要はありません。

その理由を理解するには、データベース内のテキストが索引付けされる際に何が起こるかを考えてください。基本語形索引または併用索引を使用している場合、用語の変化形は基本形に変形してから索引に保管されます。つまり、DB2TX.SAMPLE 表を例にとると、「decompress」という語が COMMENT 列の最初の項目にあり、「decompression」という語が 2 番目の項目にある場合でも、索引には基本形「decompress」しか含まれず、しかもこの用語 (またはその変化形) が両方の項目にあるものとして識別されます。

その後、基本形「decompress」を検索すると、その変化形がすべて検出されます。しかし、「decompression」などの変化形を検索しても、その語を直接検出

## 検索引き数の指定

することはできません。適切な検索用の辞書を指定する必要があるため、変化形は最初にその基本形に変換されます。

### ブール演算子 NOT を使用した検索

ブール演算子 NOT を使用すると、特定のテキスト文書を検索から除外することができます。たとえば次のようにします。

```
("compress", "compiler") & NOT "DB2"
```

この場合、「DB2」という用語を含んだテキスト文書はすべて、「compress」または「compiler」の検索からは除外されます。

NOT 演算子は、95ページの『同じ文または段落中の用語の検索』で説明した IN SAME SENTENCE AS または IN SAME PARAGRAPH AS と組み合わせて使用したり、96ページの『用語の同義語の検索』で説明した SYNONYM FORM OF とともに使用することはできません。

NOT 演算子は、基本検索語 (search-primary) としてしか使用できません。つまり、&、|、および NOT 演算子を自由に組み合わせることはできません (223ページの『検索引き数構文』を参照してください)。

NOT の誤った使用例:

```
NOT("compress" & "compiler")
```

正しい使用例:

```
NOT("compress" , "compiler")
```

### ファジー検索

「ファジー」検索では、検索項目と類似したつづりの語が検索されます。これは、Ngram 索引についても適用可能です。

たとえば次のようにします。

```
db2=>SELECT DATE, SUBJECT
      FROM DB2TX.SAMPLE
      WHERE DB2TX.CONTAINS (COMMENTHANDLE,
                           'FUZZY FORM OF 2 "compress"') = 1
```

この検索では、つづりが誤りである語 compress を検出することができます。

この例の一致レベル“2”は正確度を示します。これは5レベルまでサポートされますが、レベル1は最も正確度が低く、約20パーセントの一致を示し、レベル5は最も高い正確度で、約90パーセントの一致を示します。光學式文



字認識装置または音声による入力によく起こることがありますが、文書内につづりの誤りがあると考えられる場合は、ファジー検索を使用します。

### 語句の境界の考慮

韓国語用に「Bound」検索が開発されました。これは、DB2 テキスト・エクステンダーが検索時に、必ず語の境界を考慮するようにするものです。たとえば次のようにします。

```
db2=>SELECT DATE, SUBJECT
      FROM DB2TX.SAMPLE
      WHERE DB2TX.CONTAINS (COMMENTHANDLE,
                           'BOUND "korean-expression") = 1
```

### 発音の似た語の検索

「Sound」検索では、検索引き数と発音の似た語を検出します。これは、発音は似ているが、つづりが異なる語が文書に含まれている可能性があるときに役立ちます。たとえば、my-er と発音するドイツ語の名前にはいくつかのつづりがあります。

```
db2=>SELECT DATE, SUBJECT
      FROM DB2TX.SAMPLE
      WHERE DB2TX.CONTAINS (COMMENTHANDLE,
                           'SOUNDS LIKE "Meyer") = 1
```

この検索では、「Meyer」、「Mayer」、および「Maier」が検出できます。

### シソーラス検索

シソーラス検索は、DB2 テキスト・エクステンダーのもう 1 つの高機能な検索項目拡張関数です。検索される追加の用語は、ユーザー自身が作成したシソーラスから入手されるので、それらを直接制御できます。たとえば、「database」を検索して、「repository」や「DB2」といった用語を検出できます。

この検索タイプは、頻繁に検索を行う、関心のある特定の領域向けのものです。つまり、きわめて有効な検索結果を生成するために類語事典を作成する時間をかける価値がある領域です。

シソーラスの作成方法の詳細と説明については、275ページの『シソーラスの概念』を参照してください。276ページの図18の例は、データベースの対象のシソーラスからの小規模な抜粋です。これは、シソーラスの拡張を使用する構文を示す次の例で使用されます。

## 検索引き数の指定

この例は、「object relational database management system」という用語を選び、それにシソーラス「myterms」で検出されるこの用語のすべてのインスタンスを追加して拡張します。ここでは、「DB2」が検索に追加されます。

```
db2=>SELECT DATE, SUBJECT
        FROM DB2TX.SAMPLE
        WHERE DB2TX.CONTAINS (COMMENTHANDLE,
                               'THESAURUS "myterms"
                               EXPAND "INST"
                               TERM OF "object relational database management system") = 1
```

次の例は、「document management system」という用語を選び、それにすべての同義語を追加して、それを拡張します。ここでは、1つの同義語である「library」が追加されます。

```
db2=>SELECT DATE, SUBJECT
        FROM DB2TX.SAMPLE
        WHERE DB2TX.CONTAINS (COMMENTHANDLE,
                               'THESAURUS "myterms"
                               EXPAND "SYN"
                               TERM OF "document management system") = 1
```

## フリー・テキストと混成検索

「フリー・テキスト検索」では、検索項目が自由書式テキストで表されます。句または文が、検索の対象を自然言語で記述しています。フリー・テキスト照会における語順は関連がありません。さらに、語いの類縁性がサポートされません。これは検索では、特定の最小頻度、かつ特定の最短距離が指定されたフリー・テキスト照会用語に出現する語と文書集合内に出現する語のあるペアです。英語の文書の最短距離は5語です。

フリー・テキスト引き数内の検索ストリングに対しては、文字または語のマスクングがサポートされないことに注意してください。

たとえば次のようにします。

```
db2=>SELECT DATE, SUBJECT
        FROM DB2TX.SAMPLE
        WHERE DB2TX.CONTAINS (COMMENTHANDLE,
                               'IS ABOUT "everything related to AIX installation") = 1
```

混成検索は、ブール検索とフリー・テキスト検索の組み合わせです。たとえば次のようにします。

```
db2=>SELECT DATE, SUBJECT
        FROM DB2TX.SAMPLE
        WHERE DB2TX.CONTAINS (COMMENTHANDLE,
                               '"DB2" & IS ABOUT "everything related to AIX installation") = 1
```

## 直前の検索の詳細化

検索引き数によって検出されるものが多過ぎるときは、最初の検索引き数と 2 番目の検索引き数をブール AND の関係で組み合わせて検索範囲を狭くする、つまり検索を詳細化すると便利な場合がよくあります。

検索結果の詳細化は、その結果を表に保管し、次にその表と対照して検索を行うことにより、REFINE 関数を使わなくても行うことができます。しかし、この方式は、修飾用語の数によっては最新の検索引き数を保管して REFINE を使用する方式よりも効果的でない場合があります。

以下のステップでは、検索後に REFINE 関数を使用してその結果を詳細化する方法を示します。REFINE 関数は、2 つの入力パラメーターをブール AND で組み合わせた検索引き数を戻します。REFINE により戻される検索引き数の組み合わせは、LONG VARCHAR タイプの値となります。

1. 古い検索引き数用の表を作成します。

すでに実行した検索の検索引き数を保持するには、表 PREVIOUS\_SEARCHES を作成してください。

```
db2=>CREATE TABLE PREVIOUS_SEARCHES (step INT,
                                     searchargument LONG VARCHAR)
```

### PREVIOUS\_SEARCHES

STEP	SEARCHARGUMENT
------	----------------

2. 最初の検索引き数を検索します。

サンプル表の中で「compress」という語を検索します。

```
db2=>SELECT COMMENT
      FROM DB2TX.SAMPLE
      WHERE DB2TX.CONTAINS (COMMENTHANDLE, "compress") = 1
```

その後のステップで使用するために、検索引き数を PREVIOUS\_SEARCHES 表に挿入してください。

```
db2=>INSERT INTO PREVIOUS_SEARCHES
      VALUES (1, "compress")
```

## 直前の検索の詳細化

### PREVIOUS\_SEARCHES

STEP	SEARCHARGUMENT
1	"compress"

### 3. 検索を詳細化します。

検索により戻されるテキスト文書が多過ぎると仮定し、REFINE 関数を使用して直前の検索項目を「compiler」という語と組み合わせて検索を詳細化します。

```
db2=>WITH LAST_STEP(STEP_MAX)
      AS (SELECT MAX(STEP)
          FROM PREVIOUS_SEARCHES),
      LAST_SEARCH(LAST_SEARCH)
      AS (SELECT SEARCHARGUMENT
          FROM PREVIOUS_SEARCHES, LAST_STEP
          WHERE STEP = STEP_MAX)
SELECT COMMENT
      FROM DB2TX.SAMPLE, LAST_SEARCH
      WHERE DB2TX.CONTAINS(COMMENTHANDLE,
                           DB2TX.REFINE(LAST_SEARCH, "compiler")) = 1
```

その後のステップで使用するために、詳細化された検索引き数を PREVIOUS\_SEARCHES 表に挿入してください。

```
db2=>INSERT INTO PREVIOUS_SEARCHES
      WITH LAST_STEP(STEP_MAX)
      AS (SELECT MAX(STEP)
          FROM PREVIOUS_SEARCHES)
SELECT STEP_MAX+1, DB2TX.REFINE(SEARCHARGUMENT, "compiler")
      FROM PREVIOUS_SEARCHES, LAST_STEP
```

### PREVIOUS\_SEARCHES

STEP	SEARCHARGUMENT
1	"compress"
2	"compress" & "compiler"

このステップは、検出されるテキスト文書数が十分少なくなるまで繰り返すことができます。

## ハンドル内の情報の設定と抽出

ハンドルには、CCSID、書式、およびそれらのテキスト文書の言語が含まれます。外部ファイルのハンドルにはさらに、その外部ファイルへのポインターが含まれます。このようなハンドルは、テキスト列または外部ファイルを使用可能にすると作成されます。

以下に説明する DB2 テキスト・エクステンダー関数を使用すると、ハンドル内のテキスト情報の設定や変更を行うことができます。

### 新規テキストを挿入する際のテキスト情報の設定

```

▶▶—INIT_TEXT_HANDLE—┌(—format—,—language—)—————▶▶
                       └(—CCSID—,—format—,—language—,—filename—)┘
  
```

ENABLE TEXT COLUMN コマンドを実行して、すでにテキストを含むテキスト列を使用可能にする場合は、テキストの書式と言語を、テキスト構成設定で指定された値に暗黙的に設定することができます。この書式と言語の設定は、ハンドルの中に保管されます。これとは別の書式と言語の値を指定したい場合は、ENABLE TEXT COLUMN コマンドで明示的に指定することができます。

ENABLE TEXT FILES コマンドを実行するときに、文書の CCSID と位置も設定できます。

後でテキストを含んだ行を挿入すると、挿入トリガーがハンドルを作成し、そのテキストの書式と言語を、テキスト列が使用可能なときに使われていた値に設定します。

書式と言語をこれらとは異なる値に設定するには、INSERT コマンドで INIT\_TEXT\_HANDLE 関数を使用してください。INIT\_TEXT\_HANDLE 関数は、行が挿入されるときに、指定された言語と書式を含む、部分的に初期化されたハンドルを作成します。次に、挿入トリガーがハンドル内に別の値を書き込みます。

次の例では、INIT\_TEXT\_HANDLE が、初期化されたハンドルの言語および書式を事前設定します。INSERT コマンドは、このハンドルを COMMENTHANDLE 列に置きます。

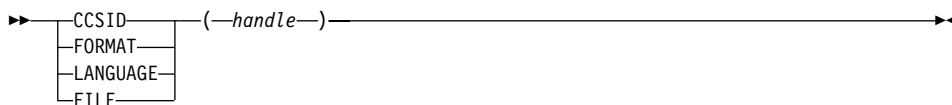
```

db2=>INSERT INTO DB2TX.SAMPLE (DOCID, COMMENT, COMMENTHANDLE)
      VALUES ('doc 101',
              'I have installed...',
              DB2TX.INIT_TEXT_HANDLE('AMI', 'GERMAN') )
  
```

## ハンドル内の情報の設定と抽出

INIT\_TEXT\_HANDLE によって返される値は DB2TEXTH タイプか DB2TEXTFH タイプです。

### ハンドルからの情報の抽出



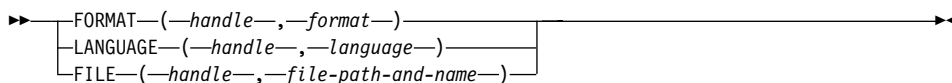
次の例は、ハンドルから CCSID を抽出します。

```
db2=>SELECT DISTINCT DB2TX.CCSID(COMMENTHANDLE)
      FROM DB2TX.SAMPLE
```

同じ方法を使って、テキスト文書の書式や言語、または外部ファイルの位置をハンドルから抽出することができます。次に示す内容は、FORMAT 関数の使用法について説明した例です。この関数は、ASCII (TDS) 文書の数サンプル表の中に戻します。

```
db2=>SELECT COUNT(*)
      FROM DB2TX.SAMPLE
      WHERE DB2TX.FORMAT(COMMENTHANDLE) = 'TDS'
```

### ハンドル内の情報の変更



FORMAT および LANGUAGE 関数は、ハンドル内のそれぞれに対応する仕様を変更することもできます。これらの関数は、変更されたハンドルを DB2TEXTH タイプまたは DB2TEXTFH タイプの値として戻します。

次の例では、テキスト文書の言語設定を変更する方法について示します。

```
db2=>UPDATE DB2TX.SAMPLE
      SET COMMENTHANDLE = DB2TX.LANGUAGE(COMMENTHANDLE, 'FRENCH')
      WHERE ...
```

LANGUAGE 関数を再び使用すると、変更が行われたことを確認することができます。

```
db2=>SELECT DISTINCT DB2TX.LANGUAGE(COMMENTHANDLE)
      FROM DB2TX.SAMPLE
```

## 検索パフォーマンスの向上

SEARCH\_RESULT 関数では、DB2 の表の値に基づく関数の概念が活用されています。関数は、SQL ステートメントの FROM 文節内で使用されて、指定された検索ストリングの検索結果の入った中間表を戻します。検索ストリングの構文は、221ページの『第12章 検索引き数の構文』で説明するものと同じです。CONTAINS または RANK と比較した場合のこの関数の利点は、ラージ表が含まれている場合にかなりパフォーマンスの向上が見られることです。

戻される表の構造は次のとおりです。

列名	データ・タイプ
HANDLE	DB2TX.DB2TEXTH, DB2TX.DB2TEXTFH
NUMBER_OF_MATCHES	INTEGER
RANK	DOUBLE

例:

```
db2=>SET CURRENT FUNCTION PATH = db2tx
db2=>SELECT author
      FROM db2tx.sample T1,
      table(DB2TX.SEARCH_RESULT('DB2TX','SAMPLE','MYDOCHANDLE','compress')) T2
```

T1.handle = T2.mydochandle です。

現行パスを設定していない場合、例の最後の行は以下ようになります。

```
where cast(T1.handle as varchar(60)) = cast(T2.mydochandle as varchar (60))
```

SELECT NUMBER\_OF\_MATCHES,RANK,HANDLE では、3つの項目がすべて戻されますが、3つの項目を任意の組み合わせで指定することもできます。たとえば、RANK を省略して、それに関連した集中処理を回避することもできます。

HANDLE 値だけがが必要な場合は、単に SELECT COUNT(\*) を使用することができます。

## 変更の要約



---

## 第7章 管理

この章では、テキスト索引の保守方法、役立つ情報の入手方法、およびテキスト準備プロセスの逆処理方法について説明しています。

---

### テキスト索引の保守

保守タスクは以下のとおりです。

- 索引の更新
- 索引の設定の変更
- 索引状況のリセット
- 索引事象の削除
- 索引の再編成

これらのタスクは、いつでもどの順序でも実行することができます。

### 索引の更新

#### 要約

**実行時** 周期的な索引付けが行われるのを待たずに、即時に索引を更新する必要がある場合。(周期的な索引付けに関する情報は、72ページの『テキスト列の使用可能化』を参照してください。)

#### コマンド

```
UPDATE INDEX
```

**権限** 表に関する ALTER、SELECT、UPDATE

次の例は、共通索引表に関する索引を更新します。

```
db2tx UPDATE INDEX db2tx.mytable
```

次の例は、複数の索引表の列に関する索引を更新します。

```
db2tx UPDATE INDEX db2tx.mytable HANDLE commenthandle
```

## テキスト索引の保守

次の周期的な索引付けが自動的に行われるのを待たずに索引を即時に更新する場合に、このコマンドを使用してください。この方法は、データベースに複数のテキスト文書を追加した直後にこれらの文書を検索する場合に便利です。

DB2 テキスト・エクステンダーは、挿入または変更が加えられたこの列 (または表の中のすべての列) のテキスト文書を索引付けし、削除された文書の用語を索引から除去します。索引に関連したログ表には、挿入、更新、および削除が行われた文書に関する情報が含まれています。

### 外部ファイルの索引の更新

文書を同じ絶対パス名をもつ新しいバージョンで置き換えたりすることなどにより、ユーザーに索引付けされている可能性のあるどの外部ファイル (79ページの『外部テキスト・ファイルの使用可能化』) の変更に関する情報も自動的にログ表に入れられることはありません。DB2 テキスト・エクステンダーは、ログ表内で、このようなファイルに対する更新を監視することはできません。このような更新は、DB2 の効力範囲外で起きるからです。

外部ファイルにおける更新が DB2 テキスト・エクステンダー索引に反映されるようにするには、以下のことを実行できます。

1. 「変更」項目に対応するハンドル列で、実際には何も行わない更新ステートメントを出すことによって、その「変更」項目をログ表に強制的に入れる。

```
UPDATE table
SET filehandlecol = filehandlecol
WHERE DB2TX.FILE(filehandlecol) = filename
```

ここで、*filename* は更新された外部ファイルの絶対パス名です。

2. UPDATE INDEX を実行して索引を更新し、外部ファイルに適用された変更を組み込む。

### 索引の設定の変更

#### 要約

**実行時** 索引の更新頻度を変更する場合。

#### コマンド

```
CHANGE INDEX SETTINGS
```

**権限** 表に関する ALTER、SELECT、UPDATE

このコマンドを使用して、索引の更新頻度を変更してください。

### 更新頻度

詳細は、51ページの『索引の更新』を参照してください。更新頻度を指定しない場合は、現行の設定はこのまま変更されません。

テキスト文書の索引付けを即時に開始するかどうか、また次回いつ周期的な索引付けをスケジュールするかを決めるには、`UPDATEINDEX` キーワードを使用してください。このキーワードを使用しない場合は、現行の設定はこのまま変更されません。

### 例

最少 100 のテキスト文書が待ち行列にある場合に月曜から金曜の 12:00 または 15:00 に索引が更新されるよう索引の更新頻度を変更するには、次のようにします。

```
db2tx CHANGE INDEX SETTINGS db2tx.mytable
      HANDLE      commenthandle
      UPDATEFREQ min(100) d(1,2,3,4,5) h(12,15) m(00)
```

索引の周期的な更新を停止するには次のようにします。

```
db2tx CHANGE INDEX SETTINGS db2tx.mytable
      HANDLE      commenthandle
      UPDATEFREQ none
```

## 索引状況のリセット

### 要約

**実行時** 索引をこれ以上検索または更新できない場合

### コマンド

```
RESET INDEX STATUS
```

**権限** なし

索引の検索や更新ができない状態が生じる場合があります。113ページの『索引状況の表示』では、このような事象が生じた場合の判別方法を説明しています。`RESET INDEX STATUS` は、索引を再度使用できるように再活動化します。

次の例は、共通索引表の索引の状況をリセットします。

```
db2tx RESET INDEX STATUS db2tx.mytable
```

## テキスト索引の保守

構文を使用すると、特定のテキスト列の索引状況をリセットできます。次の例は、複数索引表列の索引の状況をリセットします。

```
db2tx RESET INDEX STATUS db2tx.mytable HANDLE commenthandle
```

## 索引事象の削除

### 要約

**実行時** 索引のログ表内のメッセージがもう必要なくなった場合

### コマンド

```
DELETE INDEX EVENTS
```

**権限** なし

何らかの原因で索引の検索や更新ができない場合や、文書が索引付けできない場合、この状態は索引付け事象と呼ばれます。索引付け事象に関する情報は、その索引のログ表の中に保管されます。これは、問題の原因を判別する際に役立ちます。これらのメッセージは、必要なくなったときには削除できます。

次の例は、共通索引表からメッセージを削除します。

```
db2tx DELETE INDEX EVENTS db2tx.mytable
```

構文を使用しても、特定のテキスト列の索引付け事象を削除できます。次の例は、複数索引表列の索引に対するメッセージを削除します。

```
db2tx DELETE INDEX EVENTS db2tx.mytable HANDLE commenthandle
```

## 索引の再編成

### 要約

**実行時** GET INDEX STATUS で、索引を手操作で再編成しなければならないことが示される場合

### コマンド

```
REORGANIZE INDEX
```

**権限** なし

テキスト列が頻繁に更新される場合、索引の検索効率が下がる可能性があります。検索の効率を再び上げるには、索引を再編成する必要があります。DB2 テキスト・エクステンダーは索引を再編成する必要を認識し、バックグラウンドで自動的に再編成を行いますが、REORGANIZE INDEX を使用して手操作で再編成しなければならない場合もあります。索引を再編成する必要があるかどうかを調べるには、コマンド GET INDEX STATUS を使用します。

REORGANIZE INDEX の実行中には索引の検索は行えますが、索引の更新は行えません。

次の例は、共通索引表の索引を再編成します。

```
db2tx REORGANIZE INDEX db2tx.mytable
```

次の例は、複数索引表列の索引を再編成します。

```
db2tx REORGANIZE INDEX db2tx.mytable HANDLE commenthandle
```

---

### 有用な情報の入手

この説では、以下のものに関する情報を表示するためのコマンドについて説明します。

- データベース、表、列、およびファイルの使用可能状況
- 環境変数の設定
- テキスト構成設定値
- 索引状況
- エラー事象
- 索引設定値
- 列のテキスト設定値

### 使用可能状況情報の表示

#### 要約

**実行時** データベース、表、およびテキスト列、または外部ファイルの使用可能状況についての情報が必要な場合。

#### コマンド

GET STATUS

**権限** なし

## 有用な情報の入手

次のように入力します。

```
db2tx GET STATUS
```

次の内容は GET STATUS によって表示される出力の例です。これは、データベースの使用可能状況とデータベースに含まれる使用可能表、テキスト列、またはテキスト・ファイルの状況を示します。

```
Database is enabled for Text Extender

Table DB2TX.MYTABLE is enabled as a common-index table

Table DB2TX.MYTABLE is enabled as a common-index table
TextColumnName      HandleColumnName
-----
COMMENT              COMMENTHANDLE

Table DB2TX.TEST is enabled as a multi-index table
TextColumnName      HandleColumnName
-----
ABSTRACT1            ABSTRACT1HANDLE
ABSTRACT2            ABSTRACT2HANDLE
```

## 環境変数の設定値の表示

### 要約

**実行時** 環境変数の設定に関する情報が必要な場合

### コマンド

```
GET ENVIRONMENT
```

**権限** なし

次のように入力します。

```
db2tx GET ENVIRONMENT
```

次の内容は GET ENVIRONMENT によって表示される出力の例です。この出力により DB2 テキスト・エクステンダー環境変数の現行設定が表示されます。

```
Current database manager instance (DB2INSTANCE) = db2
Default database name             (DB2DBDFT) = my_database
Instance name                      (DB2TX_INSTOWNER) = user1
Instance directory                 (DB2TX_INSTOWNERHOMEDIR) = /usr/instance1
```

## テキスト構成設定値の表示

### 要約

**実行時** テキスト、索引、および処理情報の省略時設定値が必要な場合

### コマンド

GET TEXT CONFIGURATION

**権限** なし

これらの設定値については、21ページの『テキスト構成設定値』で説明しています。これらを変更するには、22ページの『テキスト構成の変更』を参照してください。

テキスト構成を表示するには、次のように入力します。

```
db2tx GET TEXT CFG
```

次の内容は GET TEXT CONFIGURATION によって表示される出力の例です。この出力により、現行のテキスト構成設定値が示されます。

```
Coded character set ID      (CCSID) = 850
Language                    (LANGUAGE) = US_ENGLISH
Format                      (FORMAT) = TDS

Index type                  (INDEXTYPE) = LINGUISTIC
Update frequency           (UPDATEFREQ) = NONE
Index directory             (DIRECTORY) = user1/db2tx/indexes

Update index option        (UPDATEINDEX) = UPDATE
Commit count               (COMMITCOUNT) = 10 000
Tablespace name            (TABLESPACE) = TXLOG
```

## 索引状況の表示

### 要約

**実行時** 索引が検索可能か、更新可能かを判別する必要がある場合

### コマンド

GET INDEX STATUS

**権限** なし

## 有用な情報の入手

索引の検索や更新ができない状態が生じる場合があります。このような状態では、メッセージが索引のログ表内に保管され、原因を判別するのに役立ちます。したがって、索引の状況を検査して、参照可能なメッセージの有無を調べるのに利用できます。

次の例は、共通索引表の索引の状況を表示します。

```
db2tx GET INDEX STATUS db2tx.mytable
```

構文を使用すると、特定のテキスト列の索引状況を表示できます。次の例は、複数索引表列の索引の状況を入力します。

```
db2tx GET INDEX STATUS db2tx.mytable HANDLE commenthandle
```

次の内容は GET INDEX STATUS によって表示される出力の例です。

```
Node 1
Search status           = Search available
Update status          = Update available
Reorg status           = started 13.55
Scheduled documents    = 0
Indexed documents      = 187000
Primary index documents = 130000
Secondary index documents = 57000
Error events           = No error events
```

索引がいくつかのノードに分割される場合には、ノード情報はノードごとに表示されます。

### 検索状況 (Search status)

指定されたハンドル列を使用して索引を検索できるかどうかを示します。検索が可能でない場合は、示されている理由コードをチェックし、その状態が起きた原因に関する詳細な情報がないかを調べてから、RESET INDEX STATUS を使用して、その索引を再度処理できるようにします。331ページの『第20章 エラー事象理由コード』を参照してください。

### 更新状況 (Update status)

指定された表または列の索引を更新できるかどうかを示します。索引更新関数が利用可能でない場合は、示されている理由コードをチェックして、その状態が起きた原因に関する詳細な情報がないかを調べてから、RESET INDEX STATUS を使用して、その索引を再度処理できるようにします。

### 再編成状況 (Reorg status)

指定された表または列の索引を再編成できるかどうかを示します。再編成が可能でない場合は、示されている理由コードをチェックし、その状



態が起きた原因に関する詳細な情報がないかを調べます。再編成ができない理由として一般的なのは、現在索引を更新中であることです。

### スケジュールされた文書 (Scheduled documents)

索引付けのために (あるいは、索引からの削除のために) 待ち行列内にリストされた文書の数を示します。

### 索引付け文書 (Indexed documents)

すでに、スケジュール済み文書の待ち行列から索引付けされている文書の数を示します。

#### 1 次索引文書 (Primary index documents)

1 次索引内の文書の数を示します。

#### 2 次索引文書 (Secondary index documents)

2 次索引内の文書の数を示します。

### エラー事象 (Error events)

索引のログ表内で使用可能な事象の数を示します。この情報は、『エラー事象の表示』の説明にしたがって表示することができます。この情報が必要なくなったら、110ページの『索引事象の削除』の説明に従って、削除することができます。

## エラー事象の表示

索引付けの間に、索引付け用にスケジュールされた文書が見つからないといった問題が生じた場合は、エラー事象 と呼ばれる事象が索引のログ表に書き込まれます。

事象戻りコードについては、331ページの『第20章 エラー事象理由コード』に説明してあります。

このエラー事象は、db2tx.LOGIXnnnnnn と呼ばれるログ表の視点でアクセスすることができます。ここで、IXnnnnnn は索引の名前で、カタログ表示から入手できます。

索引の名前を得るためには、次のように入力します。

```
DB2 SELECT TABLENAME,
          HANDLENAME,
          INDEXNAME
FROM     DB2TX.TEXTCOLUMNS
```

エラー事象視点のレイアウトは次のとおりです。

## 有用な情報の入手

UPDATESTATUS	SMALLINT
EVENTREASON	INTEGER
EVENTMESSAGE	VARCHAR(1024)
UPDATETIME	TIMESTAMP
HANDLE	DB2TEXTH or DB2TEXTFH
NODENUMBER	INTEGER

次の例は、索引ログから情報をアクセスする方法を示します。

```
DB2 SELECT EVENTREASON,  
           EVENTMESSAGE,  
           UPDATETIME,  
           HANDLE  
FROM   DB2TX.LOGIXNNNNNN
```

## 索引設定の表示

### 要約

**実行時** 索引の設定に関する情報が必要な場合

### コマンド

```
GET INDEX SETTINGS
```

**権限** なし

次の例は、共通索引表の索引の設定を入手します。

```
db2tx GET INDEX SETTINGS db2tx.mytable
```

次の例は、複数索引表列の索引の索引設定値を入手します。

```
db2tx GET INDEX SETTINGS db2tx.mytable  
      HANDLE commenthandle
```

複数の索引表として使用可能になっている表の場合は、このコマンドは表の中  
のすべての使用可能なテキスト列の索引設定値を表示します。

次の内容は共通索引表に対する `GET INDEX SETTINGS` によって表示される  
出力の例です。複数索引表の出力は、索引ごとに同じ情報を示します。構文を  
使用すると、特定のテキスト列の索引設定を要求できます。

Current index settings:

Index type	(INDEXTYPE)	=	LINGUISTIC
Update index option	(UPDATEINDEX)	=	UPDATE
Update frequency	(UPDATEFREQ)	=	NONE
Node 1			
Index directory	(DIRECTORY)	=	/home/user1/db2tx/indices

索引がいくつかのノードに分割される場合には、ノード情報は索引ディレクトリーごとに表示されます。

## 列のテキスト設定の表示

### 要約

**実行時** 列のテキスト設定に関する情報が必要な場合

### コマンド

GET TEXT INFO

**権限** なし

次の例は、共通索引表の索引のテキスト情報を入手します。

```
db2tx GET TEXT INFO db2tx.mytable
```

次の例は、複数索引表の索引のテキスト情報を入手します。

```
db2tx GET TEXT INFO db2tx.mytable HANDLE commenthandle
```

構文を使用すると、表名とハンドル列名を指定できます。

表名だけをコマンド中で指定した場合は、その表の中の使用可能な各列のテキスト情報が表示されます。ハンドル列名も指定した場合は、その列のテキスト情報だけが表示されます。

次の例は、複数索引表に対するこのコマンドによって表示される内容を示します。

```
Text information for column ABSTRACT1
      with handle column ABSTRACT1HANDLE:
Coded character set ID (CCSID) = 850
Language              (LANGUAGE) = US_ENGLISH
Format                (FORMAT) = TDS
```

```
Text information for column ABSTRACT2
      with handle column ABSTRACT2HANDLE:
Coded character set ID (CCSID) = 850
Language              (LANGUAGE) = US_ENGLISH
Format                (FORMAT) = TDS
```

## DB2 テキスト・エクステンダー・カタログ表示の処理

DB2 テキスト・エクステンダーはデータベースごとに DB2TX.TEXTINDEXES というカタログ表示を作成し、保守します。これは ENABLE DATABASE の実行時に作成されます。カタログ表示には、DB2 テキスト・エクステンダーで使用可能にする表および列に関する情報が含まれています。

表、列、または外部ファイルが使用可能であれば必ず、DB2TX.TEXTINDEXES 内に新規項目が作成されます。項目は、CHANGE INDEX SETTINGS コマンドを使用して索引設定を修正するたびに更新されます。列または表が使用不能になると、項目は削除されます。

カタログ表示中のデータは、通常の SQL 照会機能によって利用できます。しかし、カタログ表示の修正は、通常の SQL データ操作コマンドでは行えません。カタログ表示を明示的に作成または消去することはできません。表7は、カタログ表示の内容を示します。

表7. DB2 テキスト・エクステンダーのカタログ表示

列名	データ・タイプ	ヌル可能	説明
TABLESCHEMA	CHAR(8)	不可	この項目を適用する表のスキーマです。
TABLENAME	VARCHAR(18)	不可	この項目を適用する表の名前です。
COLUMNNAME	VARCHAR(18)	可	この表の中で使用可能になった列の名前です。表が使用可能になっても、すべての列が使用不能であればこの値はヌルになります。
HANDLENAME	VARCHAR(18)	可	ハンドル列の名前。 TABLESCHEMA.TABLENAME 表の中に使用可能な列がない場合は、この値はヌルになります。
INDEXNAME	CHAR(8)	不可	テキスト表またはテキスト列を使用可能にする際に作成されるテキスト索引の名前です。
LOGTABLE	VARCHAR(18)	不可	INDEXNAME 索引のログ表の名前です。表 DB2TX.LOGTABLE には、テキスト索引の次の更新用にスケジュールされたテキスト文書がどれであるかとエラー事象についての情報が含まれます。
INDEXTYPE	VARCHAR(30)	不可	索引タイプ: LINGUISTIC、PRECISE、NGRAM
MINIMUM	INTEGER	可	索引更新の実行前に必要な、索引の更新要求回数の最小数です。51ページの『索引の更新』を参照してください。更新頻度が NONE に設定されている場合は、この値はヌルになります。

## DB2 テキスト・エクステンダー・カタログ表示の処理

表 7. DB2 テキスト・エクステンダーのカタログ表示 (続き)

列名	データ・タイプ	ヌル可能	説明
DAYS	VARCHAR(15)	可	更新をスケジュールする日です。 51ページの『索引の更新』を参照してください。更新頻度が NONE に設定されている場合は、この値はヌルになります。
HOURS	VARCHAR(75)	可	索引更新をスケジュールする時です。 51ページの『索引の更新』を参照してください。更新頻度が NONE に設定されている場合は、この値はヌルになります。
MINUTES	VARCHAR(185)	可	更新をスケジュールする分です。 51ページの『索引の更新』を参照してください。更新頻度が NONE に設定されている場合は、この値はヌルになります。
INDEXDIRECTORY	VARCHAR(254)	不可	テキスト索引を保管する、ファイル・システム中のディレクトリーの名前です。
UPDATEONCREATE	VARCHAR(10)	不可	値 “update” または “noupdate”。 ENABLE TEXT TABLE または ENABLE TEXT COLUMN の UPDATEINDEX オプションで、または最後の CHANGE INDEX SETTINGS で指定された値です。
COMMONINDEX	VARCHAR(4)	不可	TABLESCHEMA.TABLENAME 表が共通索引表である場合は、「可」です。 TABLESCHEMA.TABLENAME 表が複数の索引表である場合は、「不可」です。
CCSID	SMALLINT	可	テキスト列を使用可能にするコマンドで指定した、 TEXTCOLUMN テキスト列の CCSID です。 TEXTCOLUMN がヌルである場合は、この値はヌルになります。
LANGUAGE	VARCHAR(30)	可	TEXTCOLUMN テキスト列を処理する際に使用するディレクトリーの名前です。 TEXTCOLUMN がヌルである場合は、この値はヌルになります。
FORMAT	VARCHAR(30)	可	TEXTCOLUMN テキスト列に指定した書式です。 TEXTCOLUMN がヌルである場合は、この値はヌルになります。

## DB2 テキスト・エクステンダー・カタログ表示の処理

表 7. DB2 テキスト・エクステンダーのカタログ表示 (続き)

列名	データ・タイプ	ヌル可能	説明
FUNCTIONSCHEMA	CHAR(8)	可	ENABLE TEXT COLUMN 中で FUNCTION オプションを使用して指定したアクセス関数のスキーマです。FUNCTION オプションが指定されていない場合は、この値はヌルになります。
FUNCTIONNAME	VARCHAR(18)	可	ENABLE TEXT COLUMN 中で FUNCTION オプションを使用して指定したアクセス関数の名前です。FUNCTION オプションが指定されていない場合は、この値はヌルになります。
PROTOTYPEHANDLE	VARCHAR(60)	可	パフォーマンス関数で使用するハンドル。これには、テキスト列全体に共通な索引名だけが含まれています。
INDEXOPTION	VARCHAR(30)	可	索引作成時のオプション: CASE_ENABLED
INDEXPROPERTY	VARCHAR(30)	可	索引作成時のプロパティ: SECTIONS_ENABLED
NODENUMBER	INTEGER	不可	表区画のノード番号。

### テキスト準備処理の逆処理

DB2 テキスト・エクステンダーで使用するためにテキストを作成すると、特定の管理上の変更が行われます。この節では、このようなプロセスを取り消す際に役立つ関数について説明します。

### テキスト列を使用不可にする

#### 要約

**実行時** テキスト列でのテキスト検索をこれ以上行わない場合

#### コマンド

```
DISABLE TEXT COLUMN
```

**権限** 表に関する ALTER、SELECT、UPDATE

例:

```
db2tx DISABLE TEXT COLUMN db2tx.mytable  
HANDLE commenthandle
```

テキスト列を使用不能にすると、以下のことが起こります。

- 複数索引表の場合、すなわち列に独自のテキスト索引とログ表がある場合は、索引、ログ表、およびログ表トリガーが削除されます。
- 共通索引の場合、すなわち、1つの索引がすべてのテキスト列によって共有される場合は、この列の文書の用語が共通索引から除去されます。これが表の中に残された唯一の使用可能テキスト列である場合は、索引、ログ表、およびログ表トリガーが削除されます。

## テキスト・ファイルを使用不可にする

### 要約

**実行時** 外部テキスト・ファイル内でテキスト検索をこれ以上行わない場合

### コマンド

DISABLE TEXT FILES

**権限** 表に関する ALTER、SELECT、UPDATE

例:

```
db2tx DISABLE TEXT FILES db2tx.mytable
                        HANDLE commenthandle
```

外部テキスト・ファイルを使用不能にすると、以下のことが起こります。

- このハンドル列の索引が削除されます。
- ログ表とトリガーが削除されます。

## テキスト表を使用不可にする

### 要約

**実行時** テキスト表でのテキスト検索をこれ以上行わない場合

### コマンド

DISABLE TEXT TABLE

**権限** 表に関する ALTER、SELECT、UPDATE

## テキスト準備処理の逆処理

例:

```
db2tx DISABLE TEXT TABLE db2tx.mytable
```

テキスト表を使用不能にすると、以下のことが起こります。

- 表のテキスト列に関する共通索引がある場合は、この索引が削除される。あるいは、テキスト列ごとに個別の索引がある場合は、テキスト列に関する索引がすべて削除される。
- どのテキストを索引付けするかを自動的に記録するのに使用される共通ログ表が削除される。あるいは、テキスト列ごとに個別のログ表がある場合は、すべてのログ表が削除される。
- ログ表の保守に使用されるトリガーが削除される。
- ハンドル列の内容が空白設定になる。

## データベースを使用不可にする

### 要約

**実行時** このデータベースでのテキスト検索をこれ以上行わない場合

### コマンド

```
DISABLE DATABASE
```

**権限** データベースに関する SYSADM または DBADM

接続中のデータベースを切断するには、次のように入力します。

```
db2tx DISABLE DATABASE
```

データベースを切断すると、以下のオブジェクトが削除されます。

- データベースを使用可能にした際に作成された DB2 テキスト・エクステンダーのカタログ表示
- DB2 テキスト・エクステンダーの SQL 関数 (UDF) の宣言、およびこのデータベースの DB2 テキスト・エクステンダー特殊タイプ (UDT)
- このデータベースのテキスト表またはテキスト列に関連したすべての索引
- どのテキスト文書を索引付けするかを自動的に記録するのに使用されるログ表とそれらを保守するのに使用されるトリガー

ハンドル列は特殊タイプの 1 つですが削除できないので、特殊タイプの中には削除されないものもあります。



---

## 第8章 検索およびブラウズのための API 関数の使用法

この章では、DB2 テキスト・エクステンダー API の検索関数とブラウズ関数の使用方法について説明します。これらの関数の詳細については、235ページの『第13章 検索およびブラウズのための API 関数』を参照してください。API 関数を使用したプログラムの例は、261ページの『第14章 サンプル API プログラム』に示します。同じ章の中で、サンプル・ブラウズ関数 `DesBrowseDocument` について説明しています。

### ヒント

検索する前には、45ページの『検索のタイプ』をお読みください。使用する索引タイプによっては、検索結果が異なる場合があります。GET INDEX SETTINGS を使用して、検索するテキストと関連したテキスト索引タイプを調べてください。

---

### アプリケーションのセットアップ

DB2 テキスト・エクステンダー API を使用するアプリケーション・プログラムは、DB2 CLI アプリケーションです。その理由は、API 関数の中にはデータベース接続ハンドルを入力用として必要とするものがあるためです。これは、DB2 CLI アプリケーションに対して考慮すべき規則が、DB2 テキスト・エクステンダー API を使用するアプリケーションにも適用されることを意味します。

ご使用のアプリケーションに、DB2 テキスト・エクステンダーのインストール・ディレクトリーの `include` サブディレクトリーにある `des_ext.h` を組み込んでください。

DB2 テキスト・エクステンダー API を用いてアプリケーション・プログラムを使用するには、そのプログラムを API にリンクしてください。

### UNIX アプリケーションのリンク

ライブラリー `libdesc1.a` を、使用中のアプリケーションにリンクする必要があります。このライブラリーは、DB2 テキスト・エクステンダー・インストール・ディレクトリーの `lib` サブディレクトリーの中にあります。

## アプリケーションのセットアップ

### OS/2 または Windows アプリケーションのリンク

DB2 テキスト・エクステンダー API 関数はすべて、ダイナミック・リンク・ライブラリー `desclapi.dll` の中にあります。外部 API 関数呼び出しは、インポート・ライブラリー `desclapi.lib` の中で解決されます。前者のライブラリーは DB2 テキスト・エクステンダー・インストール・ディレクトリーの `dll` サブディレクトリーの中に、後者は `lib` サブディレクトリーの中に入っています。

API ダイナミック・リンク・ライブラリー `desclapi.dll` とアプリケーションをリンクするには、`desclapi.lib` をライブラリー・ファイルとして使用します。

---

### API 関数の概説

以下に示すのは、検索関数およびブラウズ関数です。最初の関数は検索関数で、残りはブラウズ関数です。

DesGetSearchResultTable  
DesGetBrowseInfo  
DesStartBrowseSession  
DesOpenDocument  
DesGetMatches  
DesCloseDocument  
DesEndBrowseSession  
DesFreeBrowseInfo

#### ヒント

API 関数の多くには接続ハンドル (`hdbc`) が必要です。このハンドルは `SQLConnect` 関数を使用して指定しなければなりません。それによって組み込み SQL プログラムから DB2 テキスト・エクステンダーを呼び出せなくなることはありません。DB2 コール・レベル・インターフェースの手引きおよび解説書では、CLI ステートメントと組み込み SQL ステートメントとの組み合わせ方法が説明されています。

以下のことを行うために API 機能を使用できます。

- テキストの検索

このシナリオでは、検索関数 `DesGetSearchResultTable` だけが必要です。この関数は、入力用として検索引き数とハンドル列名をとります。そしてテキストを検索し、事前に作成されていた結果表の中へ文書に関する情報を入れます。

この関数は、126ページの『テキストの検索』で詳しく説明されています。261ページの『第14章 サンプル API プログラム』も参照してください。

#### • テキストのブラウズ

示されている順序に従って、以下の機能を使用してください。

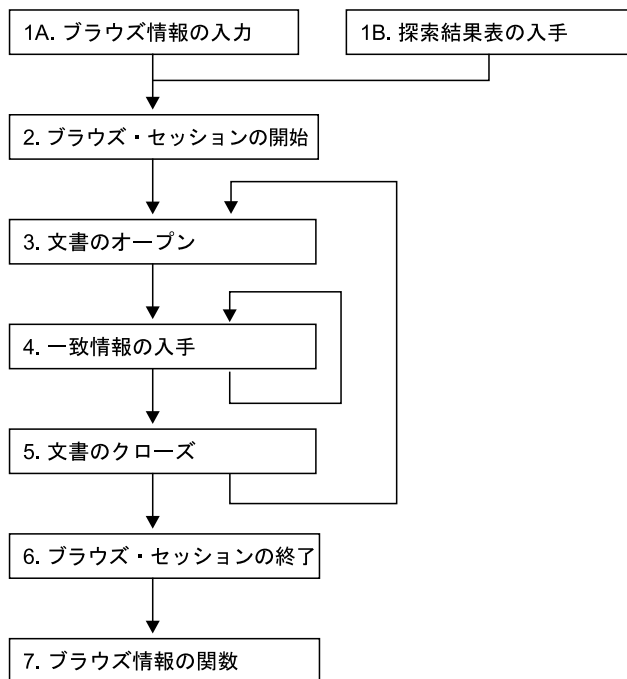


図16. API 関数を使用する順序

これらの関数は、強調表示情報を入手した後、ブラウズ・セッションを開始してテキスト文書を表示し、そこにある用語を強調表示します。

ブラウズ・セッションでは、同じ強調表示情報を使用している文書をさらにオープンして表示することができます。以上の関数は、127ページの『テキストのブラウズ』でさらに詳しく説明されています。

---

### テキストの検索

テキスト検索のための API 関数は、DesGetSearchResultTable 関数の 1 つだけです。

#### 検索結果表の入手 (DesGetSearchResultTable)

DesGetSearchResultTable 関数は、特定のテキスト列中にあるテキスト文書を検索するための検索引き数を受け取り、結果を表中に保管します。結果の表には、検出された個々の文書のハンドルが含まれます。また、指定する検索オプションによっては、ランク情報と一致数を含めることもできます。

RANK および NO\_OF\_MATCHES 関数を使って、この情報を入手することもできます。次に例を示します。

```
INSERT INTO RESULT
  SELECT COMMENTHANDLE,
         RANK(COMMENTHANDLE, "stored procedures"),
         NO_OF_MATCHES(COMMENTHANDLE, "stored procedures")
  FROM DB2TX.SAMPLE
  WHERE CONTAINS(COMMENTHANDLE, "stored procedures") = 1
```

DesGetSearchResultTable は基礎表にしか使用できませんが、テキスト専用照会の場合には DB2 テキスト・エクステンダー関数を使用するよりも速く処理されます。つまり、DB2 テキスト・エクステンダー・サーバーに直接アクセスしてランク情報と一致数を入手し、検出された一致文書の数だけループします。一方、UDF の例では、表中の行ごとに CONTAINS 関数が呼び出されてから、修飾行ごとに RANK および NO\_OF\_MATCHES 関数が呼び出されます。検出文書にはそれぞれ、3 つの別々の検索が行われます。

#### 入力

入力パラメーターは以下のとおりです。

- データベース接続用のハンドル
- 検索される表
- 検索されるテキスト列に関連したハンドル列の名前
- 検索引き数
- 検索オプション
- ブラウズ・オプション (ブラウズ情報を戻す)
- 結果が保管されるテーブルの名前

#### 出力

ブラウズ・オプションを指定すると、この関数はブラウズ情報に対するポインターを戻します。

## テキストのブラウズ

125ページの図16 中のこの関数のグループは、強調表示されている用語を検出します。次に、ブラウズ・セッションを開始して文書をオープンし、ご使用のブラウザーを呼び出すアプリケーション・プログラムにより分析されるデータ・ストリームの書式で一致情報を入力します。

### ブラウズ情報の入手 (DesGetBrowseInfo)

DesGetBrowseInfo 関数は、検索引き数とハンドルを受け取ります。そして DesStartBrowseSession が必要とするブラウズ情報に対するポインターを戻します。ブラウズ情報には、強調表示される全用語のリストが含まれます。

ブラウズ情報を入手する別の方法として、DesGetSearchResultTable 関数に Browse オプションを指定する方法もあります。

#### 入力

入力パラメーターは以下のとおりです。

- データベース接続用のハンドル
- ハンドル
- 検索引き数

#### 出力

この関数は、ブラウズ情報に対するポインターを戻します。

### ブラウズ・セッションの開始 (DesStartBrowseSession)

DesStartBrowseSession 関数はブラウズ・セッションを開始し、テキスト文書のブラウズおよびその一致情報の強調表示に必要な環境を設置します。この関数は、DesGetBrowseInfo または DesGetSearchResultTable のどちらかからブラウズ情報に対するポインターを受け取り、他のブラウズ関数が使用するためのブラウズ・セッション・ハンドルを戻します。

#### 入力

入力パラメーターは以下のとおりです。

- DesGetBrowseInfo または DesGetSearchResultTable からのブラウズ情報に対するポインター
- ユーザー ID
- パスワード

#### 出力

この情報は、ブラウズ・セッション・ハンドルを戻します。

### 文書のオープン (DesOpenDocument)

DesOpenDocument 関数はブラウズ・セッション・ポインター、ハンドル、および検出された用語を強調するのに使用される言語処理タイプを示すオプションである DES\_FAST または DES\_EXTENDED を受け取ります。274ページの『ステージ 2: 拡張突き合わせ』を参照してください。DES\_FAST は、辞書を使用せずに基本テキスト分析を使用して、どの用語を強調表示するかを判別します。DES\_EXTENDED は、拡張一致情報を使用します。

DesOpenDocument は、文書テキストおよび強調表示情報を入手するためのハンドルに対応するテキスト文書を作成し、DesGetMatches の反復呼び出しに使用される文書ハンドルを戻します。

#### 入力

入力パラメーターは以下のとおりです。

DesStartBrowseSession からのブラウズ・セッション・ハンドル  
テキスト・ハンドル  
一致オプション: DES\_FAST または DES\_EXTENDED

#### 出力

この関数は、DesGetMatches および DesCloseDocument が使用する文書ハンドルを戻します。

### 一致情報の入手 (DesGetMatches)

DesGetMatches 関数は、文書ハンドルに記述されたテキスト文書の強調表示情報に対するポインターを戻します。強調表示情報はデータ・ストリームです。これには、テキスト文脈 (最低 1 つの段落) と、その文脈内のテキストを強調表示させるための情報が含まれています。データ・ストリームについては、245ページの『データ・ストリーム構文』で説明します。アプリケーション・プログラムは、データ・ストリームを解析し、ユーザー独自のブラウザーを使用して、それを処理することができます。

DesGetMatches は、データ・ストリームの一部だけを戻します。これは、その出力構造体の一部の長さを示します。

DesGetMatches を連続して呼び出すと、テキスト文書全体の内容を入力できます。テキスト文書の最後に達すると標識が戻されます。

#### 入力

入力パラメーターは以下のとおりです。

ブラウズ・セッション・ハンドル

DesOpenDocument から文書ハンドル

### 出力

この関数は、データ・ストリームの一部およびその長さを含んだ構造体に対するポインターを戻します。

## 文書のクローズ (DesCloseDocument)

DesCloseDocument 関数は、DesOpenDocument がオープンしたテキスト文書をクローズし、文書テキストおよび強調表示情報の戻り時に割り当てられる記憶域を解放します。

### 入力

入力パラメーターは以下のとおりです。

ブラウズ・セッション・ハンドル

DesOpenDocument から文書ハンドル

### 出力

ありません。

## ブラウズ・セッションの終了 (DesEndBrowseSession)

DesEndBrowseSession 関数は、DesStartBrowseSession が開始したブラウズ・セッションを終了し、ブラウズ・セッションに割り当てられた記憶域を解放します。

### 入力

入力パラメーターは以下のとおりです。

ブラウズ・セッション・ハンドル

### 出力

ありません。

## ブラウズ情報の解放 (DesFreeBrowseInfo)

DesFreeBrowseInfo 関数は、DesGetBrowseInfo によってブラウズ情報に割り当てられた記憶域を解放します。

### 入力

入力パラメーターは以下のとおりです。

ブラウズ情報に対するポインター

## テキストのブラウズ

### 出力

ありません。



---

## 第2部 参照情報



## 第9章 クライアント向けのテキスト作成コマンドと管理コマンド

コマンド	用途	ページ
CHANGE INDEX SETTINGS	索引の特性を変更する。	135
CHANGE TEXT CONFIGURATION	テキスト構成設定値を変更する。	137
CONNECT	データベースに接続する。	140
DELETE INDEX EVENTS	索引事象をログ表から削除する。	142
DISABLE DATABASE	データベースを DB2 テキスト・エクステンダーで使用できないようにする。	143
DISABLE TEXT COLUMN	テキスト列を DB2 テキスト・エクステンダーで使用できないようにし、それに関連した索引を削除する。	144
DISABLE TEXT FILES	テキスト列を DB2 テキスト・エクステンダーで使用できないようにし、それらに関連した索引を削除する。	145
DISABLE TEXT TABLE	表を DB2 テキスト・エクステンダーで使用できないようにし、表に関連した索引を削除する。	146
ENABLE DATABASE	DB2 テキスト・エクステンダーで使用するためのデータベースを作成する。	147
ENABLE TEXT COLUMN	DB2 テキスト・エクステンダーで使用するためのテキスト列を作成し、列に対して個別のテキスト索引を作成する。	148
ENABLE TEXT FILES	DB2 テキスト・エクステンダーで使用するためのテキスト・ファイルを作成し、そのファイルに対して個別のテキスト索引を作成する。	157
ENABLE TEXT TABLE	表の共通テキスト索引を作成する。	160
GET ENVIRONMENT	環境変数の現行設定を表示する。	165
GET INDEX SETTINGS	索引の特性を表示する。	166
GET INDEX STATUS	索引のための状況情報を表示する。	168
GET STATUS	データベース、表、列の使用可能状況を表示する。	169
GET TEXT CONFIGURATION	テキスト構成設定値を表示する。	170
GET TEXT INFO	テキスト列のテキスト情報を表示する。	171
QUIT	DB2 テキスト・エクステンダー・コマンド行プロセッサ・モードを終了する。	172
REORGANIZE INDEX	索引を再編成して検索の効率を高める。	173

コマンド	用途	ページ
RESET INDEX STATUS	索引状況を再度使用できるようにリセットする。	174
UPDATE INDEX	テキスト索引を更新する。	175

この章では、クライアント向けのテキスト作成コマンドと管理コマンドの構文について説明します。これらのコマンドの使用方法については、63ページの『第5章 テキストを検索可能にする』および107ページの『第7章 管理』で説明しています。

これらのコマンドを使用する前に、コマンド `db2tx` を入力して DB2 テキスト・エクステンダー・コマンド行プロセッサを開始してください。これにより対話式入力モードに入り、このモードでは後続のコマンドはすべて DB2 テキスト・エクステンダー・コマンドとして解釈されます。通常は、DB2 の開始と同時にコマンド・プロセッサを開始します。

このモードを終了するには `QUIT` と入力します。

DB2 の場合と同じように、`db2tx` を前に付けてオペレーティング・システム・プロンプトから直接コマンドを出すことができ、コマンド行プロセッサ要求をシェル・スクリプト・コマンド・ファイルに入れることができます。

#### ヒント

前に `db2tx` を付けたコマンドを出すと、環境変数 `DB2DBDFT` で指定された省略時データベースに接続されます。次のコマンド順序列は、`MYDATABASE` データベースではなく、省略時のデータベースを使用可能にします。

```
db2tx CONNECT TO MYDATABASE
db2tx ENABLE DATABASE
```



## CHANGE INDEX SETTINGS コマンド

### **HANDLE** handle-column-name

索引更新頻度を変更したいハンドル列の名前です。テキスト列が独自の索引を持っている場合、すなわち **ENABLE TEXT COLUMN** コマンドを使用して索引を作成した場合に、このパラメーターは必須です。

ただし、**ENABLE TEXT TABLE** コマンドを使用して索引を作成した場合、すなわち表がすべてのテキスト列について 1 つのテキスト索引を持っている場合には、このキーワードは無視されます。

### **UPDATEFREQ** update-frequency

更新を実行するとき、およびログ表で待ち行列に入れるテキスト文書の最少数に関する索引の更新頻度です。所定の日時にログ表中に十分な数のテキスト文書がなければ、索引は更新されません。

構文については、51ページの『索引の更新』で説明しています。

**NONE** 以後索引の更新を行いません。これは、これ以上変更を行わないテキスト列のためのものです。

**UPDATEFREQ** キーワードを指定しない場合は、頻度設定はそのまま変更されません。

## CHANGE TEXT CONFIGURATION

このコマンドは、データベースを使用可能にするときに使用されるテキスト構成の省略時設定値を変更します。これらはテキスト構成 設定値です。DB2 テキスト・エクステンダーがインストールされたときのこの表の初期テキスト構成設定値については、21ページの『テキスト構成設定値』で説明しています。

特定のデータベースが使用可能にされた後で、これらの設定値を変更するには、135ページの『CHANGE INDEX SETTINGS』を使用します。

### 許可

以下の権限を持っていないければなりません。

SYSADM 権限

DBADM 権限

### コマンド構文

```

▶▶ CHANGE TEXT CONFIGURATION USING text-information
      |-----|
      | CFG
  
```

```

|-----|
| INDEXTYPE PRECISE
|           |-----|
|           | LINGUISTIC
|           |-----|
|           | NGRAM
|           | TABLESPACE table-space-name
|-----|
  
```

```

|-----|
| UPDATEFREQ NONE
|           |-----|
|           | update-frequency
|           |
|           | DIRECTORY directory
|-----|
  
```

```

|-----|
| UPDATEINDEX UPDATE
|           |-----|
|           | NOUPDATE
|           | COMMITCOUNT count
|-----|
  
```

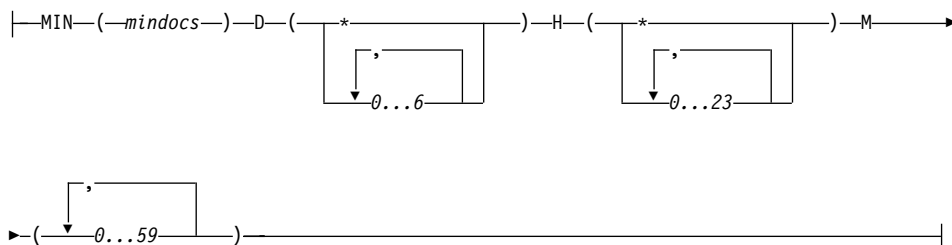
#### text-information:

```

|-----|
| CCSID ccsid
|-----|
| LANGUAGE language
|-----|
| FORMAT format
|-----|
  
```

#### update-frequency:

## CHANGE TEXT CONFIGURATION コマンド



### コマンド・パラメーター

#### INDEXTYPE

省略時索引タイプを変更するには、次のいずれかを選択します。詳細については、45ページの『検索のタイプ』を参照してください。

#### PRECISE

用語はテキスト文書に出てくるとおりに索引付けされ、検索されます。

#### LINGUISTIC

用語は索引付けの前に言語的な処理が行われます。また、検索用語も検索開始前に言語的な処理が行われます。

#### NGRAM

用語は、辞書を使用するのではなく、一連の文字を解析することによって索引付けされます。索引付けしている文書に DBCS 文字が含まれている場合は、この索引タイプが必須ですが、SBCS 文書に対して Ngram 索引を使用することもできます。

#### TABLESPACE *tablename*

既存の表スペースの名前を指定してください。表スペースは、DB2 テキスト・エクステンダーにより作成される索引固有の表 (ログ表など) を保持するために使用されます。ラージ表の場合には、別個の表スペースを使用してください。表スペースを指定しない場合には、表は DB2 省略時表スペースに作成されます。

#### UPDATEFREQ *update-frequency*

更新を実行するとき、およびログ表で待ち行列に入れるテキスト文書の最少数に関する索引の更新頻度です。所定の日時にログ表中に十分な数のテキスト文書がなければ、索引は更新されません。

構文については、51ページの『索引の更新』で説明しています。

**NONE** 以後索引の更新を行いません。これは、これ以上変更を行わないテキスト列のためのものです。



### **DIRECTORY directory**

テキスト索引を保管するディレクトリーです。

### **UPDATEINDEX**

UPDATEFREQ で設定された次の周期的索引付けを待たず、このオプションを使用するコマンドの完了後すぐに、テキスト文書を索引付けするかどうかを決めるキーワードです。これらのコマンドは ENABLE TEXT COLUMN、および ENABLE TEXT FILES です。

### **UPDATE**

このコマンドの完了後すぐに、テキスト文書の索引付けが行われます。

### **NOUPDATE**

このコマンドの UPDATEFREQ またはテキスト構成設定値によって指定された更新頻度設定値に設定された時間に、索引付けが行われます。

### **COMMITCOUNT count**

DB2 中間コミット・ステートメントが出されるまでに行われる挿入または更新の数を示す 500 ~ 1 000 000 の値です。この値を使用することにより、ラージ表、列、または多数の外部ファイル用のログ・スペースが不足するという状態が起きるのを避けることができます。

### **CCSID ccsid**

テキスト文書を索引付けする際に使用するコード化文字セット識別子です。

サポートされる CCSID に関する情報については、39ページの『CCSID』を参照してください。

### **LANGUAGE language**

テキストを書き込む言語です。これにより、テキスト文書を索引付けするときやテキスト文書内を検索するとき使用される辞書が決まります。263ページの『第15章 基本語形索引と一致語句索引の言語的な処理』で、辞書の使用方法を説明しています。

サポートされる言語は、59ページの表5 に列挙されています。

### **FORMAT format**

保管されるテキスト文書のタイプで、WordPerfect または ASCII など。DB2 テキスト・エクステンダーでは、文書を索引付けするときに、この情報が必要です。サポートされる文書形式は、35ページの『サポートされる文書形式』に列挙されています。

## CONNECT コマンド

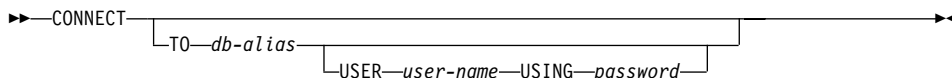
### CONNECT

このコマンドは、DB2 テキスト・エクステンダーをデータベースに接続します。

#### 許可

データベースに対する CONNECT 権限。

#### コマンド構文



#### コマンド・パラメーター

##### TO db-alias

接続するデータベースです。名前は、データベース名ではなく、データベースの別名でなければなりません。

##### USER user-name

ユーザー名を指定しなければ、オペレーティング・システムから検索されます。

##### USING password

パスワードは、ユーザー名を指定した場合に限り必須です。

##### CONNECT no operand

オペランドを指定せず、データベースに接続されていない場合は、CONNECT は DB2DBDFT 環境変数で指定されたデータベースに暗黙に接続します。オペランドを指定せず、データベースに接続している場合は、CONNECT は現行のデータベースに関する情報を表示します。

#### 使用上の注意

データベースに接続される前に DB2TX コマンドを発行すると、DB2 テキスト・エクステンダーは DB2DBDFT 環境変数で指定されたデータベースに接続します。

特定のデータベースに明示的に接続するには、CONNECT TO コマンドを出してください。

1 度に接続できるデータベースは 1 つだけで、これを現行のデータベースといいます。対話モードでは、別の CONNECT TO ステートメントでデータベース

を変更するか、または QUIT コマンドを発行するまで接続は継続されます。コマンド行モードでは、CONNECT コマンドは無効となります。

OS/2 クライアントを使用していて、システムの起動後初めてデータベースに接続する場合には、ユーザー ID とパスワードを入力するようにユーザー・プロフィール管理 (UPM) からプロンプト指示されます。すでにログオンしている場合は、UPM は、このセッションからユーザー ID とパスワードを入手して、リモート・サーバーに接続を試みます。65ページの『DB2 テキスト・エクステンダー・コマンド行プロセッサの開始』を参照してください。

## DELETE INDEX EVENTS コマンド

---

### DELETE INDEX EVENTS

このコマンドは、所定のハンドル列または表のための索引のログ表から、索引付け事象を削除します。

#### 許可

ありません。

#### コマンド構文

```
▶▶—DELETE INDEX EVENTS—table-name—└─HANDLE—handle-column-name—▶▶
```

#### コマンド・パラメーター

##### **table-name**

ログ表からエラー事象が削除される、接続しているデータベース内のテキスト表の名前です。スキーマ名がユーザー ID と異なる場合は、表名に明示のスキーマ名 (修飾子) を含めなければなりません。

##### **HANDLE handle-column-name**

ログ表からメッセージが削除されるハンドル列の名前です。

#### 使用上の注意

ハンドル列名を指定した場合は、その列の索引付け事象だけが削除されます。

ログ表をクリーンアップする場合、索引事象を削除するのは、事象の原因を調べ、可能であればエラーの原因を除去してからにしてください。

## DISABLE DATABASE

このコマンドは、DB2 テキスト・エクステンダーで実行されたデータベースに関する準備作業の内容をリセットし、すべてのテキスト表を DB2 テキスト・エクステンダーで使用できないようにします。

### 許可

以下の権限を持っていないければなりません。

SYSADM 権限

DBADM 権限

### コマンド構文

▶▶—DISABLE DATABASE—▶▶

### コマンド・パラメーター

ありません。

### 使用上の注意

このコマンドは接続しているデータベースをリセットし、以後これを DB2 テキスト・エクステンダーで検索できないようにします。すなわち、データベース内のすべての DB2 テキスト・エクステンダーのテキスト表とテキスト列を使用できないようにします。DB2 テキスト・エクステンダーのテキスト表、テキスト列、および外部ファイルを使用可能にするためにデータベース内で行われたすべての変更がリセットされます。つまり、すべての関連テキスト索引が削除され、データベース内の DB2 テキスト・エクステンダー・カタログ表示 TEXTCOLUMNS が削除され、すべての DB2 テキスト・エクステンダーのトリガーが削除されます。

## DISABLE TEXT COLUMN コマンド

---

### DISABLE TEXT COLUMN

このコマンドは、テキスト列を DB2 テキスト・エクステンダーで使用できないようにします。

#### 許可

表に関して以下のうち少なくとも 1 つを持っていないければなりません。

ALTER 特権

SELECT 特権

UPDATE 特権

#### コマンド構文

►—DISABLE TEXT COLUMN—*table-name*—HANDLE—*handle-column-name*—►

#### コマンド・パラメーター

##### **table-name**

使用不能にしたい列が含まれた、接続しているデータベース内のテキスト表の名前です。スキーマ名がユーザー ID と異なる場合は、表名に明示のスキーマ名 (修飾子) を含めなければなりません。

##### **HANDLE handle-column-name**

DB2 テキスト・エクステンダーで使用できないようにするハンドル列の名前です。

#### 使用上の注意

索引が削除されます。

ハンドル列内の変更 (挿入、更新、および削除) を記録するログ表が削除されます。

ログ表に項目を書き込むトリガーが削除されます。

ハンドル列は変更されません。

## DISABLE TEXT FILES

このコマンドは、DB2 テキスト・エクステンダーに使用される外部ファイルの集合を使用不可にします。

### 許可

表に関して以下のうち少なくとも 1 つを持っていないければなりません。

ALTER 特権

SELECT 特権

UPDATE 特権

### コマンド構文

▶▶—DISABLE TEXT FILES—*table-name*—HANDLE—*handle-column-name*————▶▶

### コマンド・パラメーター

#### **table-name**

使用不可にされる外部テキスト・ファイルのハンドル列に含まれる、接続しているデータベース内のテキスト表の名前です。スキーマ名がユーザー ID と異なる場合は、表名に明示のスキーマ名 (修飾子) を含めなければなりません。

#### **HANDLE handle-column-name**

DB2 テキスト・エクステンダーで使用できないようにするハンドル列の名前です。

### 使用上の注意

索引が削除されます。

ハンドル列内の変更 (挿入、更新、および削除) を記録するログ表が削除されます。ログ表に項目を書き込むトリガーも削除されます。

## DISABLE TEXT TABLE コマンド

---

### DISABLE TEXT TABLE

このコマンドは、表中のすべてのテキスト列を DB2 テキスト・エクステンダーで使用できないようにします。

#### 許可

表に関して以下のうち少なくとも 1 つを持っていないければなりません。

ALTER 特権

SELECT 特権

UPDATE 特権

#### コマンド構文

▶—DISABLE TEXT TABLE—*table-name*—▶

#### コマンド・パラメーター

##### **table-name**

使用不能にしたい列が含まれた、接続しているデータベース内のテキスト表の名前です。スキーマ名がユーザー ID と異なる場合は、表名に明示のスキーマ名 (修飾子) を含めなければなりません。

#### 使用上の注意

このコマンドは、表中のすべてのテキスト列を DB2 テキスト・エクステンダーで使用できないようにします。

この表中のテキスト列を ENABLE TEXT COLUMN によって個別に使用可能にした場合は、それらの列に関連したテキスト索引がすべて削除されます。(テキスト列を個別に使用不能にし、それらの列に関連したテキスト索引を個別に削除するには、DISABLE TEXT COLUMN コマンドを使用します。)

ENABLE TEXT TABLE も使用してこの表中のテキスト列を使用可能にした場合は、すべてのテキスト列について 1 つの共通索引があります。このコマンドは共通索引を削除します。

テキスト列での変更 (挿入、更新、および削除) を記録するために使用されるログ表が削除されます。ログ表に項目を書き込むトリガーが削除されます。



## ENABLE DATABASE

このコマンドは、現行のデータベースにテキスト・データを保管できるようにします。

### 許可

以下の権限を持っていなければなりません。

SYSADM 権限

DBADM 権限

### コマンド構文

▶▶—ENABLE DATABASE—▶▶

### コマンド・パラメーター

ありません。

### 使用上の注意

このコマンドを発行するには、データベースに明示または暗黙に接続されていなければなりません (140ページの『CONNECT』を参照)。

このコマンドは、接続しているデータベースを DB2 テキスト・エクステンダーで使用できるよう準備します。これは、DB2 テキスト・エクステンダーのテキスト表またはテキスト列をデータベース内で使用できるようにするための必須ステップです。

ENABLE DATABASE は、DB2TX.TEXTINDEXES と呼ばれる DB2 テキスト・エクステンダー・カタログ表示を作成します。これについては、118ページの『DB2 テキスト・エクステンダー・カタログ表示の処理』で説明しています。さらに、『パフォーマンス』照会に使用される DB2TX.TEXTCOLUMNS というカタログ表示も作成します。

また、21ページの『テキスト構成設定値』で説明されている、テキスト構成設定値も作成します。

DB2 テキスト・エクステンダー特殊タイプおよび DB2 テキスト・エクステンダー関数の宣言のような、他の作業も実行されます。

## ENABLE TEXT COLUMN コマンド

### ENABLE TEXT COLUMN

このコマンドは、テキスト列を DB2 テキスト・エクステンダーで使用できるようにします。

#### 許可

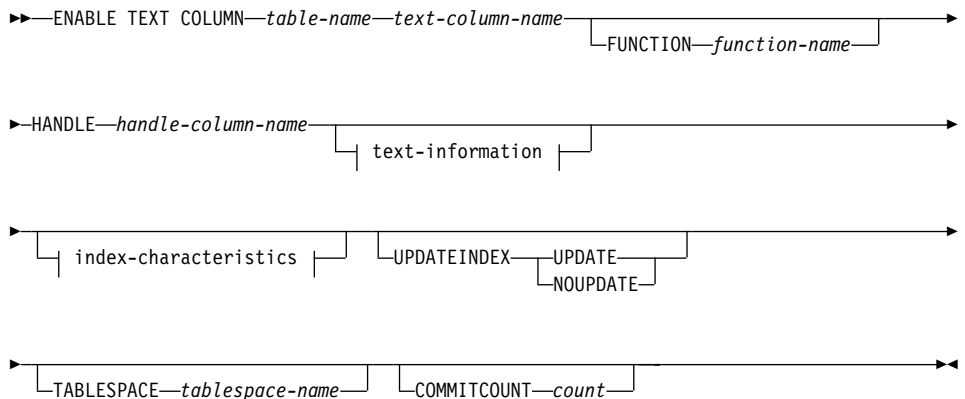
表に関して以下のうち少なくとも 1 つを持っていないければなりません。

ALTER 特権

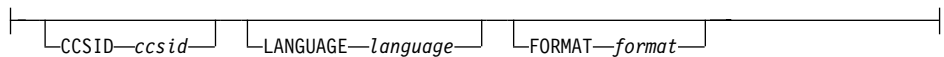
SELECT 特権

UPDATE 特権

#### コマンド構文



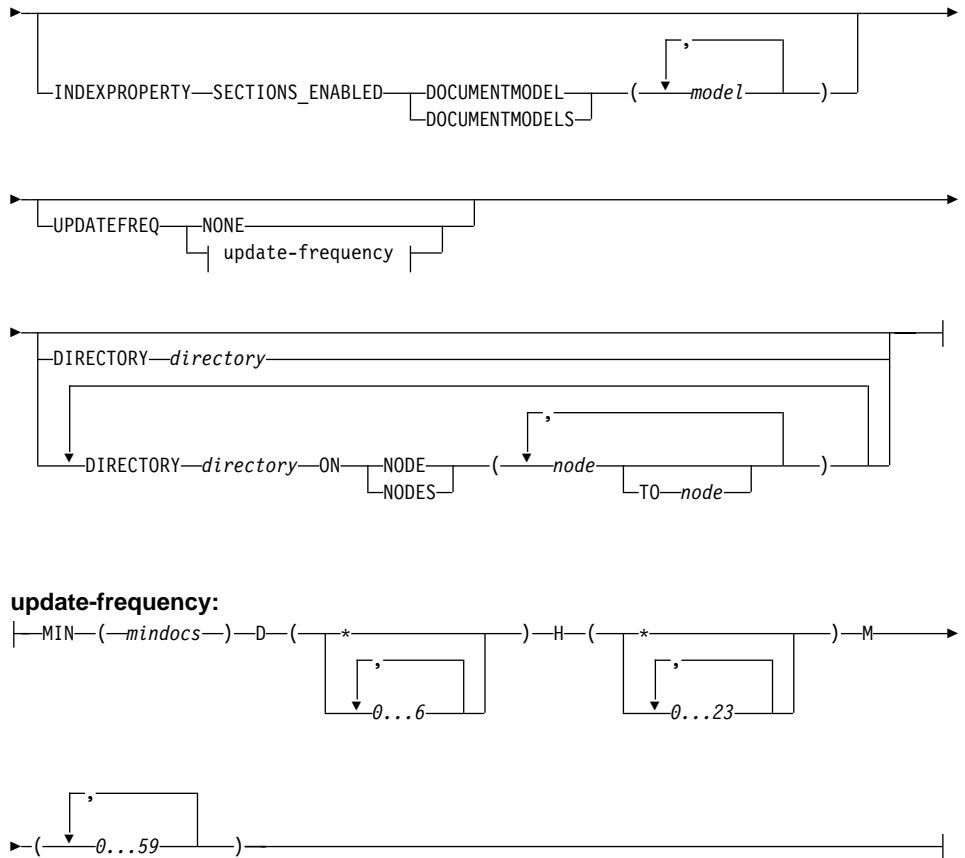
#### text-information:



#### index-characteristics:



## ENABLE TEXT COLUMN コマンド



### コマンド・パラメーター

#### table-name

使用可能列が含まれた、接続しているデータベース内のテキスト表の名前です。スキーマ名がユーザー ID と異なる場合は、表名に明示のスキーマ名 (修飾子) を含めなければなりません。

#### text-column-name

DB2 テキスト・エクステンダーで使用できる列の名前です。この列は次のいずれかのタイプです。CHAR、VARCHAR、LONG VARCHAR、CLOB、DBCLOB、GRAPHIC、VARGRAPHIC、または LONG VARGRAPHIC。文書タイプがこれらのいずれでもない場合は、FUNCTION を使用して、文書タイプを変換してください。

#### FUNCTION function-name

DB2 テキスト・エクステンダーが、以下のいずれのタイプでもない列

## ENABLE TEXT COLUMN コマンド

の中のテキスト文書をアクセスするために使用するユーザー定義関数の名前。CHAR、VARCHAR、LONG VARCHAR、CLOB、DBCLOB、GRAPHIC、VARGRAPHIC、または LONG VARGRAPHIC。詳細については、77ページの『サポートされないデータ・タイプのテキスト列の使用可能化』を参照してください。

### **HANDLE** handle-column-name

DB2 テキスト・エクステンダーの関数で使用するために表に追加するハンドル列の名前です。

### **CCSID** ccsid

テキスト文書を索引付けする際に使用するコード化文字セット識別子です。

Ngram 索引のテキスト列を使用可能にする場合に、CCSID を指定する際には、データベースの CCSID と同じ CCSID を指定しなければならず、検索の際にもこの CCSID (データベースの CCSID) と同じものを使用しなければなりません。

省略時の CCSID を調べるには、以下のコマンドを使用します。

```
db2tx get text cfg
```

インストール時の省略時はデータベースの CCSID です。

キーワードを指定しない場合は、テキスト構成設定値内に指定された CCSID が使用されます。後でテキスト構成設定値に対して行う変更は無視されるため、使用される値はテキスト文書を索引付けするときの値ではなく、列が使用可能になったときの値となります。

サポートされる CCSID に関する情報については、39ページの『CCSID』を参照してください。

### **LANGUAGE** language

テキストを書き込む言語です。これにより、テキスト文書を索引付けするときやテキスト文書内を検索するときを使用される辞書が決まります。263ページの『第15章 基本語形索引と一致語句索引の言語的な処理』で、辞書の使用方法を説明しています。

このキーワードは、1度に列全体に言語を指定します。INSERT ステートメントで INIT\_TEXT\_HANDLE 関数を使うと、別々に挿入されるテキスト文書ごとにこの値を上書きすることができます。

キーワードを指定しない場合は、テキスト構成設定値内に指定された言語が使用されます。後でテキスト構成設定値に対して行う変更は無視さ

れるため、使用される値はテキスト文書を索引付けするときの値ではなく、列が使用可能になったときの値となります。

サポートされる言語は、59ページの表5 に列挙されています。

### FORMAT format

保管されるテキスト文書のタイプで、WordPerfect または ASCII など。DB2 テキスト・エクステンダーでは、文書を索引付けするとき、この情報が必要です。サポートされる文書形式は、35ページの『サポートされる文書形式』に列挙されています。

構造化文書用にサポートされている文書形式は以下のとおりです。

- ASCII\_SECTIONS

ASCII\_SECTIONS 形式の文書には、ネストされたセクションを含めることはできません。(ネスト・セクションについては、53ページの『構造化文書 (セクション・サポート) での作業』を参照してください。) セクションの開始タグは、次の開始タグで終わります。

- HTML

サンプルの文書モデルは HTML 文書に関するものです。この文書には標準的な HTML 定義のサブセットが含まれており、必要に応じて修正することができます。HTML 文書には、ネストされたセクションを含めることはできません。

- XML

XML 文書の処理には、文書タイプ定義 (DTD) の評価が含まれます。文書に割り当てられたモデルは、DTD と比較してチェックされます。文書モデル・ファイル中で定義されているタグが DTD で定義されていない場合、その文書は索引付けされません。認識されている DTD に関する定義がモデル中にない場合、その文書は索引付けされません。XML 文書にはネストされたセクションを含めることができます。

上記の3つの形式については、文書モデル・ファイルに構造情報を指定しなければなりません。53ページの『構造化文書 (セクション・サポート) での作業』を参照してください。書式 TDS と

INDEXPROPERTY SECTION\_ENABLED を指定した場合、文書形式は ASCII\_SECTIONS であるとみなされます。

モデル・ファイル中で定義されていないタグは、索引タイプに従って通常の方法で索引付けされます。

## ENABLE TEXT COLUMN コマンド

このキーワードは、1 度に列全体に書式を指定します。INSERT ステートメントで INIT\_TEXT\_HANDLE 関数を使うと、別々に挿入されるテキスト文書ごとにこの値を上書きすることができます。

このキーワードを指定しない場合は、テキスト構成設定値内に指定された書式が使用されます。後でテキスト構成設定値に対して行う変更は無視されるため、使用される値はテキスト文書を索引付けするときの値ではなく、列が使用可能になったときの値となります。

### INDEXTYPE

作成する索引のタイプです。詳細については、45ページの『検索のタイプ』を参照してください。

### PRECISE

用語はテキスト文書に出てくるとおりに索引付けされ、検索されます。

### LINGUISTIC

用語は索引付けの前に言語的な処理が行われます。また、検索用語も検索開始前に言語的な処理が行われます。

### NGRAM

用語は、辞書を使用するのではなく、一連の文字を解析することによって索引付けされます。索引付けしている文書に DBCS 文字が含まれている場合は、この索引タイプが必須ですが、SBCS 文書に対して Ngram 索引を使用することもできます。

INDEXTYPE キーワードを指定しない場合、テキスト構成設定値の値が使用されます。

Ngram 索引では、XML 形式の文書はサポートされていません。

### INDEXOPTION

索引作成時に使用されるオプションです。

### CASE\_ENABLED

このオプションは、**Ngram 索引に対してのみ**使用可能です。通常、Ngram 索引では、大文字小文字を区別する検索は行われません。CASE\_ENABLED を指定することによって、文書は必ず、大文字小文字を区別する検索が可能であるように索引付けされます。詳細については、48ページの『ファジー検索または DBCS 文書の検索』を参照してください。

**NORMALIZED**

このオプションは、一致語句索引に対してのみ使用可能です。正規化された一致語句索引は、以下の点で一致語句索引と異なります。

- 大文字小文字の区別がない。すべて大文字の場合を除き、すべての語は小文字に変換されます。
- すべて大文字の語は停止ワードのフィルター処理の対象にならない。たとえば、省略語 UK は索引付けされます。
- 英語の検索用語を拡張して、発見的手法アルゴリズムを使用する lemmatization 形式を含めることができる。house を検索すると houses も検索されます。

**INDEXPROPERTY SECTIONS\_ENABLED DOCUMENTMODEL(S) model**

選択した索引タイプのプロパティ。

SECTIONS\_ENABLED を指定すると、選択した索引タイプに、文書構造に関する情報を含めることができます。

DOCUMENTMODEL/DOCUMENTMODELS model は、索引付けされる文書に省略時値として関連付けられる 1 つまたは複数のモデルを指定します。モデル名を指定しなければならないのは、索引プロパティ SECTIONS\_ENABLED を使用している場合です。モデルのリストを指定すると、最初のモデルが索引の省略時のモデルとして使用されます。モデルに対する参照が文書にない場合や、検索の際にモデルを指定しない場合は、省略時のモデルが使用されます。

モデル名に使用できる文字は、a ~ z、A ~ Z、および 0 ~ 9 です。

モデル定義ファイル desmodel.ini 中のモデル定義に対応するモデル名を指定しなければなりません。

索引に関連付けられている 1 つまたは複数のモデルを変更するには、以下のようにします。

1. DISABLE TEXT COLUMN を使用して索引を使用不能にする。
2. ENABLE TEXT COLUMN を使用し、異なる文書モデル名を指定して文書モデル名を索引付けし直す。

**UPDATEFREQ update-frequency**

更新を実行するとき、およびログ表で待ち行列に入れるテキスト文書の最少数に関する索引の更新頻度です。所定の日時にログ表中に十分な数のテキスト文書がなければ、索引は更新されません。

構文については、51ページの『索引の更新』で説明しています。

## ENABLE TEXT COLUMN コマンド

UPDATEFREQ を指定しない場合は、テキスト構成設定値内に指定された省略時の頻度が使用されます。

### ヒント

表がたくさんある場合は、省略時値を使用しないことを考慮してください。表ごとに個別の更新頻度を設定すると、すべての表を同時に索引付けし、その結果システム資源上で不必要にロードが延長されるのを防ぐことができます。

**NONE** 以後索引の更新を行いません。これは、これ以上変更を行わないテキスト列のためのものです。

これらの更新頻度設定は、すでに ENABLE TEXT TABLE によって表全体について設定済みである場合は無視されます。

### DIRECTORY directory

テキスト索引を保管するディレクトリー・パスです。指定されたパスは、“txinsnnn”と連結されます。ここで、*nnn* はノード番号です。

これは、DB2 テキスト・エクステンダー・サーバーが稼働しているシステム上のディレクトリーです。ディレクトリーが既存でない場合は、作成されます。既存の場合は、UNIX プラットフォーム上で、インスタンス所有者にディレクトリーの読み取り / 書き込み許可があるか確認し、DB2 インスタンス作成時に指定したものと違う隔離ユーザー ID が指定されている場合は、その隔離ユーザー ID を検査してください。

DIRECTORY キーワードを指定しない場合は、テキスト構成設定値内の DIRECTORY 設定の値が使用されます。

設定は、ENABLE TEXT TABLE によってすでに設定済みの場合は無視されます。

区分データベースを使用している場合には、すべてのノードまたはノードの範囲に 1 つの共通索引パス名を指定するか、またはノードごとに別の索引ディレクトリー・パス名を指定することができます。

### ON NODE node [TO node]

ディレクトリー・パス名が割り当てられるノードまたはノードの範囲の番号。

### UPDATEINDEX

UPDATEFREQ で設定された次の周期的索引付けを待たず、このコマン



ドの完了後すぐに、このハンドル列に関連したテキスト文書を索引付けするかどうかを指定するキーワードです。

### UPDATE

このコマンドの完了後すぐに、テキスト文書の索引付けが行われます。

### NOUPDATE

このコマンドの UPDATEFREQ またはテキスト構成設定値によって指定された更新頻度設定値に設定された時間に、索引付けが行われます。

このキーワードを指定しない場合は、テキスト構成設定値内の値が採用されます。

### TABLESPACE tablespace-name

ハンドル列に内部的に作成される索引の表スペースの名前。表スペースは事前に作成済みでなければなりません。

### COMMITCOUNT count

挿入または更新の数を示す 500 ~ 1 000 000 の値で、この値を超えると、DB2 により中間コミット・ステートメントが出されなければなりません。この値を使用することにより、ラージ表、列、または多数の外部ファイル用のログ・スペースが不足するという状態が起きるのを避けることができます。

## 使用上の注意

このコマンドは、指定の DB2 表にハンドル列を追加します。各ハンドル列はテキスト列に関連付けられていて、DB2 テキスト・エクステンダーの関数で使用されます。

この表がまだ使用可能になっておらず共通索引を作成できない場合は、このテキスト列に関連した索引が作成されます。

また、ログ表もデータベース内で作成されます。ログ表は、テキスト列に対する変更内容、すなわち挿入、更新、および削除を記録するために使用されます。ログ表を自動的に最新のものに維持するために、テキスト列に関する挿入、更新、および削除トリガーが定義されます。

使用可能にしているテキスト列がマルチノード・ノードグループの一部である表に属している場合には、指定する索引ディレクトリは、すべての物理ノードで使用できなければなりません。テキスト構成で指定される省略時ディレクトリを使用する場合には、ノードグループのすべてのノードでパスを使用

## ENABLE TEXT COLUMN コマンド

きることを確認してください。これが適当でない場合には、それぞれのノードについて特定のパスを ENABLE TEXT COLUMN コマンドに指定することができます。

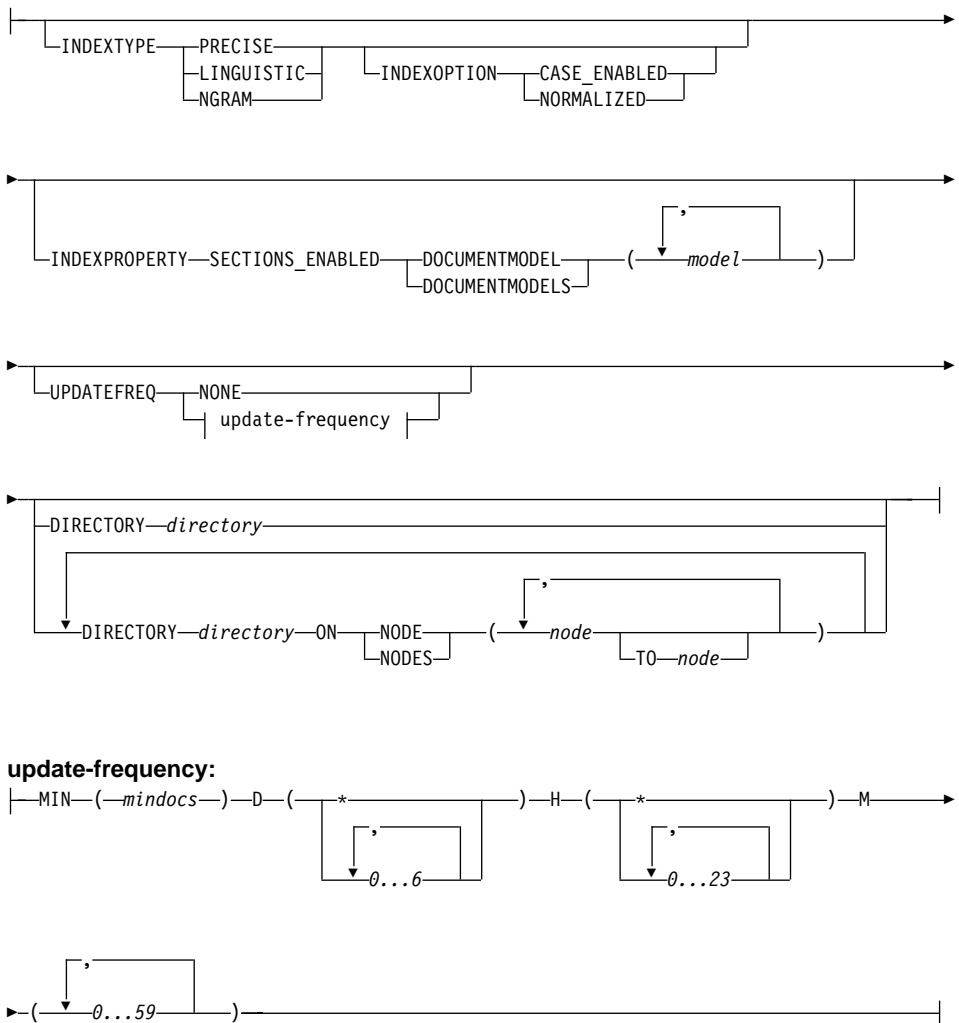
DB2 テキスト・エクステンダーに使用可能にされた表を含むノードグループのノード構成を変更するには、表を索引付けし直さなければなりません。

### ヒント

このステップで、ログ用スペースを使い尽くしてしまった場合は、76ページの『ラージ表でのテキスト列の使用可能化』を参照して、可能な解決方法を見つけてください。



## ENABLE TEXT FILES コマンド



## コマンド・パラメーター

### table-name

索引付けされる外部テキスト・ファイルに関連付けられる、接続しているデータベース内のテキスト表の名前です。スキーマ名がユーザー ID と異なる場合は、表名に明示のスキーマ名 (修飾子) を含めなければなりません。

その他のコマンド・パラメーターの説明については、148ページの『ENABLE TEXT COLUMN』を参照してください。

## 使用上の注意

このコマンドは、指定の DB2 表にハンドル列を追加します。各ハンドル列はテキスト列に関連付けられていて、DB2 テキスト・エクステンダーの関数で使用されます。これらのファイルに関連付けられる索引が作成されます。

テキスト・ファイルを使用可能にしたら、INIT\_TEXT\_HANDLE を使用して、ハンドル列内のハンドルを初期化します。その後で、UPDATE INDEX を使用して、索引を埋めていきます。

ハンドル列名は、ENABLE TEXT FILES で以前に使用されていた場合には、テキスト列のハンドル列を識別するために再使用することはできません。

ログ表は、ファイルに対する変更を記録するために作成されますが、DB2 は、それらのファイルを制御しないため、そのような変更を認知できません。したがって、変更を記録するためには、手操作で、トリガーを活性化させる必要があります。これを行う方法については、108ページの『外部ファイルの索引の更新』を参照してください。

このステップで、ログ用スペースを使い尽くしてしまった場合は、76ページの『ラージ表でのテキスト列の使用可能化』を参照して、可能な解決方法を見つけてください。

## ENABLE TEXT TABLE コマンド

### ENABLE TEXT TABLE

後で使用可能にする、表のすべてのテキスト列で使用するための共通索引を作成します。作成後、表は共通索引表になります。この方法で使用可能にならない表では、後で使用可能にするテキスト列はそれぞれ個別の索引を作成し、表は複数索引表になります。

#### 許可

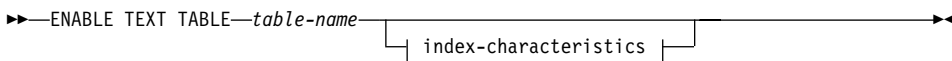
表に関して以下のうち少なくとも 1 つを持っていないければなりません。

ALTER 特権

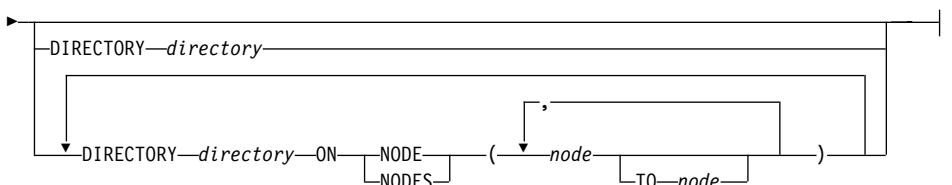
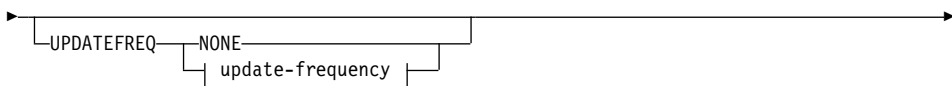
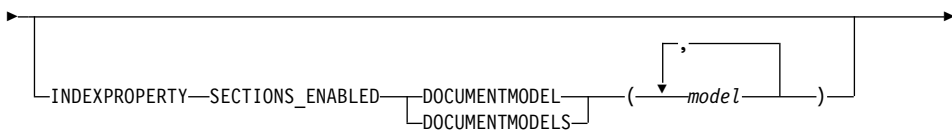
SELECT 特権

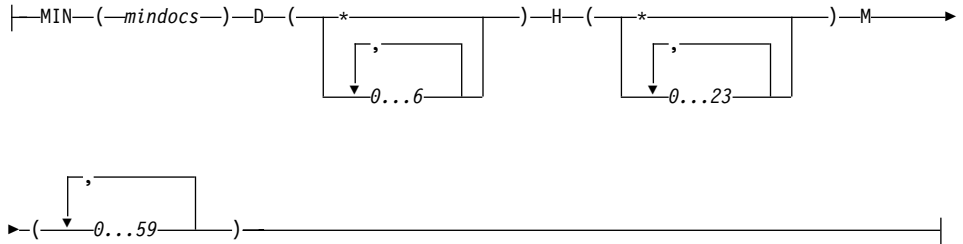
UPDATE 特権

#### コマンド構文



#### index-characteristics:



**update-frequency:****コマンド・パラメーター****table-name**

接続しているデータベース内で使用できるようにしたいテキスト列の名前です。スキーマ名がユーザー ID と異なる場合は、表名に明示のスキーマ名 (修飾子) を含めなければなりません。

**INDEXTYPE**

作成する索引のタイプです。詳細については、45ページの『検索のタイプ』を参照してください。

**PRECISE**

用語はテキスト文書に出てくるとおりに索引付けされ、検索されます。

**LINGUISTIC**

用語は索引付けの前に言語的な処理が行われます。また、検索用語も検索開始前に言語的な処理が行われます。

**NGRAM**

用語は、辞書を使用するのではなく、一連の文字を解析することによって索引付けされます。索引付けしている文書に DBCS 文字が含まれている場合は、この索引タイプが必須ですが、SBCS 文書に対して Ngram 索引を使用することもできます。

INDEXTYPE キーワードを指定しない場合は、テキスト構成が使用されます。

**INDEXOPTION**

索引作成時に使用されるオプションです。

**CASE\_ENABLED**

このオプションは、**Ngram** 索引に対してのみ使用可能です。通常、Ngram 索引では、大文字小文字を区別する検索は行われ

## ENABLE TEXT TABLE コマンド

ません。CASE\_ENABLED を指定することによって、文書は必ず、大文字小文字を区別する検索が可能であるように索引付けされます。詳細については、48ページの『ファジー検索または DBCS 文書の検索』を参照してください。

### INDEXPROPERTY SECTIONS\_ENABLED DOCUMENTMODEL(S) model

選択した索引タイプのプロパティ。

SECTIONS\_ENABLED を指定すると、選択した索引タイプに、文書構造に関する情報を含めることができます。

DOCUMENTMODEL/DOCUMENTMODELS model は、索引付けされる文書に省略時値として関連付けられる 1 つまたは複数のモデルを指定します。モデル名を指定しなければならないのは、索引プロパティ SECTIONS\_ENABLED を使用している場合です。モデルのリストを指定すると、最初のモデルが索引の省略時のモデルとして使用されます。モデルに対する参照が文書にない場合や、検索の際にモデルを指定しない場合は、省略時のモデルが使用されます。

モデル名に使用できる文字は、a ~ z、A ~ Z、および 0 ~ 9 です。

モデル定義ファイル desmodel.ini 中のモデル定義に対応するモデル名を指定しなければなりません。

索引に関連付けられている 1 つまたは複数のモデルを変更するには、以下のようにします。

1. DISABLE TEXT TABLE を使用して索引を使用不能にする。
2. ENABLE TEXT TABLE を使用し、異なる文書モデル名を指定して文書モデル名を索引付けし直す。

### UPDATEFREQ update-frequency

更新を実行するとき、およびログ表で待ち行列に入れるテキスト文書の数に関する索引の更新頻度。所定の日時にログ表中に十分な数のテキスト文書がなければ、索引は更新されません。

構文については、51ページの『索引の更新』で説明しています。

UPDATEFREQ を指定しない場合は、テキスト構成設定値内に指定された省略時の頻度が使用されます。

**NONE** 以後索引の更新を行いません。これは、これ以上変更を行わないテキスト列のためのものです。



**ヒント**

表がたくさんある場合は、省略時値を使用しないことを考慮してください。表ごとに個別の更新頻度を設定すると、すべての表を同時に索引付けし、その結果システム資源上で不必要にロードが延長されるのを防ぐことができます。

**DIRECTORY directory**

テキスト索引を保管するディレクトリー・パスです。指定されたパスは、“txinsnnn”と連結されます。ここで、*nnn* はノード番号です。

これは、DB2 テキスト・エクステンダー・サーバーが稼働しているシステム上のディレクトリーです。ディレクトリーが既存でない場合は、作成されます。既存の場合は、UNIX プラットフォーム上で、インスタンス所有者にディレクトリーの読み取り / 書き込み許可があるか確認し、DB2 インスタンス作成時に指定したものと違う隔離ユーザー ID が指定されている場合は、その隔離ユーザー ID を検査してください。

DIRECTORY キーワードを指定しない場合は、テキスト構成設定値内の DIRECTORY 設定の値が使用されます。

区分データベースを使用している場合には、すべてのノードまたはノードの範囲に 1 つの共通索引パス名を指定することができます。または、それぞれのノードに異なる索引ディレクトリーを指定することもできます。

**ON NODE node [TO node]**

ディレクトリー・パス名が割り当てられるノードまたはノードの範囲の番号。

**使用上の注意**

この表中のすべてのテキスト列に関連した新規のテキスト索引が作成されます。テキスト列ごとに別々の索引を作成するのではなく、表のすべてのテキスト列について 1 つの共通索引を作成したい場合にこれを実行します。

表を使用可能にした場合は、検索したいテキスト列ごとに ENABLE TEXT COLUMN を実行しなければなりません。

## ENABLE TEXT TABLE コマンド

ログ表がデータベース内で作成されます。この表は、後で使用可能にするテキスト列での変更内容、すなわち挿入、更新、および削除を記録するために使用されます。

テキスト列が使用可能になるとトリガーが作成され、このトリガーがテキストに対する変更を監視し、索引付けが必要な文書のログ表に自動的に記録を保持します。

DB2 テキスト・エクステンダーは、ログ表にリストされたテキスト文書を UPDATEFREQ キーワードで指定されたように周期的に索引付けします。

---

## GET ENVIRONMENT

このコマンドは、環境変数の設定を表示します。

### 許可

ありません。

### コマンド構文

▶—GET ENVIRONMENT—▶

### コマンド・パラメーター

ありません。

### 使用上の注意

表示される環境変数は以下のとおりです。

#### **DB2INSTANCE**

DB2 インスタンス名。

#### **DB2DBDFT**

省略時のデータベース名。

#### **DB2TX\_INSTOWNER**

DB2 テキスト・エクステンダー・インスタンス名。 UNIX サーバーに接続している場合に限り必須です。

#### **DB2TX\_INSTOWNERHOMEDIR**

インスタンス所有者のホーム・ディレクトリー。 UNIX サーバーに接続している場合に限り必須です。

## GET INDEX SETTINGS コマンド

### GET INDEX SETTINGS

このコマンドは、索引の設定を表示して以下を示します。

- 索引タイプ
- 索引オプション (任意選択)
- 索引更新オプション
- 索引ディレクトリー
- 更新頻度
- 省略時のモデル

### 許可

ありません。

### コマンド構文

```
▶▶ GET INDEX SETTINGS—table-name ───────────────────────────────────▶▶  
└── HANDLE—handle-column-name ───────────────────────────────────▶▶
```

### コマンド・パラメーター

#### **table-name**

索引設定を表示したい、接続しているデータベース内のテキスト表の名前です。スキーマ名がユーザー ID と異なる場合は、表名に明示のスキーマ名 (修飾子) を含めなければなりません。

#### **HANDLE handle-column-name**

索引設定値が表示されるハンドル列の名前。

### 使用上の注意

複数の索引表が使用可能になっている場合は、このコマンドは指定した表中のすべての使用可能なテキスト列の索引設定を表示します。 *handle-column-name* を指定すると、このコマンドは指定された列の索引設定を表示します。

共通索引表の場合は、共通索引の設定が表示されます。 *handle-column-name* を指定しても無視されます。

索引プロパティ `SECTIONS_ENABLED` によって表または列が使用可能になっている場合、コマンド `GET INDEX SETTINGS` は索引の省略時モデルを表示します。省略時のモデルになるのは、使用可能にする際に指定したモデル名か、またはモデル名リスト中の最初のモデル名です。次に例を示します。

## GET INDEX SETTINGS コマンド

Current index settings:

Index type	(INDEXTYPE)	=	LINGUISTIC
Default model	(DOCUMENTMODEL)	=	mymodel
Update index option	(UPDATEINDEX)	=	UPDATE
Update frequency	(UPDATEFREQ)	=	NONE
Node 1			
Index directory	(DIRECTORY)	=	/home/user1/db2tx/indices

## GET INDEX STATUS コマンド

---

### GET INDEX STATUS

このコマンドは、指定されたハンドル列または表について、以下の索引の状況情報を表示します。

- 検索関数を使用できるかどうか
- 索引更新関数を使用できるかどうか
- 再編成関数を使用できるかどうか
- スケジュールされた文書の数
- 索引付けされた文書の数
- 1 次索引中の索引付けされた文書の数
- 2 次索引中の索引付けされた文書の数
- エラー事象

### 許可

ありません。

### コマンド構文

▶▶ GET INDEX STATUS *table-name* ───────────────────────────────────▶  
└── HANDLE *handle-column-name* ───────────────────────────────────▶

### コマンド・パラメーター

#### **table-name**

状況を表示したいテキスト列が含まれている、接続されているデータベース内のテキスト表の名前。スキーマ名がユーザー ID と異なる場合は、表名に明示のスキーマ名 (修飾子) を含めなければなりません。

#### **HANDLE handle-column-name**

状況を表示したいハンドル列の名前。

### 使用上の注意

複数索引表については、ハンドル列の名前を指定する必要があります。

---

## GET STATUS

このコマンドは、データベース、表、またはテキスト列の使用可能状況に関する情報を表示します。

### 許可

ありません。

### コマンド構文

▶▶—GET STATUS—◀◀

### コマンド・パラメーター

ありません。

### 使用上の注意

このコマンドは、データベースが使用可能であるかどうか、データベース内の使用可能なテキスト表の名前、使用可能なテキスト列とそれらに関連したハンドル列の名前、および外部ファイルのハンドル列の名前を表示します。

## GET TEXT CONFIGURATION コマンド

---

### GET TEXT CONFIGURATION

このコマンドは、接続されているデータベースのテキスト構成の省略時設定値を表示します。

これらの省略時設定値を変更するには、137ページの『CHANGE TEXT CONFIGURATION』を使用してください。

#### 許可

表に関して以下のうち少なくとも 1 つを持っていないければなりません。

SELECT 特権

#### コマンド構文

```
▶ GET TEXT [CONFIGURATION | CFG] ▶▶
```

#### コマンド・パラメーター

ありません。

#### 使用上の注意

テキスト構成情報の例については、113ページの『テキスト構成設定値の表示』を参照してください。



## GET TEXT INFO

このコマンドは、テキスト列のテキスト情報設定を表示します。以下のものが該当します。

- CCSID
- 言語
- 書式

### 許可

ありません。

### コマンド構文

```
▶▶ GET TEXT INFO table-name ───────────────────────────────────▶  
└── HANDLE handle-column-name ───────────────────────────┘
```

### コマンド・パラメーター

#### **table-name**

テキスト情報設定を表示したいテキスト列が含まれた、接続しているデータベース内のテキスト表の名前です。スキーマ名がユーザー ID と異なる場合は、表名に明示のスキーマ名 (修飾子) を含めなければなりません。

#### **HANDLE handle-column-name**

テキスト情報設定値を表示したいハンドル列の名前。

### 使用上の注意

ハンドル列名を指定した場合は、その列のテキスト情報だけが表示されます。

ハンドル列名を指定しない場合は、この表中の使用可能な各列のテキスト情報が表示されます。

## QUIT コマンド

---

### QUIT

このコマンドは、DB2 テキスト・エクステンダー・コマンド行プロセッサを停止し、オペレーティング・システムに制御を戻します。

### 許可

ありません。

### コマンド構文

▶—QUIT—▶▶

### コマンド・パラメーター

ありません。

### 使用上の注意

データベースへの接続は終了します。

## REORGANIZE INDEX

テキスト列が頻繁に更新される場合、索引の検索効率が下がる可能性があります。検索の効率を再び上げるには、索引を再編成する必要があります。DB2 テキスト・エクステンダーは索引を再編成する必要を認識し、バックグラウンドで自動的に再編成を行いますが、REORGANIZE INDEX を使用して手操作で再編成しなければならない場合もあります。索引を再編成する必要があるかどうかを調べるには、コマンド GET INDEX STATUS を使用します。

### 許可

ありません。

### コマンド構文

```

▶▶ REORGANIZE INDEX table-name ───────────────────────────────────▶
    |
    └─ HANDLE handle-column-name ───────────────────────────────────▶
  
```

### コマンド・パラメーター

#### **table-name**

索引を再編成したい、接続しているデータベース内のテキスト表の名前です。スキーマ名がユーザー ID と異なる場合は、表名に明示のスキーマ名 (修飾子) を含めなければなりません。

#### **HANDLE handle-column-name**

索引を再編成したいハンドル列の名前。

### 使用上の注意

複数索引表については、ハンドル列の名前を指定する必要があります。

REORGANIZE INDEX の実行中には索引の検索は行えますが、索引の更新は行えません。

## RESET INDEX STATUS コマンド

### RESET INDEX STATUS

表または列の索引状況が Search not available (検索が実行不能) または Update not available (更新が実行不能) を示している場合は、索引付けの実行中に、索引を処理できなくなるエラーが発生しています。

このコマンドは、索引処理を継続できるように、索引状況をリセットします。索引状況をリセットする前に、索引のログ表に記録される場合があるエラーをチェックします (115ページの『エラー事象の表示』を参照してください)。

#### 許可

ありません。

#### コマンド構文

```
▶▶—RESET INDEX STATUS—table-name—┬───┬───▶▶  
└───┬───┬───  
HANDLE—handle-column-name—┬───┬───▶▶
```

#### コマンド・パラメーター

##### **table-name**

状況をリセットしたいテキスト列が含まれている、接続されているデータベース内のテキスト表の名前。スキーマ名がユーザー ID と異なる場合は、表名に明示のスキーマ名 (修飾子) を含めなければなりません。

##### **HANDLE handle-column-name**

状況をリセットしたいハンドル列の名前。

#### 使用上の注意

複数索引表については、ハンドル列の名前を指定する必要があります。

共通索引表については、その表内の使用可能になった個々の列がリセットされます。

## UPDATE INDEX

このコマンドは、即時に索引付けを開始します。これは索引を最新のものにし、索引に関連したテキスト列（複数の場合もある）の現行の内容を反映します。

外部ファイルの更新を索引に反映させるためには、対応するハンドル列で更新ステートメントを出すことによって、その「変更」項目をログ表に強制的に入れます。この例については、108ページの『外部ファイルの索引の更新』を参照してください。

### 許可

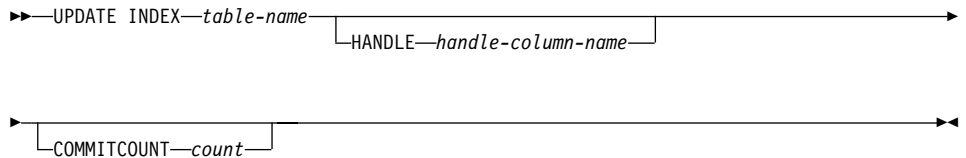
表に関して以下のうち少なくとも 1 つを持っていないければなりません。

ALTER 特権

SELECT 特権

UPDATE 特権

### コマンド構文



### コマンド・パラメーター

#### table-name

索引を更新したいテキスト列が含まれた、接続しているデータベース内のテキスト表の名前です。また、これは共通索引表の名前であってもかまいません。スキーマ名がユーザー ID と異なる場合は、表名に明示のスキーマ名（修飾子）を含めなければなりません。

#### HANDLE handle-column-name

これが共通索引表である場合は、*handle-column-name* は不要であり、無視されます。更新された索引は、個別のテキスト列にではなく表全体に関連したものです。

これが複数索引表である場合、*handle-column-name* は索引を更新したいハンドル列の名前です。

## UPDATE INDEX コマンド

### **COMMITCOUNT count**

挿入または更新の数を示す 500 ~ 1 000 000 の値で、この値を超えると、DB2 により中間コミット・ステートメントが出されなければなりません。この値を使用することにより、ラージ表、列、または多数の外部ファイル用のログ・スペースが不足するという状態が起きるのを避けることができます。

## 第10章 サーバー向け管理コマンド

この章では、サーバー向け管理コマンドの構文を説明します。サーバー管理は、ユーザーが実行できる以下のタスクからなります。DB2 テキスト・エクステンダー・サーバーの開始、停止、およびその状況のチェックとサンプル・データベースおよびサンプル表の作成。これらのコマンドの使用方法については、23ページの『DB2 テキスト・エクステンダー・サーバーの設定および保守』で説明しています。

コマンド	用途	ページ
TXICRT	DB2 テキスト・エクステンダー・インスタンスを作成する。	179
TXIDROP	DB2 テキスト・エクステンダー・インスタンスを除去する。	181
TXILIST (UNIX のみ)	DB2 テキスト・エクステンダー・インスタンスをリストする (UNIX のみ)。	182
TXIPCLEAN (UNIX のみ)	システム障害後にシステム資源をクリーンアップする (UNIX のみ)。	183
TXNADD	DB2 テキスト・エクステンダー・サーバーを追加する。	184
TXNCHECK	DB2 テキスト・エクステンダー・データベースの統合性をチェックする。	185
TXNDROP	DB2 テキスト・エクステンダー・サーバーを除去する。	186
TXSAMPLE	サンプル表を作成して使用可能にする。	187
TXSTART	DB2 テキスト・エクステンダー・サービスを開始する。	188
TXSTATUS	検索サービスの状況を表示する。	189
TXSTOP	DB2 テキスト・エクステンダー・サービスを停止する。	190
TXTHESC	シソーラス定義ファイルをコンパイルする。	191
TXTHESN	Ngram シソーラス定義ファイルをコンパイルする。	193
TXTRACE	トレース情報を生成する。	195

コマンド	用途	ページ
TXVERIFY	サンプル・データベースを作成し、使用可能にする。	200
TXWIZARD	索引付けおよび検索用サンプル・プログラム (AIX、Windows NT、および Windows 2000 のみ)。	201



## TXICRT

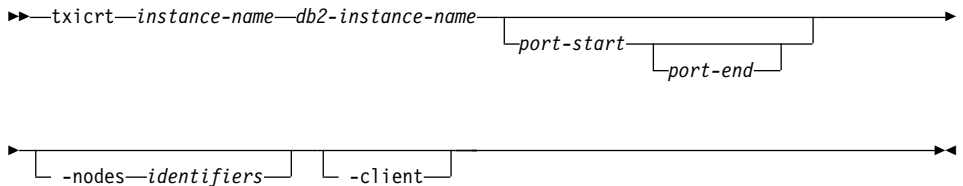
このコマンドは、DB2 テキスト・エクステンダー・インスタンスを作成します。

### 許可

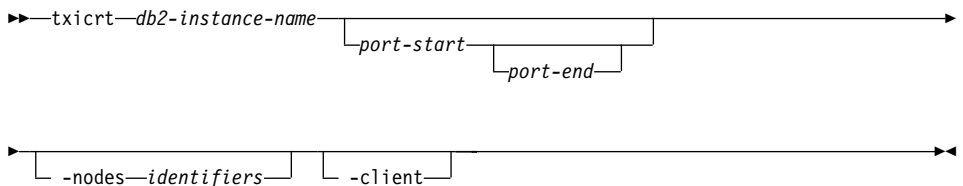
(UNIX のみ) Root

### コマンド構文

UNIX の場合:



Windows NT および Windows 2000 の場合:



### コマンド・パラメーター

*instance-name*

(UNIX のみ) インスタンスの所有者のユーザー ID。

*db2-instance-name*

作成されるインスタンスに関連付けられる DB2 インスタンスの名前。

*port-start/port-end*

複数の物理ノードを使用して作業する場合に使用される TCP/IP ポート範囲。論理ノードしか使用していない場合には、必要ではありません。

ポート範囲は、作業したいノードの番号と等しくなければなりません。1 つのノードを使用して作業している場合には、port-start パラメーターを 1 つだけ指定してください。ノードのサブセットをサポートするには、port-last パラメーターを使用してください。

## TXICRT コマンド

インスタンスが単一区画データベースで作成される場合には、ポート範囲を指定してはなりません。

TCP/IP `etc/services` ファイルにまだリストされていないポート番号のみ使用してください。クライアントのみのインストールではポート番号を指定しないでください。

### **-nodes** *identifiers*

ノード識別子のリスト。例: `-nodes 0 1 2`。

ご使用の表がノードグループ内の特定のノードに属している場合には、このパラメーターを使用してください。

このパラメーターが指定されない場合には、すべてのノードが使用されます。つまり、`db2nodes.cfg` で定義されているすべてのノードが使用されます。

クライアントのみのインストールでは、ノード識別子を指定しないでください。

### **-client**

DB2 テキスト・エクステンダー・クライアントのみのインストールをした場合には、このパラメーターを指定してください。このパラメーターを指定しない場合には、TXICRT は DB2 テキスト・エクステンダーのクライアント / サーバー・インスタンスを構成します。

## 使用上の注意

テキスト表または列の使用可能化は、DB2 テキスト・エクステンダー・インスタンスを作成したノードでのみ実行できます。

DB2 インスタンスは、すでにそのユーザー ID について作成されていなければなりません。

UNIX インストールでは、このインスタンス固有のコマンドは `/usr/lpp/db2tx_07_01/instance` ディレクトリーにあります。

Windows NT または Windows 2000 システムでは、省略時のインスタンスは、インストール時に非区分データベース環境に作成されます。

## TXIDROP

このコマンドは、DB2 テキスト・エクステンダー・インスタンスとその索引すべてを一緒に除去します。

### 許可

(UNIX のみ) Root

### コマンド構文

UNIX の場合:

```
▶▶ txidrop instance-name db2-instance-name ▶▶
```

Windows NT および Windows 2000 の場合:

```
▶▶ txidrop instance-name ▶▶
```

### コマンド・パラメーター

#### **instance-name**

除去するインスタンスの名前。

#### **db2-instance-name**

除去されるインスタンスに関連した DB2 インスタンスの名前。

### 使用上の注意

インスタンスを除去する前に、それが使用可能であるデータベースはいずれも使用不可にしておきます。

UNIX インストールでは、このインスタンス固有のコマンドは /usr/lpp/db2tx\_07\_01/instance ディレクトリーにあります。

## TXILIST コマンド (UNIX のみ)

---

### TXILIST (UNIX のみ)

このコマンドは、現行バージョンの DB2 テキスト・エクステンダーのインスタンスをリストします。

#### 許可

SYSADM、DBADM

#### コマンド構文

▶—txilist—▶▶

#### コマンド・パラメーター

ありません。

#### 使用上の注意

複数のバージョンの DB2 テキスト・エクステンダーがインストールされている場合、このコマンドは現在使用しているバージョンの DB2 テキスト・エクステンダーのインスタンスだけをリストします。したがって、リストされるバージョンは現行ディレクトリーによって決まります。

---

## TXIPCLEAR (UNIX のみ)

このコマンドは、ログオン・ユーザーのシステム資源をクリーンアップします。つまり、`desssdem` プロセスを強制終了し、共用メモリーをクリーンアップし、メッセージ・キューとセマフォアをセグメント化します。

### 許可

`smadmin` グループのメンバー

### コマンド構文

▶▶—txipclear—◀◀

### コマンド・パラメーター

ありません。

### 使用上の注意

このコマンドを使用する際には注意してください。このコマンドは、DB2 テキスト・エクステンダーの資源だけでなく、ログオン・ユーザーの共用資源をすべてクリーンアップします。常にインスタンス所有者としてログオンすることにより、クリーンアップの対象がインスタンスの資源だけになるよう制限してください。

DB2 テキスト・エクステンダーの開始や停止を適切に行えない状態になることもあります。このような場合は、まず DB2 と DB2 テキスト・エクステンダーを停止してから TXIPCLEAR を実行してみてください。

## TXNADD コマンド

---

### TXNADD

このコマンドは、指定されたノードに追加の DB2 テキスト・エクステンダー・サーバーを作成します。TXSTART コマンドは、続けてサーバーを開始します。

#### 許可

(UNIX のみ) SYSADM、DBADM

#### コマンド構文

→ txnadd—node-number —————→  
                                  └port┘

#### コマンド・パラメーター

##### node-number

追加するノードの番号。ノード番号は、db2nodes.cfg ファイルで指定されているように指定してください。

##### port

追加されたノードに使用されるポート番号。物理ノードを追加している場合や、ご使用のノード構成に物理ノードが含まれている場合には、ポート番号を指定しなければなりません。

#### 使用上の注意

TXNCHECK を使用して、ご使用のデータベースと同期をとるにはどの索引を再作成しなければならないかをチェックしてください。

#### ヒント

論理ノードのあるノード構成に物理ノードを追加する場合には、最初にインスタンスを除去してから、すべてのノードのポート番号を指定してインスタンスを作成し直します。

## TXNCHECK

このコマンドは、指定されたデータベースの DB2 テキスト・エクステンダー統合性をチェックします。

### 許可

(UNIX のみ) SYSADM、DBADM

### コマンド構文

```
▶▶—txncheck—┌──────────┴──────────▶▶  
                database-name
```

### コマンド・パラメーター

#### **database-name**

チェックされるデータベースの名前。データベース名が指定されない場合には、DB2 省略時データベースの名前が使用されます。

### 使用上の注意

DB2 テキスト・エクステンダー使用可能表は、それが入っているノードグループが変更されている場合には再索引付けされなければなりません。TXNCHECK を使用して、再配布されたノードグループのためにどの DB2 テキスト・エクステンダー使用可能表が再索引付けされる必要があるかを表示してください。

## TXNDROP コマンド

---

### TXNDROP

このコマンドは、指定されたノードについて DB2 テキスト・エクステンダー・サーバーを除去します。

#### 許可

(UNIX のみ) SYSADM、DBADM

#### コマンド構文

▶—txndrop—*node-number*—▶

#### コマンド・パラメーター

##### **node-number**

除去するノードの名前。ノード番号は、db2nodes.cfg ファイルで指定されているように指定してください。

#### 使用上の注意

TXNCHECK を使用して、ご使用のデータベースと同期をとるにはどの索引を再作成しなければならないかをチェックしてください。



## TXSAMPLE

このコマンドは、サンプル表を作成し、英語文書のサンプルを COMMENT 列にロードし、そのテキスト列を使用可能にします。18ページの『インストール検査のためにサンプル・データベースを準備する』を参照してください。

### 許可

(UNIX のみ) SYSADM、DBADM

### コマンド構文

```
▶▶ txsample database-name user-id password
```

### コマンド・パラメーター

#### データベース-name

すでに DB2 テキスト・エクステンダーによって使用可能にされているデータベースの名前。ここにサンプル表が作成されます。

#### user-id

これは、クライアント・ワークステーションから処理している場合にのみ必要です。

#### password

これは、クライアント・ワークステーションから処理している場合にのみ必要です。

### 使用上の注意

データベースの CCSID が 850 でない場合は、Ngram 索引タイプの場合にサンプルを使用可能にするステップを参照し、データベースの CCSID に合わせて CCSID を変更してください。

#### ヒント

このコマンドは、クライアント・ワークステーションにおいても使用できます。

## TXSTART コマンド

---

### TXSTART

このコマンドは DB2 テキスト・エクステンダー・サービスを開始します。

#### 許可

(UNIX のみ) SYSADM、DBADM

#### コマンド構文

```
▶ txstart [NODENUM nodenum] ▶▶
```

#### コマンド・パラメーター

##### **NODENUM nodenum**

開始するサーバーのノード番号。ノード番号が指定されない場合には、すべてのサーバーが開始されます。

#### 使用上の注意

以下の場合にこのコマンドを実行します。

- SM 管理グループに含まれているユーザー ID でログオンしているとき
- サーバー・システムを停止して再始動する場合

DB2 テキスト・エクステンダー・サービスの開始時に問題が生じる場合は、DB2 を停止してから TXIPCLEAR コマンドを実行してください。183ページの『TXIPCLEAR (UNIX のみ)』を参照してください。

---

## TXSTATUS

このコマンドは、DB2 テキスト・エクステンダーが起動され、実行中であるかどうかを表示します。

### 許可

(UNIX のみ) SYSADM、DBADM

### コマンド構文

```
▶▶ txstatus [NODENUM nodenum] ▶▶
```

### コマンド・パラメーター

#### **NODENUM nodenum**

状況をチェックするサーバーのノード番号。ノード番号が指定されない場合には、すべてのサーバーの状況が表示されます。



## TXTHESC

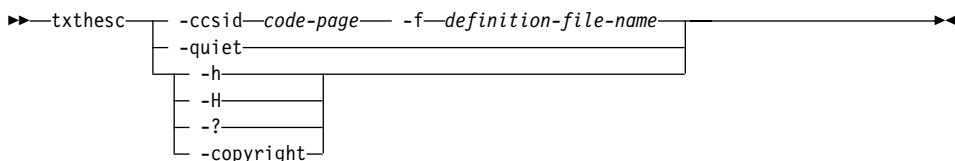
このコマンドは、シソーラス定義ファイルをコンパイルします。このシソーラスは、一致語句索引または基本語形索引の検索だけに使用できます。

Ngram シソーラス定義ファイルのコンパイルについては、193ページの『TXTHESN』を参照してください。

### 許可

(UNIX のみ) ありません。

### コマンド構文



### コマンド・パラメーター

#### **-f** *definition-file-name*

シソーラス定義の入っている SGML ファイルの名前。このファイル名には、ファイルに対する絶対パスか相対パスが含まれていなければなりません。

シソーラス辞書は、定義ファイルと同じディレクトリーに生成されます。この辞書は定義ファイルと同じ名前になりますが、拡張子は th1 ~ th6 になります。

#### ヒント

同じ名前のシソーラス・ファイルがあると上書きされるので、シソーラスごとに別個のディレクトリーを使用してください。

#### **-ccsid** *code-page*

シソーラス定義ファイルが作成される際のコード・ページ。現時点では、コード・ページ 850 だけがサポートされています。

#### **-quiet** 出力情報は表示されません。

## TXTHESC コマンド

### **-copyright**

製品の内部製造番号を戻します。この番号は、問題を報告する際に使用してください。

### **-h、-H、または -?**

ヘルプ情報を表示します。

## 使用上の注意

このコマンドは、標準シソーラス定義ファイルを 2 進数形式にコンパイルするのに使用します。定義ファイルは SGML 形式でなければなりません。

コンパイル済みのシソーラス・ファイルを使用するには、サーバー・インスタンスの辞書ディレクトリーに移動してから、検索時にファイルの場所を指定してください。

OS/2 および Windows システムの辞書ディレクトリーは、次のディレクトリーです。

```
drive:¥dmb¥db2tx¥dict
```

AIX、HP-UX および SUN-Solaris システムの場合は、次のディレクトリーです。

```
DB2TX_INSTOWNER /db2tx/dicts
```

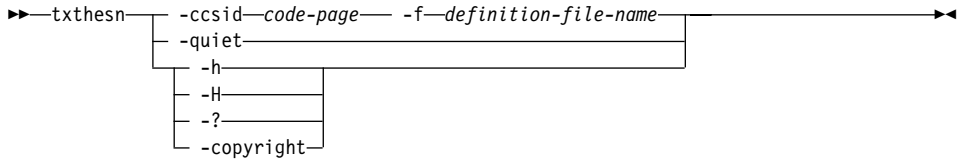
## TXTHESN

このコマンドは、Ngram シソーラス定義ファイルをコンパイルします。このシソーラスは、Ngram 索引の検索だけに使用できます。

### 許可

(UNIX のみ) ありません。

### コマンド構文



### コマンド・パラメーター

#### **-f** *definition-file-name*

シソーラス定義の入っているファイルの名前。このファイル名には、ファイルに対する絶対パスか相対パスが含まれていなければなりません。ファイル名は 8+3 文字に制限されています。拡張子は任意指定です。

シソーラス辞書は、定義ファイルと同じディレクトリーに生成されます。この辞書は定義ファイルと同じ名前になりますが、拡張子は wdf、wdv、grf、grv、MEY、ROS、NEY、SOS、および 1kn (*n* は数字) になります。

#### ヒント

同じ名前のシソーラス・ファイルがあると上書きされるので、シソーラスごとに別個のディレクトリーを使用してください。

#### **-ccsid** *code-page*

シソーラス定義ファイルが作成される際のコード・ページ。サポートされているコード・ページのリストについては、39ページの『CCSID』を参照してください。

**-quiet** 出力情報は表示されません。

## TXTHESN コマンド

### **-copyright**

製品の内部製造番号を戻します。この番号は、問題を報告する際に使用してください。

### **-h、-H、または -?**

ヘルプ情報を表示します。

## 使用上の注意

このコマンドは、シソーラス定義ファイルを 2 進数形式にコンパイルするのに使用します。定義ファイルは、285ページの『Ngram シソーラスの作成』で説明されている形式でなければなりません。

コンパイル済みのシソーラス・ファイルを使用するには、サーバー・インスタンスの辞書ディレクトリーに移動してから、検索時にファイルの場所を指定してください。

OS/2 および Windows システムの辞書ディレクトリーは、次のディレクトリーです。

```
drive:¥dmb¥db2tx¥dict
```

AIX、HP-UX および SUN-Solaris システムの場合は、次のディレクトリーです。

```
DB2TX_INSTOWNER /db2tx/dicts
```



## TXTRACE

このコマンドは、共用メモリー内のトレース・バッファに処理情報を書き込みます。この情報は障害をトレースするために使用されます。これは、トレース・バッファからファイルへ 2 進数形式で書き込まれ、後に、トレースがオフに切り替えられたときに形式設定することもできますし、あるいはトレース機能がオンの間に形式設定して、それをファイルに書き込むこともできます。

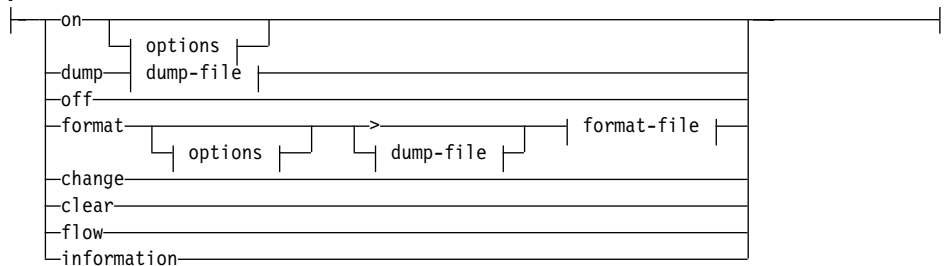
### 許可

(UNIX のみ) SYSADM、DBADM

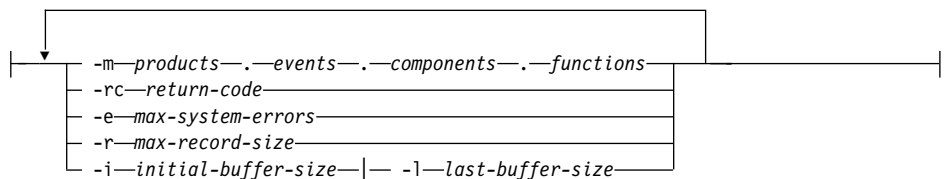
### コマンド構文

```
▶▶ txtrace | parameters |
```

#### parameters:



#### options:

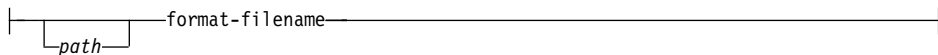


#### dump-file:



#### format-file:

## TXTRACE コマンド



### コマンド・パラメーター

#### 注:

-u オプションは、すべての TXTRACE パラメーターでも使用可能で、そのパラメーターに関する情報を表示します。

**on** トレース機能を開始します。

#### **dump | dmp**

トレース情報を 2 進数形式でファイルに書き込みます。

**off** トレース機能を停止します。

#### **format | fmt**

2 進数形式のトレース情報を形式設定します。トレース機能がオフに切り替えられたら、ダンプ・ファイルの名前と形式設定済みトレース情報を保持するファイルの名前を指定することによって、ダンプ・ファイルを形式設定することができます。トレース機能がオンに切り替えられているままである間に、トレース・バッファーから直接トレース情報を形式設定するためには、`destrc format > filename.tmp` を入力します。

#### **change | chg**

トレース用マスク `maxSysErrors` または `maxRecordSize` を変更します。

#### **clear | clr**

トレースを消去します。

#### **flow | flw**

トレースの制御フローを表示します。

#### **information | info | inf**

トレースに関する情報を入手します。

#### **options**

トレース機能情報がオンであるとき（これは共用メモリーの使用率を減らします）またはトレース情報を形式設定するときに、トレース情報にフィルター処理を適用します。トレース情報が非常に大量でない限り、通常はすべてのトレース情報を書き込んでから、形式設定時にそれをフィルター処理する方が簡単です。

**-m** 「マスク」を追加して、トレースに組み込む事象、構成要素、

および関数を指定します。省略時には、すべてのトレースが行われます。マスクは次のように、それぞれがピリオドで区切られた 4 つの部分からなります 2.2-6.1,3.\*。範囲を示す区切り記号として “-” を使用することができますし、リストを示す区切り記号として “;” を使用することができます。たとえば、2-6 は、ID の範囲が 2 から 6 までの事象だけを組み込みます。構成要素 2 および 6 だけを組み込む場合は、2,6 と指定します。

#### *products*

プロダクト ID。DB2 テキスト・エクステンダーのプロダクト ID は “2” です。TextMiner のプロダクト ID は “3” です。

*events* トレースに組み込む事象タイプのセットです。

<b>0</b>	system_error
<b>1</b>	system_error
<b>2</b>	system_error
<b>3</b>	non-fatal_error
<b>4</b>	non-fatal_error
<b>5</b>	api_errcode
<b>7</b>	fnc_errcode
<b>8</b>	trap_error
<b>10</b>	api_entry
<b>11</b>	api_exit
<b>13</b>	api_retcode
<b>15</b>	api_data
<b>30</b>	fnc_entry
<b>31</b>	fnc_exit
<b>33</b>	fnc_retcode
<b>35</b>	fnc_data

#### *components*

トレースすべき構成要素です。

DB2 テキスト・エクステンダーの構成要素 ID は次のとおりです。

<b>1</b>	COMMAND_LINE_INTERFACE
<b>2</b>	UDF
<b>3</b>	STORED_PROCEDURES
<b>4</b>	ADMINISTRATION
<b>5</b>	INDEX_CONTROL

## TXTRACE コマンド

- 6 LIBRARY\_SERVICES
- 7 DES\_PARSER
- 8 DES\_DEMON
- 9 DES\_API
- 10 SERVICES

TextMiner の構成要素 ID は次のとおりです。

- 1 automachine
- 2 bgproc (バックグラウンド・プロセス)
- 3 cluster
- 4 common
- 5 commsvc (共通サービス)
- 6 communic (通信)
- 7 daemon
- 8 dsclient
- 9 environ (環境)
- 10 glue
- 11 idxcomm (索引作成、共通部分)
- 12 libsrv (ライブラリー・サービス)
- 13 search
- 14 trace
- 15 guru
- 16 indexbld (索引作成、tm のみ)
- 17 indexeng (索引エンジン、tm のみ)
- 18 smsearch
- 19 (検索エンジン、tm のみ)
- 20 tmsearch
- 21 gtrcm (gtr、共通部分)
- 22 gtrsrch (検索、gtr のみ)
- 23 gtridx (索引作成、gtr のみ)

### *functions*

アスタリスク (\*)。トレースする関数のセット。IBM サポート・センターから他の指示がないかぎり、アスタリスク (\*) を使用して、すべての関数をトレースします。

### **-rc** *return-code*

*return-code* をシステム・エラーとして扱います。

### **-e** *max-system-errors*

整数。この数のエラーが発生したら、トレースを停止します。

省略時値は 1 で、これは、最初にシステム・エラーが起きたときに、それ以降に発生するより低い重大度の事象をすべて抑止することを示します。これは、最初に起きる重大エラーのみを重視する場合には受け入れることができますが、最初のシステム・エラー発生後の完全なトレースが見たい場合には、もっと大きい数 (-e 50 など) を指定する必要があります。トレース宛先は共用メモリーです。

**-r** *max-record-size*

整数。この数のレコードがトレース・ファイルに書き込まれた後で、トレースを停止します。省略時値は 16 KB です。

**-i** *initial-buffer-size*

整数。トレースの先頭からこの数のレコードを保持します。-i を指定した場合、省略時値は 16 KB です。UNIX システムでは、推奨バッファ・サイズは 2 MB です。

-i も -l も指定しないと、省略時値は -1 になります。

-i を指定すると、折り返し起きません。すなわち、レコードの量が *max-record-size* の値を超えた場合、すべてのトレース項目を消去しても、それ以上のトレース項目は書き込まれません。新規トレース項目を書き込むためには、バッファ・サイズを増やし、トレース機能をオフにしてから、再度オンにします。

**-l** *last-buffer-size*

整数。トレースの終わりからこの数のレコードを保持します。省略時値は 16 KB です。UNIX システムでは、推奨バッファ・サイズは 2 MB です。

**path** 対応するファイルが保管されるディレクトリ。

**dump-filename**

2 進数形式のトレース情報を含むファイルの名前。

**formatted-filename**

定様式トレース情報を含むファイルの名前。

**例**

26ページの『障害のトレース』を参照してください。

## TXVERIFY コマンド

---

### TXVERIFY

このコマンドは、サンプル・データベースを作成し、それを使用可能にします。

#### 許可

(UNIX のみ) SYSADM、DBADM

#### コマンド構文

```
▶▶ txverify database-name [user-id password] ▶▶
```

#### コマンド・パラメーター

##### **database-name**

サンプル表が作成されるデータベースの名前。

##### **user-id**

これは、クライアント・ワークステーションから処理している場合にのみ必要です。

##### **password**

これは、クライアント・ワークステーションから処理している場合にのみ必要です。

#### 使用上の注意

このコマンドは、クライアント・ワークステーションにおいても使用できません。

## TXWIZARD

このコマンドは、AIX、Windows NT、および Windows 2000 システムだけで使用可能です。このコマンドは、選択した表または列に対して索引を作成する方法、およびこれらの索引に対して検索を実行する方法の例として提供されています。

### コマンド構文

▶—txwizard—▶

### コマンド・パラメーター

ありません。

### 使用上の注意

このサンプル・ウィザードは、DB2 テキスト・エクステンダー検索機能により表の選択された列に対して検索を実行するために必要なステップをガイドします。サンプル・ウィザードで提供される Java ソース・コードを使用すれば、テキストの準備と検索を行う独自のアプリケーションを作成したり、Java 環境で DB2 テキスト・エクステンダーの機能を使用し呼び出す方法について理解したりすることができます。

AIX では、ウィザードの Java ソース・コードは、次のディレクトリーに保管されています。

```
/usr/lpp/db2tx_06_01/samples/wizard
```

Windows NT および Windows 2000 では、次のディレクトリーに保管されています。

```
%DMBMPATH%\samples\wizard
```

## TXWIZARD コマンド



---

## 第11章 検索関数

DB2 テキスト・エクステンダーには、DB2 に保管されたテキスト文書を検索してその結果を処理する、SQL 関数が備わっています。この関数のパラメーターの中には、DB2 テキスト・エクステンダーに備わっている特殊タイプと呼ばれるデータ・タイプのものがあります。

この章では、DB2 テキスト・エクステンダーの SQL 関数と特殊タイプについて説明します。

DB2 テキスト・エクステンダーは、txsample.udf と呼ばれる DB2 コマンド行プロセッサ用入力ファイルを備えています。これには、18ページの『インストール検査のためにサンプル・データベースを準備する』で説明されているサンプル表に対して実行される DB2 テキスト・エクステンダー関数の例が含まれています。このファイルは、テキスト作成関数および検索関数の構文の例を調べるのにご使用ください。

---

### DB2 テキスト・エクステンダー特殊タイプ

特殊タイプ	ソース・データ・タイプ	注釈
DB2TEXTH	VARCHAR(60) FOR BIT DATA	<p>テキスト・ハンドル。テキスト列に保管されるテキスト文書の索引付けに必要な情報を含む可変長ストリングです。ハンドル内の情報には、文書 ID、テキストが索引付けされるサーバーの名前、索引名、およびテキスト文書に関する情報が入っています。</p> <p>ハンドルは、DB2 テキスト・エクステンダーが作成する列の中に保管されて、個々のテキスト列と関連付けられます。</p> <p>関数 HANDLE および INIT_TEXT_HANDLE は、このデータ・タイプを返します。</p>

## DB2 テキスト・エクステンダー特殊タイプ

特殊タイプ	ソース・データ・タイプ	注釈
DB2TEXTFH	VARCHAR(210) FOR BIT DATA	<p>ファイル・ハンドル。 DB2 の制御外に保管されたファイルである外部ファイルの索引付けに必要な情報を含む可変長ストリングです。テキスト・ハンドル内の情報には、文書 ID、テキストが索引付けされるサーバーの名前、索引名、テキスト・ファイルに関する情報、およびファイルの場所に関する情報が入っています。</p> <p>ファイル・ハンドルは、DB2 テキスト・エクステンダーが作成する列の中に保管されて、外部ファイルの個々のグループと関連付けられます。</p> <p>関数 FILE および INIT_TEXT_HANDLE は、このデータ・タイプを返します。</p>
DB2TEXTHLISTP	VARCHAR(16) FOR BIT DATA	<p>ハンドル・リスト・ポインター。 検索で検出されたテキスト文書に関連付けられたハンドルのリストに対するポインターです。</p> <p>関数 HANDLE_LIST は、このデータ・タイプを返します。</p>
DB2TEXTFHLISTP	VARCHAR(16) FOR BIT DATA	<p>ハンドル・リスト・ポインター。 検索で検出された外部ファイルに関連付けられたハンドルのリストに対するポインターです。</p> <p>関数 HANDLE_LIST は、このデータ・タイプを返します。</p>

## DB2 テキスト・エクステンダー関数の要約

検索関数	用途	ページ
CCSID	ハンドルから CCSID を戻す。	206
CONTAINS	特定の文書でテキストの検索を行う。	207
FILE	既存のハンドル内のファイルのパスまたは名前を戻すか、変更する。	208
FORMAT	ハンドル内の文書形式の設定を戻す、または変更する。	209

検索関数	用途	ページ
HANDLE <sup>1</sup>	ハンドル・リストからハンドルを戻す。	210
HANDLE_LIST <sup>1</sup>	ハンドルのリストを検索して戻す。	211
INIT_TEXT_HANDLE	書式または言語設定値などの情報を含む部分的に初期化されたハンドルを戻す。	212
LANGUAGE	ハンドル内の言語設定値を戻す、または変更する。	214
NO_OF_DOCUMENTS <sup>1</sup>	ハンドル・リスト中にリストされている文書数を戻す。	215
NO_OF_MATCHES	検出された一致数を検索して戻す。	216
RANK	検出テキスト文書のランク値を検索して戻す。	217
REFINE	検索引き数および詳細化した検索引き数を受け取り、結合された検索引き数を戻す。	218
SEARCH_RESULT	指定された検索ストリングの検索結果の入った中間表を戻す。	219

DB2 テキスト・エクステンダー関数の使用例は、83ページの『第6章 検索方法』に示しています。

---

1. これらの検索関数は、DB2 テキスト・エクステンダーの以前のリリースの機能です。これらの関数は互換性を保つために引き続きサポートされていますが、これらの機能は SEARCH\_RESULT 検索関数に取って替わられています。

### CCSID

CCSID 関数は、ハンドルから CCSID (データ・タイプ SMALLINT) を戻します。これは、対応するテキスト文書の索引付けに使用される CCSID パラメータです。これについては、39ページの『CCSID』で説明されています。個々のテキスト列に関する CCSID を設定するには、ENABLE TEXT COLUMN コマンドを使用します。

#### 関数の構文

▶▶—CCSID—(—*handle*—)————▶▶

#### 関数のパラメーター

##### **handle**

結果が DB2TEXTFH または DB2TEXTH タイプの値となる式です。通常は、戻される CCSID 設定が入っているハンドル列の名前です。

---

## CONTAINS

CONTAINS 関数は、特定のテキスト文書内のテキストを検索します。その文書の中にテキストがあると、INTEGER 値 1 が戻されます。含まれていない場合は 0 が戻されます。

### 関数の構文

▶▶—CONTAINS—(—*handle*—,—*search-argument*—)—————▶▶

### 関数のパラメーター

#### handle

結果が DB2TEXTFH または DB2TEXTH タイプの値となる式です。通常は、検索したいテキスト文書のハンドルが入っているハンドル列の名前です。

#### search-argument

検索する用語を含んだ LONG VARCHAR タイプのストリングです。221ページの『第12章 検索引き数の構文』を参照してください。

### FILE

FILE 関数は以下のいずれかを行います。

- パスまたはファイル名をハンドル内に戻す。
- ハンドル内のパスまたはファイル名を変更して、そのパスとファイル名を戻す。

戻されるハンドルは、DB2TEXTFH タイプの値です。

### 関数の構文

▶▶FILE—(*handle*)————▶▶

▶▶FILE—(*handle*, *file-name*)————▶▶

### 関数のパラメーター

#### handle

結果が DB2TEXTFH タイプの値となる式です。通常は、戻されるファイル名が入っているハンドル列の名前です。

#### file-name

ハンドルに関連付けられる外部ファイルの新しい絶対パスおよびファイル名を指定する VARCHAR(150) タイプのストリングです。このパスは、たとえば、LAN ドライブまたは NFS 取り付けドライブなどです。ファイル・アクセス許可により、DB2 インスタンス所有者によるファイルのアクセスを許可する必要があります。

## FORMAT

FORMAT 関数は、以下のどちらかを行います。

- ハンドル内に指定された文書形式を戻す。
- 文書のハンドル内の書式指定を変更し、その変更されたハンドルを戻す。

戻される文書形式は、VARCHAR(30) タイプのストリングです。戻されるハンドルは DB2TEXTFH または DB2TEXTH のタイプです。

これは、対応するテキスト文書の索引付けに使用される書式パラメーターです。サポートされる文書形式は、35ページの『サポートされる文書形式』に列挙されています。

### 関数の構文

(1)  
 ►► `FORMAT` (`—handle—`)

注:

- 1 VARCHAR(30) タイプの書式値を戻します。

(1)  
 ►► `FORMAT` (`—handle—, —format—`)

注:

- 1 DB2TEXTFH または DB2TEXTH タイプのハンドルを戻します。

### 関数のパラメーター

#### handle

結果が DB2TEXTFH または DB2TEXTH タイプの値となる式です。通常は、書式設定が戻されるか設定されるハンドル列の名前です。

#### format

データ・タイプ VARCHAR(30) の新規の文書形式の設定です。

*format* を指定すると、ハンドル内に文書形式が設定されます。この場合、書式設定の代わりに、ハンドルが戻されます。

## HANDLE 関数

---

### HANDLE

HANDLE 関数は、ハンドルのリストから順序番号によって選択された、DB2TEXTFH または DB2TEXTH のタイプのハンドルを戻します。

HANDLE 関数は、INSERT ステートメント中でのみ使用してください。

#### ヒント

この関数は、DB2 テキスト・エクステンダーの以前のリリースの機能です。これは互換性を保つために引き続きサポートされていますが、この機能は SEARCH\_RESULT 検索関数に取って替わられています。

### 関数の構文

▶▶HANDLE(⟨handle-list⟩,⟨integer⟩)◀◀

### 関数のパラメーター

#### handle-list

結果が DB2TEXTHLISTP または DB2TEXTFHLISTP タイプの値となる式です。関数 HANDLE\_LIST によって戻されます。

検索で検出された文書のハンドルのリストに対するポインターです。

#### integer

リスト内のどのハンドルが戻されるかを示す INTEGER 値です。



## HANDLE\_LIST

HANDLE\_LIST 関数は、検索引き数を使用して、テキスト文書を検索し、検出された文書のハンドルのリストを指し示す DB2TEXTHLISTP または DB2TEXTFHLISTP のタイプの値を戻します。

**注:** このリストに対するポインターは、この関数を使用する SQL ステートメントの効力範囲内でしかアクセスできません。

検索項目を含む文書が見つからない場合は、ハンドルのリストは空です。リストが空かどうかを確かめるには、NO\_OF\_DOCUMENTS 関数を使用してください。

基礎表の列に保管されているテキストを検索する場合、この関数は各ステートメントに 1 つしか使用できません。ある視点の列に保管されているテキストを検索しているときに、その視点が複数の基礎表からできている (すなわち、結合視点である) 場合は、個々のハンドル列に 1 つのハンドルを HANDLE\_LIST 関数の *handle* 入力引き数として使用します。

### ヒント

この関数は、DB2 テキスト・エクステンダーの以前のリリースの機能です。これは互換性を保つために引き続きサポートされていますが、この機能は SEARCH\_RESULT 検索関数に取って替わられています。

## 関数の構文

►—HANDLE\_LIST—(*—handle—*,*—search-argument—*)—►

## 関数のパラメーター

### handle

結果が DB2TEXTH および DB2TEXTFH タイプの値となる式です。通常は、検索するテキスト文書のハンドルを含んだハンドル列です。また、TEXTCOLUMNS カタログ表示から抽出されたプロトタイプ・ハンドルも指定できます。

### search-argument

検索する用語を含んだ LONG VARCHAR タイプの文字列です。221ページの『第12章 検索引き数の構文』を参照してください。

### INIT\_TEXT\_HANDLE

INIT\_TEXT\_HANDLE 関数は、テキストの言語または書式の事前設定値が入っている、部分的に初期化されたテキスト・ハンドルを返します。この関数は、ハンドル列の中に挿入することができます。これは、書式や言語が、テキスト構成設定内に指定されているものと異なるテキストを含んでいる行を追加するときに有効です。

戻されるハンドルは DB2TEXTH タイプの値です。

DB2 表でなく、外部ファイルに保管されているテキスト内を検索しようとする場合は、INIT\_TEXT\_HANDLE 関数を使用して、テキストの CCSID、書式、言語、およびファイルの位置に関する事前設定値が入っている、完全に初期化されたハンドルを戻すことができます。

戻されるハンドルは、DB2TEXTFH タイプの値です。

INIT\_TEXT\_HANDLE 関数は、ハンドル値を挿入または更新するために使用します。

### 関数の構文

▶▶ INIT\_TEXT\_HANDLE—(—format—, —language—)————▶▶

▶▶ INIT\_TEXT\_HANDLE—(—CCSID—, —format—, —language—, —file-name—)————▶▶

### 関数のパラメーター

#### format

新規の文書形式設定を指定する、タイプ VARCHAR(30) のストリングです。サポートされている書式は、35ページの『サポートされる文書形式』に列挙しています。

#### language

新規の文書言語設定を指定する、タイプ VARCHAR(30) のストリングです。サポートされる言語は、59ページの表5に列挙されています。

#### file-name

ハンドルに関連付けられる外部ファイルの新しい絶対パスおよびファイル名を指定する VARCHAR(150) タイプのストリングです。UNIX のファイルにアクセスできるようにするには、DB2 インスタンスの所有者が、ファイル・アクセス許可に含まれていなければなりません。

OS/2 および Windows ユーザーの場合には、ファイル・アクセス許可にログオン・ユーザー ID を含んでいなければなりません。

### LANGUAGE

LANGUAGE 関数は、以下のどちらかを行います。

- ハンドル内に指定されている文書言語を戻す。
- 文書のハンドル内の言語指定を変更し、その変更されたハンドルを戻す。

戻される文書言語は、VARCHAR(30) タイプのストリングです。戻されるハンドルは DB2TEXTFH または DB2TEXTH のタイプです。

これは、対応するテキスト文書の索引付けに使用される言語パラメーターです。サポートされる言語は、59ページの表5 に列挙されています。

### 関数の構文

(1)  
▶▶ LANGUAGE (—handle—) ▶▶

注:

- 1 VARCHAR(30) タイプの言語値を戻します。

(1)  
▶▶ LANGUAGE (—handle—, —language—) ▶▶

注:

- 1 DB2TEXTFH または DB2TEXTH タイプのハンドルを戻します。

### 関数のパラメーター

#### handle

結果が DB2TEXTFH または DB2TEXTH タイプの値となる式です。通常は、戻される言語設定が入っているハンドル列の名前です。

#### language

データ・タイプ VARCHAR(30) の新規の文書言語の設定です。

language を指定すると、ハンドル内にこの文書言語が設定されます。この場合、言語設定の代わりにハンドルが戻されます。

## NO\_OF\_DOCUMENTS

NO\_OF\_DOCUMENTS 関数は、検索により検出されたテキスト文書のリストにある項目数を示す INTEGER 値を返します。戻される値は、ハンドルのリスト内で見つかる項目の数です。

### ヒント

この関数は、DB2 テキスト・エクステンダーの以前のリリースの機能です。これは互換性を保つために引き続きサポートされていますが、この機能は SEARCH\_RESULT 検索関数に取って替わられています。

## 関数の構文

▶▶—NO\_OF\_DOCUMENTS—(—*handle-list*—)————▶▶

## 関数のパラメーター

### handle-list

結果が DB2TEXTHLISTP または DB2TEXTFHLISTP タイプの値となる式です。関数 HANDLE\_LIST によって戻されます。

検索で検出された文書のハンドルのリストに対するポインターです。

HANDLE\_LIST および NO\_OF\_DOCUMENTS 関数は、同じ SQL ステートメント内になければなりません。リストはステートメントの効力範囲内でだけ存在するからです。

## NO\_OF\_MATCHES 関数

---

### NO\_OF\_MATCHES

NO\_OF\_MATCHES は、テキスト文書内を検索し、その結果として文書ごとに一致した数を示す INTEGER 値を戻すことができます。

#### 関数の構文

▶▶NO\_OF\_MATCHES(—*handle*—,—*search-argument*—)◀◀

#### 関数のパラメーター

##### **handle**

結果が DB2TEXTFH または DB2TEXTH タイプの値となる式です。通常は、検索したいテキスト文書のハンドルが入っているハンドル列の名前です。

##### **search-argument**

検索する用語を含んだ LONG VARCHAR タイプのストリングです。221ページの『第12章 検索引き数の構文』を参照してください。

## RANK

RANK は、テキスト文書内を検索し、検出された個々の文書のランク値を戻すことができます。この値は、検索引き数により検出文書が記述される程度を示します。

RANK は、0 ~ 1 の DOUBLE 値を戻します。ランク値は、検出された文書が検索基準を満たす度合いを、他の検出文書と比べて示す絶対値です。この値は、文書内で検出された一致数と、その文書サイズを示します。

### 関数の構文

▶▶—RANK—(—handle—,—search-argument—)————▶▶

### 関数のパラメーター

#### handle

結果が DB2TEXTFH または DB2TEXTH タイプの値となる式です。通常は、検索したいテキスト文書のハンドルが入っているハンドル列の名前です。

#### search-argument

検索する用語を含んだ LONG VARCHAR タイプのストリングです。221ページの『第12章 検索引き数の構文』を参照してください。

### REFINE

REFINE 関数は、2 つの検索引き数を取り、それらをブール演算子 AND で 1 つに結合した LONG VARCHAR タイプの検索引き数を戻します。

#### 関数の構文

▶▶REFINE(—*search-argument*—,—*search-argument*—)▶▶

#### 関数のパラメーター

##### search-argument

検索する用語を含んだ LONG VARCHAR タイプのストリングです。  
221ページの『第12章 検索引き数の構文』を参照してください。

検索引き数に、検索パラメーター IS ABOUT、THESAURUS、または EXPAND を含めることはできません。



## SEARCH\_RESULT

SEARCH\_RESULT 関数は、検索結果を中間表に戻します。この関数は、SQL ステートメントの FROM 文節で使用できます。

戻される表の構造は次のとおりです。

列名	データ・タイプ
HANDLE	DB2TX.DB2TEXTH, DB2TX.DB2TEXTFH
NUMBER_OF_MATCHES	INTEGER
RANK	DOUBLE

中間表の、選択された列の値だけが生成されます。Select count(\*) を使用すると、HANDLE 列だけが生成されます。ランク値の計算では多くのシステム資源が消費されるので、ランク値が必要ない場合は中間表のランク値を選択しないでください。

ラージ表については、この関数の方が CONTAINS または RANK より速く処理します。

### 関数の構文

►►SEARCH\_RESULT(—schema—, —table—, —handle—, —search-argument—)◄◄

### 関数のパラメーター

#### schema

検索しようとしている表のスキーマ名。

**table** 検索しようとしている表の名前。

#### handle

検索される文書を含む列に対応したハンドル列の名前。

#### search-argument

検索する用語を含んだ LONG VARCHAR タイプのストリングです。221ページの『第12章 検索引き数の構文』を参照してください。

### 例

例については、105ページの『検索パフォーマンスの向上』をお読みになるか、84ページの『検索関数の構文例の情報源』で説明しているファイルに格納されている DB2 テキスト・エクステンダー関数のサンプルを参照してください。



---

## 第12章 検索引き数の構文

検索引き数とは、テキスト文書内の用語を検索する場合に指定する条件です。検索引き数は、1 つまたは複数の検索項目と検索パラメーターとから成ります。

検索引き数の例は、92ページの『検索引き数の指定』に記載してありますが、これは `txsample.udf` と呼ばれるファイルに格納されています。これには、18ページの『インストール検査のためにサンプル・データベースを準備する』で説明されているサンプル表に対して実行される DB2 テキスト・エクステンダー関数の例が含まれています。

検索引き数を使用する DB2 テキスト・エクステンダー関数には、以下のものがあります。

- **CONTAINS**。この関数は、検索引き数を使って特定のテキスト文書にあるテキストを検索します。その文書の中にテキストがあると、INTEGER 値 1 が戻されます。含まれていない場合は 0 が戻されます。
- **NO\_OF\_MATCHES**。この関数は、検索引き数を使ってテキスト文書中を検索します。この関数により、1 文書当たりの一致数を示す INTEGER 値が戻されます。
- **RANK**。この関数は、検索引き数を使ってテキスト文書中を検索します。これは検出された文書ごとに、その文書が検索引き数により記述される程度を示す値を戻します。
- **REFINE**。この関数は、2 つの検索引き数を結合してタイプ LONG VARCHAR の検索引き数を返します。これは、元の 2 つの検索引き数をブール演算子 AND で連結したものです。
- **HANDLE\_LIST**。この関数は、検索引き数を使ってテキスト文書中を検索します。検出した文書のハンドル・リストを指し示す DB2TEXTHLISTP または DB2TEXTFHLLISTP のタイプの値を戻します。
- **SEARCH\_RESULT**。この関数は、要求された情報 (ランク、一致数、およびハンドル) が入っている表を戻します。

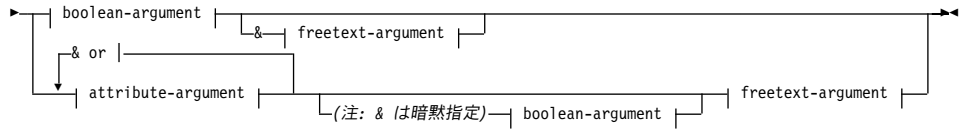
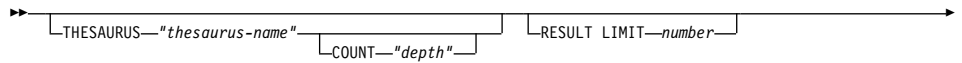
検索引き数を使用する API 関数には、以下のものがあります。

## 検索引き数の構文

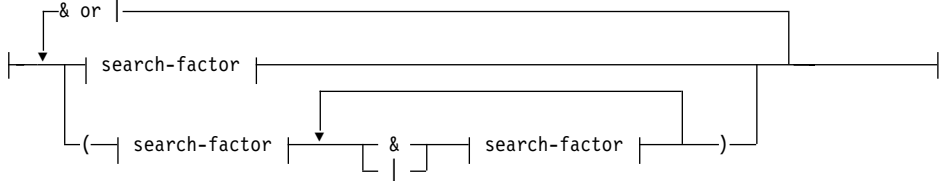
- `DesGetBrowseInfo`。この関数は、検索引き数を使って、ハンドルによって識別されるテキスト中を検索します。これは、`DesStartBrowseSession` が用語を強調表示するために必要となるブラウザ情報に対するポインターを戻します。
- `DesGetSearchResultTable`。この関数は、テキスト列によって識別されるテキスト文書を検索するために検索引き数を使用します。検出されたテキスト項目のハンドル・データは、結果表に書き込まれます。また、ランクや一致数に関するブラウザ情報も結果表に書き込むことができます。

検索引き数

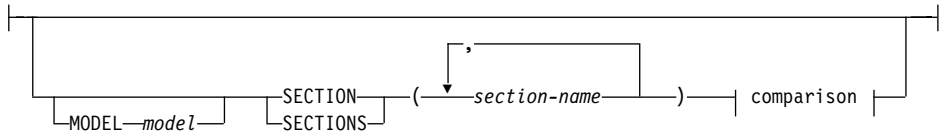
検索引き数構文



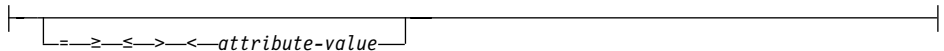
boolean-argument:



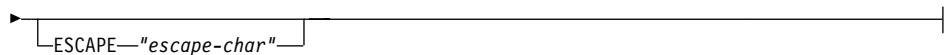
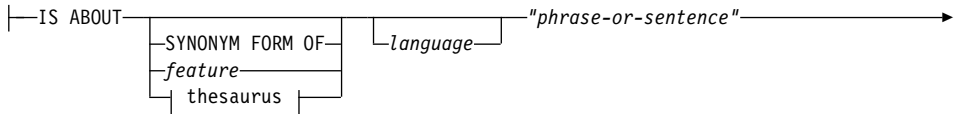
attribute-argument:



Comparison:

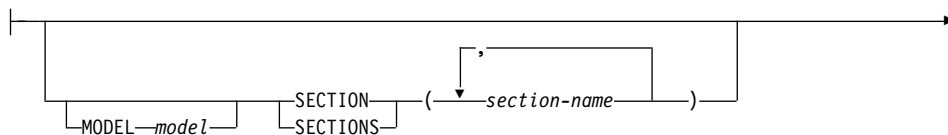


freetext-argument:



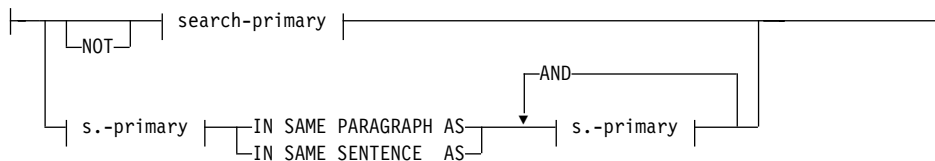
## 検索引き数の構文

### search-factor:

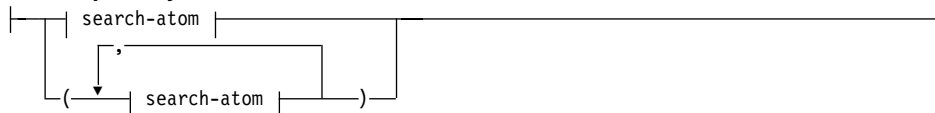


### search-element

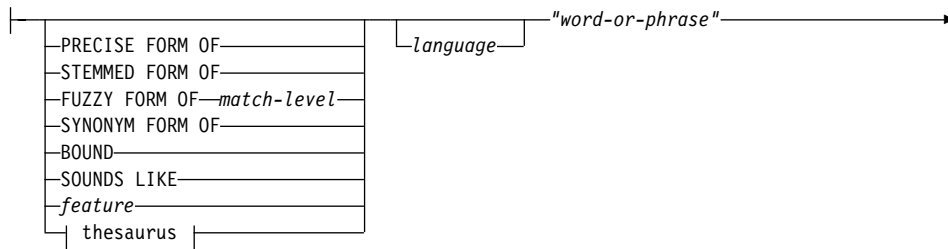
### search-element:



### search-primary:

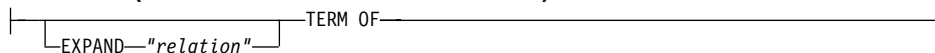


### search-atom:



### ESCAPE—"escape-character"

### thesaurus (THESAURUS が指定されている場合):



## 例

検索引き数の例は、92ページの『検索引き数の指定』に記載されています。

## 検索パラメーター

**IS ABOUT**

ユーザーがフリー・テキスト検索引き数、すなわち、検出したい概念を記述する自然言語の句または文を指定できるオプションです。100ページの『フリー・テキストと混成検索』を参照してください。

**MODEL** *model*

検索用語に使用する文書モデルの名前を指定するキーワードです。文書モデルは文書の構造を記述したものです。記述される文書には識別可能なセクションが含まれており、これらのセクションの内容を個別に検索できます。

モデル名は、53ページの『構造化文書 (セクション・サポート) での作業』で説明されている文書モデル・ファイル中に指定しなければなりません。ワイルドカード文字を使用してモデル名をマスクできます。

モデルを指定しない場合は、索引作成時に指定された省略時のモデルが使用されます。

**SECTION(S)** *section-name*

検索が限定される1つまたは複数のセクションを指定するキーワードです。セクション名は、53ページの『構造化文書 (セクション・サポート) での作業』で説明されている文書モデル・ファイル中のモデル中に指定しなければなりません。ワイルドカード文字 % および \_ を使用してセクション名をマスクできます。

セクションは、次のように他のセクション内にネストできます。

```
play/Act/Title=play/act/title
```

**制約事項:** ネストされたセクションを検索できるのは、XML形式が利用可能な列に保管された文書だけです。Ngram 索引の場合は、1つのセクション名だけを検索でき、XML形式はサポートされていません。

<*attribute-value*

先行する比較演算子と一緒に使用される、その前のセクション・リストにリストされている属性に関する値。属性の値が特定の範囲内に収まるよう要求する照会の場合は、1つの属性条件中に2つの比較演算子を使用できます。

## 検索引き数の構文

同一の条件中に同じ種類の比較演算子を組み合わせて使用する (最初に  $\geq$  を使用し、次に  $>$  を使用するなど) のは無効です。2 つの比較演算子と  $=$  演算子を指定するのも無効です。

- $=$  索引付き文書の属性と、この演算子の後に指定した属性値との等価比較を要求します。
- $\geq$  索引付き文書の属性と、この演算子の後に指定した属性値との「より大か等しい」比較を要求します。
- $>$  索引付き文書の属性と、この演算子の後に指定した属性値との「より大きい」比較を要求します。
- $\leq$  索引付き文書の属性と、この演算子の後に指定した属性値との「より小か等しい」比較を要求します。
- $<$  索引付き文書の属性と、この演算子の後に指定した属性値との「より小」比較を要求します。

セクションは、次のように他のセクション内にネストできます。

```
play/Act/Title=play/act/title
```

**制約事項:** ネストされたセクションを検索できるのは、XML 形式が利用可能な列に保管された文書だけです。Ngram 索引の場合は、1 つのセクション名だけを検索でき、XML 形式はサポートされていません。

### **THESAURUS** *thesaurus-name*

検索用語の拡張に使用するシソーラスの名前を指定するのに使用されるキーワードです。シソーラス名は、シソーラス・コンパイラー TXTHESC または TXTHESN 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。サンプル・ディレクトリーには、省略時シソーラス *desthes* および *desnthes* (*desnthes* は Ngram シソーラス) が保管されています。また、そのファイルのパス名も指定できます。省略時パス名は辞書のパスです。

### **COUNT** *depth*

シソーラスに含まれる用語のレベル数 (深さ) を指定するのに使用されるキーワードです。これらのレベルは、指定された関係の検索用語を拡張するのに使用されます。このキーワードを指定しない場合は、カウント 1 と想定されます。

### **RESULT LIMIT** *number*



結果リストに戻される項目の最大数を指定するのに使用されるキーワードです。 *number* は、1 から 32767 の値です。フリー・テキスト検索が使用される場合には、検索結果リストは、完全な検索結果リストに関してのみランクされます。その他の場合には、限定検索結果がリストの項目からのみランクされます。

### **EXPAND** *relation*

INSTANCE などのように、TERM OF に指定された検索用語と、その検索用語を拡張するのに使用されるシソーラス内の用語との関係を指定するために使用されるキーワードです。関係の名前はシソーラス内で使用されている関係に対応していなければなりません。 275ページの『シソーラスの概念』を参照してください。

Ngram シソーラスの場合は、 285ページの『Ngram シソーラスの作成』で説明されている member-relation 名を使用してください。ユーザー定義のメンバー関係の場合は、 :RELATION *n* (*n* は :RELATED (*number*) で指定されているメンバー関係番号) を使用してください。

### **TERM OF** "*word-or-phrase*"

シソーラスから他の検索用語が追加される検索用語または複数語検索用語です。

### **search-factor**

他のオペランドと結合して検索引き数を形成することができるオペランドです。評価は左から右へと行われます。論理 AND (&) 演算子を使用すると、論理 OR (|) 演算子の場合よりも強いバインドが行われます。例:

```
"passenger" & "vehicle" | "transport" & "public"
```

これは次のように評価されます。

```
("passenger" & "vehicle") | ("transport" & "public")
```

検索を行うには次のようにします。

```
"passenger" & ("vehicle" | "transport") & "public"
```

この場合、上記のように括弧を付けなければなりません。

### **NOT** search-primary

特定の用語を含むテキスト文書を検索から除外するための演算子です。

検索項目 (search factor) 内に NOT を使用する場合は、SYNONYM FORM OF キーワードは使用できません。

### **search-primary IN SAME PARAGRAPH AS search-primary**

同一段落内にある用語の組み合わせを検索するためのキーワードです。

次の検索引き数を指定すると、「air」という用語が同一段落内にある場合にのみ、「traffic」を含むテキスト文書が検出されます。

```
"traffic" IN SAME PARAGRAPH AS "air"
```

IN SAME PARAGRAPH AS キーワードは、検索項目に NOT を使用する場合は指定できません。

### **search-primary IN SAME SENTENCE AS search-primary**

同一文にある用語の組み合わせを検索するためのキーワードです。これは IN SAME PARAGRAPH AS と類似しています。

### **AND search-primary**

同一文または同一段落内で検索を行うために、複数の基本検索語を結合するキーワードです。

次の検索引き数では、同一文で「forest」、「rain」、「erosion」、および「land」が検索されます。

```
"forest" IN SAME SENTENCE AS "rain" AND "erosion" AND "land"
```

### **search-atom**

一連の検索アトム (search atom) の間にコンマを入れると、検索アトムのいずれか 1 つで用語が検出された場合に検索が正常に行われます。どの検索アトムにも語または句が最低 1 つは含まれていなければなりません。

次のステートメントは、1 つまたは複数の検索引き数が検出された場合に真となります。

```
CONTAINS (mytexthandle, '( "text",  
                           "graphic",  
                           "audio",  
                           "video" )' ) = 1
```

### **PRECISE FORM OF、STEMMED FORM OF、FUZZY FORM OF、 SYNONYM FORM OF、BOUND**

229ページの表8 は、種々のタイプの索引に該当するオプションを示しています。たとえば、基本語形索引の場合、PRECISE FORM OF 以外のすべてのオプションが有効です。PRECISE FORM OF を指定すると、これは無視されて省略時値が使用されます。

検索用語処理については、229ページの表9 で詳しく説明しています。

表 8. 基本語形オプション

検索アトム・キーワード	索引タイプ				
	基本語形	一致語句	正規化された一致語句	Ngram	大文字小文字が区別される Ngram
PRECISE FORM OF		X	X		O
STEMMED FORM OF	X			O	O
FUZZY FORM OF				O	O
IS ABOUT	O	O	O		
SYNONYM FORM OF	O	O	O		
EXPAND	O	O	O		
SOUNDS LIKE	O	O	O		
IN SAME SENTENCE AS	O	O	O	O	O
IN SAME PARAGRAPH AS	O	O	O	O	O
BOUND				O	O

X=省略時設定 O=使用可能な機能

表 9. Ngram 索引の検索用語オプション

検索アトム・キーワード	用語検索処理				
	大文字小文字		語幹形への変換	一致	
	区別あり	区別なし		完全	ファジー
PRECISE FORM OF	大文字小文字が区別される場合	X		X	
STEMMED FORM OF		X	X		
FUZZY FORM OF		X			X

X=省略時設定

索引タイプにおいて使用可能でないキーワードを使用すると、それは無視され、代わりに省略時キーワードが使用されるか、メッセージが戻されます。

### PRECISE FORM OF

PRECISE FORM OF に続く語 (または句内のすべての語) について、まずそれを語幹形式に変換するのではなく、タイプされたとおりの語として検索を行います。一致語句索引の場合、この

形式の検索では、大文字小文字が区別されます。つまり、大文字を使うか小文字を使うかが重要な意味を持っています。たとえば、mouse を検索する場合、「Mouse」は検出されません。

これは、一致語句索引での省略時オプションです。正規化された一致語句索引の場合、省略時の検索形式では大文字小文字が区別されません。このキーワードを基本語形索引で指定すると、無視されて **STEMMED FORM OF** が使用されます。

### **STEMMED FORM OF**

**STEMMED FORM OF** に続く語 (または句内のすべての語) を語幹形式に変換してから、検索を行います。この方式の検索では、大文字小文字の区別は行われません。たとえば mouse を検索すると、「Mouse」も検出されます。

語を語幹形に戻す方法は、言語によって異なります。

例: programming computer systems は、米国英語用辞書を使用すると program compute system に置き換えられ、英国英語用辞書を使用すると, programme compute system に置き換えられます。

この検索句では、「programmer computes system」、「program computing systems」、「programming computer system」などを検出することができます。

これは基本語形索引の場合の省略時オプションです。このキーワードを一致語句索引で指定すると、無視されて代わりに **PRECISE FORM OF** が使用されます。

### **FUZZY FORM OF**

検索用語と類似した用語の検索を行う「ファジー」検索を行うためのキーワードです。これは、光学式文字認識 (OCR) プログラムによって作成された文書の検索には特に有効です。そのような文書には、つづりを誤った語が含まれていることが多いからです。たとえば、economy という語は、OCR プログラムでは、econony として認識される可能性があります。

*match-level*: 類似の程度を示す 1 ~ 5 の整数です。5 は 1 より類似度が高いことを示します。

### **SYNONYM FORM OF**

**SYNONYM FORM OF** に続く語または句がその同義語と一緒に検索されるようにするキーワードです。同義語は、*language* で指定された辞書か、そうでなければ省略時の辞書に備えられています。

句の同義語とは、元の句を成す各語をその同義語の 1 つと置き換えることによって得られる同義語の組み合わせをすべて含む別の句です。このとき、語順は元の句と同じです。

このキーワードを一致語句索引で指定すると、無視されて代わりに **PRECISE FORM OF** が使用されます。

このキーワードは、検索項目内に **NOT** を使用した場合、または検索する語または句にマスキング文字が含まれている場合には指定できません。

## BOUND

韓国語 **CCSID** を使用する文書を検索するためのキーワードです。語句の境界を重視する検索を行います。 *language* を指定しても、それは無視され、韓国語と想定されます。

### *language*

索引付けや検索の最中に、テキスト文書の言語的な処理で使用する辞書を指定する変数です。この変数は、基本語形索引ばかりでなく、一致語句索引にも適用されます。これは、一致語句索引が停止ワードを処理するための辞書を使用するからです。

言語的な処理には、同義語処理と語幹処理が含まれます。

サポートされる言語は、59ページの表5 に列挙されています。

**注:** 米国英語で書かれていない文書を検索する場合は、検索引き数 *regardless of the default language* に言語を指定しなければなりません。

### *"word-or-phrase"*

検索する語または句です。1 つの語の中で使用できる文字は、言語によって異なります。また、語を分離文字で分離する必要があるかどうか、言語によって異なります。英語やほとんどの言語の場合、句内のそれぞれの語はブランク文字で分離しなければなりません。

一致語句検索または基本言語検索。 **DB2** テキスト・エクステンダーは、語または句の一致語句書式かその変化形のどちらかを使用して検索することができます。229ページの表8 に記載されているオプションの1 つを指定しない場合は、使用している索引のタイプに応じて省略時の基本語形オプションが使用されます。

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 2 回入力してください。たとえば、テキスト **"wildcard" character** を検索するには、次のように指定します。

## 検索引き数の構文

""wildcard"" character"

マスキング文字。語には以下のマスキング文字を含めることができます。

### **\_ (下線)**

任意の単一文字を表します。

### **% (パーセント)**

任意の個数の任意の文字を表します。語が 1 つの % から成っている場合、それは任意の長さの任意選択語を表します。

% を 1 つ使用して任意選択語を表す場合を除き、語をマスキング文字だけで構成することはできません。

マスキング文字を使用する場合は、SYNONYM FORM OF、*feature*、または THESAURUS を使用することはできません。

### **ESCAPE** *escape-character*

エスケープ文字とは、次に続く文字を検索対象の文字として識別し、マスキング文字として使用する文字ではないと識別するための文字です。

例: エスケープ文字が \$ である場合、\$%、\$\_、および \$\$ は、それぞれ %、\_、および \$ を表します。前に \$ が付かない % および \_ 文字は、マスキング文字を表しています。

### **規則と制約事項の要約:**

#### **ブール演算**

OR の後で NOT を使用することはできません。

#### **FUZZY FORM OF**

先頭の 3 文字は一致しなければなりません。検索アトム内の語にマスキング文字が含まれていると使用できません。NOT と一緒に使用することはできません。Ngram 索引でのみ使用できます。

#### **IN SAME PARAGRAPH AS**

検索係数内で NOT を指定する場合は使用できません。

#### **IN SAME SENTENCE AS**

検索係数内で NOT を指定する場合は使用できません。

#### **基本語形索引**

PRECISE FORM OF は使用できません。省略時値として STEMMED FORM OF を使用します。マスキング文字を使用できます。検索では、大文字小文字が区別されません。

**マスキング文字**

SYNONYM FORM OF と THESAURUS が使用されないようにします。

**Ngram 索引**

マスキング文字を使用できますが、前に英数字以外の文字を付けることはできません。索引で大文字小文字の区別が可能でない場合および PRECISE FORM OF が使用されない場合は、検索では大文字小文字を区別しません。

**NOT** SYNONYM FORM OF、IN SAME PARAGRAPH AS、および IN SAME SENTENCE AS が使用されないようにします。

**PRECISE FORM OF**

基本語形索引では無視されます。

**一致語句索引**

STEMMED FORM OF と SYNONYM FORM OF が使用されないようにします。省略時値として PRECISE FORM OF を使用します。マスキング文字を使用できます。検索では、大文字小文字が区別されます。

**STEMMED FORM OF**

一致語句索引の場合は無視されますが、英語の文書を含む正規化された一致語句索引の場合は使用できます。

**SYNONYM FORM OF**

検索アトム内の語にマスキング文字が含まれていると使用できません。NOT と一緒に使用することはできません。一致語句索引では使用できません。





## 第13章 検索およびブラウズのための API 関数

DB2 テキスト・エクステンダーには、テキスト文書を検索するための C 関数、および検出された文書をブラウズ (表示) するための C 関数があります。これらの関数は、DB2 テキスト・エクステンダー・アプリケーション・プログラミング・インターフェース (API) を構成しています。この章では、API 関数をアルファベット順に説明します。

123ページの『第8章 検索およびブラウズのための API 関数の使用法』ではこの関数とは何かを説明し、さらにこの関数を組み合わせた使用法について述べています。

関数	用途	ページ
DesCloseDocument	DesOpenDocument によって割り当てられた記憶域を解放します。	237
DesEndBrowseSession	ブラウズ・セッションをクローズし、DesStartBrowseSession によって割り当てられた記憶域を解放します。	238
DesFreeBrowseInfo	DesGetBrowseInfo によって割り当てられた記憶域を解放します。	239
DesGetBrowseInfo	検索引き数を使用しているテキストを文書内で検索し、ブラウズ情報を作成します。	240
DesGetMatches	文書ハンドルで記述されたテキスト文書の強調表示情報に対するポインターを戻します。強調表示情報はデータ・ストリームです。これには、テキスト文脈 (最低 1 つの段落) と、その文脈内のテキストを強調表示させるための情報が含まれています。	244
DesGetSearchResultTable	指定されたテキスト列内でテキスト文書を検索するための検索引き数を取得し、その結果をユーザーが作成した表に保管します。ブラウズ情報を戻すことも可能です。	250
DesOpenDocument	ブラウズ・セッション・ポインター、ハンドル、および検出された用語を強調するのに使用される言語処理タイプを示す DES_FAST または DES_EXTENDED を受け取ります。文書テキストおよび強調表示情報を入手するためのハンドルに対応するテキスト文書を作成し、DesGetMatches を反復呼び出しするための文書ハンドルを戻します。	255
DesStartBrowseSession	DesGetBrowseInfo のブラウズ情報を使用してブラウズ・セッションをオープンし、他のブラウズ関数が使用するブラウズ・セッション・ハンドルを戻します。	258

### ヒント

API 関数の多くには接続ハンドル (hdbc) が必要です。このハンドルは SQLConnect 関数を使用して指定しなければなりません、それによって組み込み SQL プログラムから DB2 テキスト・エクステンダーを呼び出せなくなることはありません。DB2 コール・レベル・インターフェースの手引きおよび解説書では、CLI ステートメントと組み込み SQL ステートメントとの組み合わせ方法が説明されています。

## DesCloseDocument

### 目的

DesOpenDocument でオープンされたテキスト文書をクローズし、文書テキストと強調表示情報が戻される際に、割り当てられていた記憶域を解放します。

### 構文

```
DESRETURN
DesCloseDocument
  (DESBROWSESESSION    BrowseSession,
   DESHANDLE            DocumentHandle);
```

### 関数引き数

表 10. DesCloseDocument 引き数

データ・タイプ	引き数	用法	説明
DESBROWSESESSION	<i>BrowseSession</i>	入力	ブラウズ・セッション・ハンドルです。
DESHANDLE	<i>DocumentHandle</i>	入力	オープンされたテキスト文書を識別する DesOpenDocument により戻されたハンドルです。

### 戻りコード

```
RC_SUCCESS
RC_INVALID_PARAMETER
RC_INVALID_SESSION
RC_SE_INCORRECT_HANDLE
RC_SE_IO_PROBLEM
RC_SE_LS_FUNCTION_FAILED
RC_SE_NOT_ENOUGH_MEMORY
RC_SE_REQUEST_IN_PROGRESS
RC_SE_WRITE_TO_DISK_ERROR
```

### 制限

この関数は、DesOpenDocument を呼び出してテキスト文書をオープンした後でしか呼び出すことができません。

### DesEndBrowseSession

#### 目的

DesStartBrowseSession によって開始されたブラウザ・セッションを終了し、そのブラウザ・セッションに割り当てられた記憶域を解放します。

#### 構文

```
DESRETURN  
DesEndBrowseSession  
(DESBROWSESESSION    BrowseSession);
```

#### 関数引き数

表 11. DesEndBrowseSession 引き数

データ・タイプ	引き数	用法	説明
DESBROWSESESSION	<i>BrowseSession</i>	入力	ブラウザ・セッション・ハンドルです。

#### 使用

この関数では、DesGetBrowseInfo によってブラウザ・セッション用に割り当てられた記憶装置は解放されません。この記憶域には、その後もその他のブラウザ・セッションで使用できるブラウザ情報が含まれています。この記憶域を解放するには、DesFreeBrowseInfo を呼び出してください。

#### 戻りコード

```
RC_SUCCESS  
  
RC_INVALID_SESSION  
RC_INVALID_PARAMETER  
RC_SE_UNEXPECTED_ERROR
```

#### 制限

この関数は、DesStartBrowseSession 呼び出しによってブラウザ・セッションを開始した後でしか呼び出すことができません。

## DesFreeBrowseInfo

### 目的

DesGetBrowseInfo によってブラウザ情報に割り当てられた記憶域を解放します。

### 構文

```
DESRETURN  
DesFreeBrowseInfo  
(DESBROWSEINFO      BrowseInfo);
```

### 関数引き数

表 12. *DesFreeBrowseInfo arguments*

データ・タイプ	引き数	用法	説明
DESBROWSEINFO	<i>BrowseInfo</i>	入力	ブラウザ情報です。

### 戻りコード

RC\_SUCCESS

RC\_INVALID\_PARAMETER

### 制限

この情報は、DesGetBrowseInfo 呼び出しによってブラウザ情報用に記憶域を割り当てた後でしか呼び出すことができません。

### DesGetBrowseInfo

#### 目的

ハンドルによって識別されるテキスト内を検索するための検索引き数を受け取ります。これは、`DesStartBrowseSession` が検出用語を強調表示するために必要となるブラウザ情報に対するポインタを戻します。

#### 構文

```

DESRETURN
DesGetBrowseInfo
(SQLHDBC          hdbc,
SQLCHAR          *pHandle,
DESUSHORT        HandleLength,
char             *pSearchArgument,
DESSMALLINT      ArgumentLength,
DESBROWSEINFO    *pBrowseInfo,
DESMESSAGE       *pErrorMessage);
    
```

#### 関数引き数

表 13. `DesGetBrowseInfo` 引き数

データ・タイプ	引き数	用法	説明
SQLHDBC	<i>hdbc</i>	入力	データベース接続ハンドルです。
SQLCHAR *	<i>pHandle</i>	入力	データベースから抽出されるハンドルへのポインタです。
DESUSHORT	<i>HandleLength</i>	入力	<i>pHandle</i> の長さです。DES_NTS はここでは使用できません。
char *	<i>pSearchArgument</i>	入力	検出したい情報を指定するテキスト検索引き数に対するポインタです。
DESSMALLINT	<i>ArgumentLength</i>	入力	<i>pSearchArgument</i> の長さ (ヌル・バイトの終止符を含まない)、または DES_NTS です。
DESBROWSEINFO *	<i>pBrowseInfo</i>	出力	テキスト文書のブラウザのために必要なデータを含むブラウザ情報に対するポインタです。このポインタは <code>DesStartBrowseSession</code> に渡されます。

表 13. DesGetBrowseInfo 引き数 (続き)

データ・タイプ	引き数	用法	説明
DESMESSAGE *	<i>pErrorMessage</i>	出力	処理系定義のメッセージ・テキスト。エラーが発生した場合は、DB2 テキスト・エクステンダーによりエラー・コードとエラー・メッセージが戻されます。アプリケーション・プログラムによってサイズ DES_MAX_MESSAGE_LENGTH のバッファが割り当てられます。 <i>pErrorMessage</i> がヌル・ポインターのとき、エラー・メッセージは返されていません。

## 使用

DesGetBrowseInfo を呼び出す前に、アプリケーション・プログラムでデータベースへの接続を設定しておかなければなりません。

検索引き数に対するポインターの場合、SQLCHAR\* ではなく char\* が使用されます。これは、パラメーター値がデータベースから得られない可能性があるためです。

SQL データ・タイプと C データ・タイプ間のマッピングでは、ハンドルとして SQL 記号名 SQL\_VARBINARY を使用する必要があります。Handle 値の C 表示を指すホスト変数のタイプは SQLCHAR\* です。

DB2 テキスト・エクステンダーにより、ブラウズ情報用に記憶域が割り当てられます。アプリケーション・プログラムでは、DesFreeBrowseInfo を呼び出して記憶域と関連する資源を解放しなければなりません。

Handle 値はビット・データで、'¥0' 文字をいくつか含んでいるため、pHandle の長さを指定する必要があります。

pSearchArgument の検索引き数については、221ページの『第12章 検索引き数の構文』で説明されています。

## 戻りコード

RC\_SUCCESS

RC\_NO\_BROWSE\_INFO

RC\_ALLOCATION\_ERROR

## DesGetBrowseInfo API 関数

RC\_FILE\_IO\_PROBLEM  
RC\_INTERNAL\_ERROR  
RC\_INVALID\_PARAMETER  
RC\_PARSER\_INVALID\_ESCAPE\_CHARACTER  
RC\_PARSER\_INVALID\_USE\_OF\_ESCAPE\_CHAR  
RC\_PARSER\_SYNTAX\_ERROR  
RC\_SE\_COMMUNICATION\_PROBLEM  
RC\_SE\_EMPTY\_INDEX  
RC\_SE\_EMPTY\_QUERY  
RC\_SE\_FUNCTION\_DISABLED  
RC\_SE\_FUNCTION\_IN\_ERROR  
RC\_SE\_INCORRECT\_HANDLE  
RC\_SE\_INDEX\_DELETED  
RC\_SE\_INDEX\_NOT\_ACCESSIBLE  
RC\_SE\_INDEX\_SUSPENDED  
RC\_SE\_INSTALLATION\_PROBLEM  
RC\_SE\_IO\_PROBLEM  
RC\_SE\_MAX\_NUMBER\_OF\_BUSY\_INDEXES  
RC\_SE\_MAX\_OUTPUT\_SIZE\_EXCEEDED  
RC\_SE\_NOT\_ENOUGH\_MEMORY  
RC\_SE\_PROCESSING\_LIMIT\_EXCEEDED  
RC\_SE\_QUERY\_TOO\_COMPLEX  
RC\_SE\_SERVER\_BUSY  
RC\_SE\_SERVER\_CONNECTION\_LOST  
RC\_SE\_SERVER\_NOT\_AVAILABLE  
RC\_SE\_UNEXPECTED\_ERROR  
RC\_SE\_UNKNOWN\_INDEX\_NAME  
RC\_SE\_UNKNOWN\_SERVER\_NAME  
RC\_SE\_WRITE\_TO\_DISK\_ERROR

重要: 以下の戻りコードは、関数は結果を返したが、予想とは異なることを示しています。

RC\_SE\_CONFLICT\_WITH\_INDEX\_TYPE  
RC\_SE\_DICTIONARY\_NOT\_FOUND  
RC\_SE\_STOPWORD\_IGNORED  
RC\_SE\_UNKNOWN\_SECTION\_NAME  
RC\_SE\_DOCMOD\_READ\_PROBLEM



**制限**

データベースに接続し、DB2 テキスト・エクステンダー関数を使用してデータベースからハンドルを抽出してからでなければ、この関数を呼び出すことはできません。

### DesGetMatches

#### 目的

文書ハンドルで記述されるテキスト文書の強調表示情報を含むデータ・ストリームを返します。245ページの『データ・ストリーム構文』を参照してください。強調表示情報には、テキスト文脈 (最低 1 つの段落) およびその文脈内のテキスト強調表示のための情報が含まれます。

DesGetMatches は、データ・ストリームの一部だけを返します。これは、その出力構造体の一部の長さを示します。

DesGetMatches を連続して呼び出すと、テキスト文書全体の内容を入手できます。テキスト文書の終わりに達すると、RC\_SE\_END\_OF\_INFORMATION が返されます。

#### 構文

```
DESRETURN
DesGetMatches
(DESbrowseSESSION BrowseSession,
DESHANDLE DocumentHandle,
DESMATCHINFO *pMatchInfo,
DESULONG *pMatchInfoLength,
DESMESSAGE *pErrorMessage);
```

#### 関数引き数

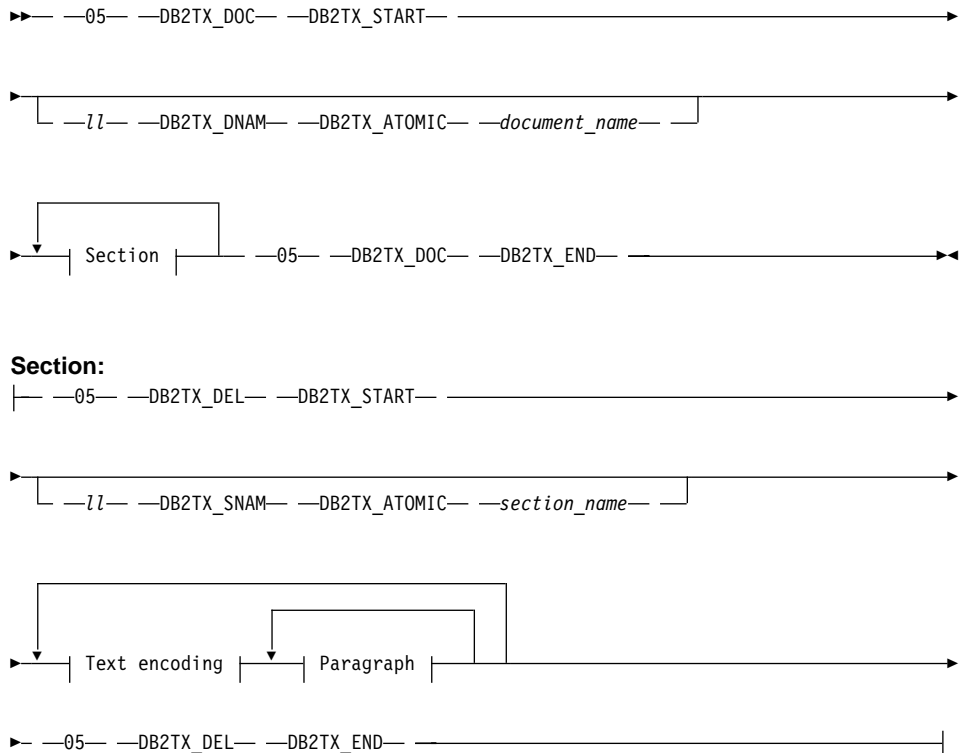
表 14. DesGetMatches 引き数

データ・タイプ	引き数	用法	説明
DESBROWSESESSION	<i>BrowseSession</i>	入力	ブラウザ・セッション・ハンドルです。
DESHANDLE	<i>DocumentHandle</i>	入力	DesOpenDocument により戻される文書ハンドルです。
DESMATCHINFO *	<i>pMatchInfo</i>	出力	受け取ったデータ・ストリーム部分を含むバッファーに対するポインタです。DesGetMatches がそのバッファーを割り当てます。
DESULONG *	<i>pMatchInfoLength</i>	出力	pMatchInfo で指されたデータ・ストリーム部分の長さです。

表 14. DesGetMatches 引き数 (続き)

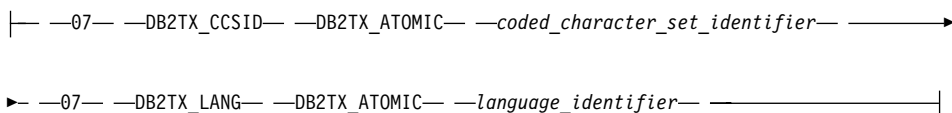
データ・タイプ	引き数	用法	説明
DESMESSAGE *	<i>pErrorMessage</i>	出力	処理系定義のメッセージ・テキスト。エラーが発生した場合は、DB2 テキスト・エクステンダーによりエラー・コードとエラー・メッセージが戻されます。アプリケーション・プログラムによってサイズ DES_MAX_MESSAGE_LENGTH のバッファが割り当てられます。 <i>pErrorMessage</i> がヌル・ポインターのとき、エラー・メッセージは返されていません。

### データ・ストリーム構文

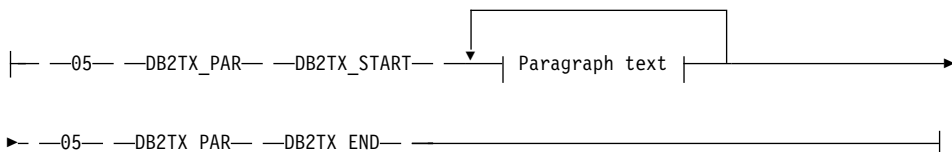


### Text encoding:

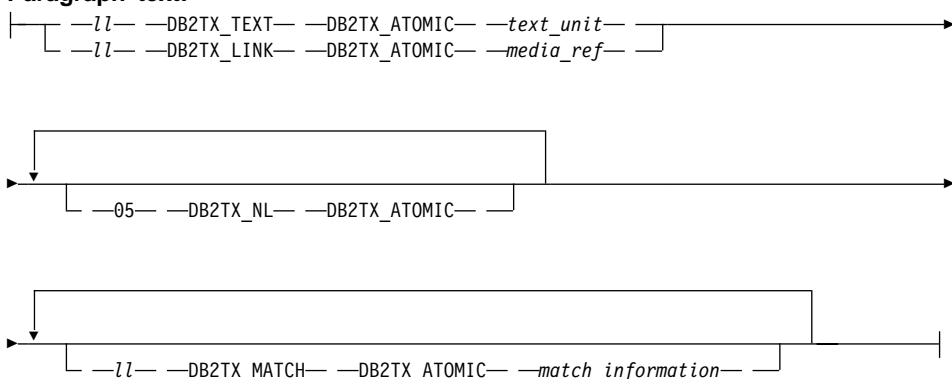
## DesGetMatches API 関数



### Paragraph:



### Paragraph text:



構文図の各セグメント、たとえば 05 DB2TX\_DOC DB2TX\_START は、整数タイプの長さフィールドで始まります。長さは、構文図中では、たとえば 05 のような明示的な数値、または変数 *ll* です。セグメントの長さには 2 バイトの長さフィールドが含まれます。

**注:** 長さはビッグ・エンディアン形式で表されます。

個々のセグメントには、以下の 1 バイト・タイプの識別子が含まれます。

### DB2TX\_START

文書や段落などのセグメントの始まりを表します。

### DB2TX\_END

セグメントの終わりを表します。

**DB2TX\_ATOMIC**

次に続く項目が、文書名や言語識別子のようにそれ以上分解できないものであることを表します。

データ・ストリーム項目は、それぞれが 2 バイトの長さを持ちます。以下はその例です。

**DB2TX\_DOC**

文書の始まりと終わりを表します。

**DB2TX\_DNAM**

文書名です。名前が指定されていない場合は、文書の識別子が使用されません。

**DB2TX\_DEL**

文書エレメントの始まりと終わりを表します。現在サポートされている唯一の文書エレメントのタイプは、*text section* です。

**DB2TX\_SNAM**

テキスト・セクションの名前を指定します。現在 DB2 テキスト・エクステンダーではただ 1 つのテキスト・セクションしかサポートされておらず、自動的に省略時の名前が使用されます。セクション名を指定しても、これは無視されます。

**DB2TX\_PAR**

現行セクション内でテキスト段落の始まりと終わりを表します。

**DB2TX\_TEXT**

現行段落内で 1 つのテキスト部分を指定します。通常、*text\_unit* にはテキストの 1 行が含まれ、TEXT 項目の後に DB2TX\_NL 項目が続きます。しかし、テキスト行をいくつかの部分に分割して、各部分をそれぞれ別の DB2TX\_TEXT 項目で指定することもできます。

テキストは、現行段落に関連した CCSID および言語を使用します。

**DB2TX\_LINK**

DB2 テキスト・エクステンダー・ハイパーメディア参照を指定します。これは現行段落の CCSID を使用します。

**DB2TX\_NL**

現行段落内の新しい行の始まりを表します。

**DB2TX\_MATCH**

現行テキスト部分内の突き合わせに関するオカレンス情報を含みます。この情報は、一対になった 2 進数として示されます。個々の対の最初の数値は現行テキスト部分内での一致語のオフセットであり、2 番目の数値は、

## DesGetMatches API 関数

文字数単位のその一致語の長さです。指定された長さが指定のテキスト部分を超えてしまうこともありえます。オフセットと長さは両方とも、ビッグ・エンディアン形式で指定された 2 バイトの値です。

### DB2TX\_CCSID

後続の段落にあるテキストの CCSID です。これは、段落の前に新しい DB2TX\_CCSID 項目が来るまで有効です。以下の CCSID が戻されます。

#### DB2TX\_CCSID\_00500

(ラテン-1 EBCDIC コード・ページ 500 で記述されたテキストの場合)

#### DB2TX\_CCSID\_04946

(ラテン-1 ASCII コード・ページ 850 で記述されたテキストの場合)

#### DB2TX\_CCSID\_00819

(ASCII コード・ページ 819 で記述されたテキストの場合)

CCSID のこれらの記号名は、DB2 テキスト・エクステンダーが提供する DES\_EXT.H で定義されます。2 バイト長の 2 進値は、ビッグ・エンディアン形式で指定されます。

### DB2TX\_LANG

後続の段落にあるテキストの言語識別子。これは、段落の前に新しい DB2TX\_LANG 項目が来るまで有効です。DB2 テキスト・エクステンダーでサポートされるすべての言語識別子の記号名は、DB2 テキスト・エクステンダーに付属のファイル DES\_EXT.H で定義されています。2 バイト長の 2 進値は、ビッグ・エンディアン形式で指定されます。

## 使用

テキスト文書の終わりに達すると、DesGetMatches は RC\_SE\_END\_OF\_INFORMATION を返します。

## 戻りコード

RC\_SUCCESS  
RC\_SE\_END\_OF\_INFORMATION  
  
RC\_INVALID\_PARAMETER  
RC\_INVALID\_SESSION  
RC\_SE\_CAPACITY\_LIMIT\_EXCEEDED  
RC\_SE\_INCORRECT\_HANDLE  
RC\_SE\_IO\_PROBLEM  
RC\_SE\_NOT\_ENOUGH\_MEMORY  
RC\_SE\_REQUEST\_IN\_PROGRESS

RC\_SE\_LS\_FUNCTION\_FAILED

RC\_SE\_UNEXPECTED\_ERROR

重要: 以下の戻りコードは、関数は結果を返したが、予想とは異なることを示しています。

RC\_SE\_DICTIONARY\_NOT\_FOUND

### 制限

この関数は、`DesOpenDocument` を呼び出してテキスト文書をオープンした後でしか呼び出すことができません。

### DesGetSearchResultTable

#### 目的

テキスト列で識別されたテキスト文書を検索するための検索引き数を使用します。検出されたテキスト項目のハンドル・データは、結果表に書き込まれます。また、ランクや一致数に関するブラウズ情報も結果表に書き込むことができます。

#### 構文

```

DESRETURN
DesGetSearchResultTable
(SQLHDBC          hdbc,
char              *pTableSchema,
DESMALLINT       TableSchemaLength,
char              *pTableName,
DESMALLINT       TableNameLength,
char              *pColumnName,
DESMALLINT       ColumnNameLength,
char              *pSearchArgument,
DESMALLINT       ArgumentLength,
char              *pResultSchema,
DESMALLINT       ResultSchemaLength,
char              *pResultTableName,
DESMALLINT       ResultTableNameLength,
DESSEARCHOPTION  SearchOption,
DESBROWSEOPTION  BrowseOption,
DESBROWSEINFO    *pBrowseInfo,
DESMESSAGE       *pErrorMessage);
    
```

#### 関数引き数

表 15. DesGetSearchResultTable 引き数

データ・タイプ	引き数	用法	説明
SQLHDBC	hdbc	入力	データベース接続ハンドルです。
char *	pTableSchema	入力	検索対象の基礎表のスキーマです。
DESMALLINT	TableSchemaLength	入力	pTableSchema の長さ (ヌル・バイトの終止符を含まない)、または DES_NTS です。
char *	pTableName	入力	検索される基礎表の名前のポインターです。
DESMALLINT	TableNameLength	入力	pTableName の長さ (ヌル・バイトの終止符を含まない)、または DES_NTS です。



表 15. DesGetSearchResultTable 引数 (続き)

データ・タイプ	引数	用法	説明
char *	pColumnName	入力	実行しようとしているテキスト検索によってアドレス指定される列名に対するポインタです。列のタイプは DESTEXTH でなければなりません。
DESSMALLINT	ColumnNameLength	入力	pColumnName の長さ (ヌル・バイトの終止符を含まない)、または DES_NTS です。
char *	pSearchArgument	入力	テキスト検索引数のポインタです。
DESSMALLINT	ArgumentLength	入力	pSearchArgument の長さ (ヌル・バイトの終止符を含まない)、または DES_NTS です。
char *	pResultSchema	入力	結果表を含むスキーマのポインタです。
DESSMALLINT	ResultSchemaLength	入力	pSchemaName の長さ (ヌル・バイト終止符を含まない)、または DES_NTS です。
char *	pResultTableName	入力	直前に作成した結果表の名前のポインタです。この表には、検索の結果が保管されます。この表の構造体については、252ページの図17を参照してください。
DESSMALLINT	ResultTableNameLength	入力	pResultTableName の長さ (ヌル・バイトの終止符を含まない)、または DES_NTS です。
DESSEARCHOPTION	SearchOption	入力	<p>ユーザーの要求が、ランキング情報、一致数、または一致するテキスト文書のハンドルのいずれであるかを指定するオプションです。</p> <p>DES_RANK DES_MATCH DES_RANKANDMATCH DES_TEXTHANDLEONLY</p> <p>このオプションは、252ページの『使用』に説明されている結果表の内容を決めます。</p>
DESBROWSEOPTION	BrowseOption	入力	予約済み。

## DesGetSearchResultTable API 関数

表 15. *DesGetSearchResultTable* 引き数 (続き)

データ・タイプ	引き数	用法	説明
DESBROWSEINFO *	<i>pBrowseInfo</i>	出力	<i>BrowseOption</i> の値に基づき、ブラウザ情報に対するポインタまたはヌルに対するポインタとなります。
DESMESSAGE *	<i>pErrorMessage</i>	出力	処理系定義のメッセージ・テキスト。エラーが発生した場合は、DB2 テキスト・エクステンダーによりエラー・コードとエラー・メッセージが戻されます。アプリケーション・プログラムによってサイズ <i>DES_MAX_MESSAGE_LENGTH</i> のバッファが割り当てられます。 <i>pErrorMessage</i> がヌル・ポインタのとき、エラー・メッセージは返されていません。

### 使用

データベースへの接続は、*DesGetSearchResultTable* を呼び出すアプリケーション・プログラムで確立しなければなりません。

名前 *pResultTableName* は、前もって作成しておいた結果表です。サンプル・ディレクトリ内のユーティリティー *DESRESTB* は、テキスト・ハンドル用の結果表を作成します。この関数を呼び出すと、結果表には検索引き数に一致するテキスト値を識別する情報が入れます。次に結果表の構造体を示します。

#### RESULT TABLE

TEXTHANDLE	RANK	MATCHES

図 17. 結果表の構造体

*TEXTHANDLE* のデータ・タイプは *DB2TEXTH* または *DB2TEXTFH* です。  
*RANK* のデータ・タイプは *DOUBLE*、そして *MATCHES* のデータ・タイプは *INTEGER* です。

*pSearchArgument* の検索引き数については、221ページの『第12章 検索引き数の構文』で説明されています。

*BrowseOption* の値が *BROWSE* である場合、DB2 テキスト・エクステンダーは、サーバー上の DB2 テキスト・エクステンダー検索プログラムからのブラウズ情報を戻します。*pBrowseInfo* は、*DesStartBrowseSession* への入力であるブラウズ情報を指します。*BrowseOption* の値が *NO\_BROWSE* のとき、*pBrowseInfo* はヌルを指します。

## 戻りコード

RC\_SUCCESS  
 RC\_NO\_BROWSE\_INFO  
 RC\_SE\_NO\_DATA

RC\_ALLOCATION\_ERROR  
 RC\_FILE\_IO\_PROBLEM  
 RC\_INTERNAL\_ERROR  
 RC\_INVALID\_BROWSE\_OPTION  
 RC\_INVALID\_PARAMETER  
 RC\_INVALID\_SEARCH\_OPTION  
 RC\_INVALID\_SESSION  
 RC\_PARSER\_INVALID\_ESCAPE\_CHARACTER  
 RC\_PARSER\_SYNTAX\_ERROR  
 RC\_RESULT\_TABLE\_NOT\_EXIST  
 RC\_SE\_COMMUNICATION\_PROBLEM  
 RC\_SE\_EMPTY\_INDEX  
 RC\_SE\_EMPTY\_QUERY  
 RC\_SE\_FUNCTION\_DISABLED  
 RC\_SE\_FUNCTION\_IN\_ERROR  
 RC\_SE\_INCORRECT\_HANDLE  
 RC\_SE\_INDEX\_DELETED  
 RC\_SE\_INDEX\_NOT\_ACCESSIBLE  
 RC\_SE\_INDEX\_SUSPENDED  
 RC\_SE\_INSTALLATION\_PROBLEM  
 RC\_SE\_IO\_PROBLEM  
 RC\_SE\_MAX\_NUMBER\_OF\_BUSY\_INDEXES  
 RC\_SE\_NOT\_ENOUGH\_MEMORY  
 RC\_SE\_PROCESSING\_LIMIT\_EXCEEDED  
 RC\_SE\_QUERY\_TOO\_COMPLEX  
 RC\_SE\_SERVER\_BUSY  
 RC\_SE\_SERVER\_CONNECTION\_LOST  
 RC\_SE\_SERVER\_NOT\_AVAILABLE

## DesGetSearchResultTable API 関数

RC\_SE\_UNEXPECTED\_ERROR  
RC\_SE\_UNKNOWN\_INDEX\_NAME  
RC\_SE\_UNKNOWN\_SERVER\_NAME  
RC\_SE\_WRITE\_TO\_DISK\_ERROR  
RC\_SQL\_ERROR\_NO\_INFO  
RC\_SQL\_ERROR\_WITH\_INFO  
RC\_TEXT\_COLUMN\_NOT\_ENABLED

重要: 以下の戻りコードは、関数は結果を返したが、予想とは異なることを示しています。

RC\_SE\_CONFLICT\_WITH\_INDEX\_TYPE  
RC\_SE\_DICTIONARY\_NOT\_FOUND  
RC\_SE\_STOPWORD\_IGNORED  
RC\_SE\_UNKNOWN\_SECTION\_NAME  
RC\_SE\_DOCMOD\_READ\_PROBLEM

## DesOpenDocument

## 目的

ブラウザ・セッション・ポインター、ハンドル、およびテキスト文書の解析に辞書を使用する必要があるかどうかを示すオプション (DES\_EXTENDED または DES\_FAST)。この関数は、文書テキストおよび強調表示情報を入手するためのハンドルに対応するテキスト文書を作成し、DesGetMatches の反復呼び出しに使用される文書ハンドルを戻します。

## 構文

```

DESRETURN
DesOpenDocument
(DES_BROWSESESSION BrowseSession,
SQLCHAR *pHandle,
DES_USHORT HandleLength,
DES_MATCHMODE MatchMode,
DESHANDLE *pDocumentHandle,
DESMESSAGE *pErrorMessage);

```

## 関数引き数

表 16. DesOpenDocument 引き数

データ・タイプ	引き数	用法	説明
DES_BROWSESESSION	<i>BrowseSession</i>	入力	ブラウザ・セッション・ハンドルです。
SQLCHAR *	<i>pHandle</i>	入力	データベースから抽出されるハンドルへのポインターです。
DES_USHORT	<i>HandleLength</i>	入力	<i>pHandle</i> の長さです (DES_NTS は使用できません)。
DES_MATCHMODE	<i>MatchMode</i>	入力	強調表示情報を検出するのに辞書を使用するかどうかを決定するモードです。  <b>DES_FAST</b> 辞書を使用しない。  <b>DES_EXTENDED</b> 辞書を使用する。
DESHANDLE *	<i>pDocumentHandle</i>	出力	DesGetMatches の反復呼び出しのための文書ハンドルです。

## DesOpenDocument API 関数

表 16. DesOpenDocument 引き数 (続き)

データ・タイプ	引き数	用法	説明
DESMESSAGE *	<i>pErrorMessage</i>	出力	処理系定義のメッセージ・テキスト。エラーが発生した場合は、DB2 テキスト・エクステンダーによりエラー・コードとエラー・メッセージが戻されます。アプリケーション・プログラムによってサイズ <code>DES_MAX_MESSAGE_LENGTH</code> のバッファが割り当てられます。 <i>pErrorMessage</i> がヌル・ポインターのとき、エラー・メッセージは返されていません。

### 使用

DES\_FAST および DES\_EXTENDED は、ブラウザ・テキストで強調表示する用語を検出するために言語処理を使用することを示します。詳細については、273ページの『ブラウザのための言語的な処理』を参照してください。基本テキスト解析を使用するには DES\_FAST を使用し、拡張突き合わせを使用するには DES\_EXTENDED を使用してください。

SQL データ・タイプと C データ・タイプ間のマッピングでは、ハンドルとして SQL 記号名 `SQL_VARBINARY` を使用する必要があります。 `TextHandle` 値の C 表示を指すホスト変数のタイプは、 `SQLCHAR*` です。

DB2 テキスト・エクステンダーにより、ブラウザ情報用に記憶域が割り当てられます。アプリケーション・プログラムで、 `DesFreeBrowseInfo` を呼び出してこの記憶域と関連する資源を解放しなければなりません。

`TextHandle` 値はビット・データで、 '¥0' 文字をいくつか含んでいるため、 `pHandle` の長さを指定する必要があります。

呼び出し元は、 `pHandle` によって参照されたテキスト文書を含む表への読み取りアクセス権を持っている必要があります。

### 戻りコード

RC\_SUCCESS

RC\_ALLOCATION\_ERROR

RC\_INTERNAL\_ERROR

RC\_INVALID\_MATCH\_OPTION

RC\_INVALID\_PARAMETER  
RC\_INVALID\_SESSION  
RC\_SE\_DOCUMENT\_NOT\_ACCESSIBLE  
RC\_SE\_DOCUMENT\_NOT\_FOUND  
RC\_SE\_INCORRECT\_HANDLE  
RC\_SE\_IO\_PROBLEM  
RC\_SE\_LS\_FUNCTION\_FAILED  
RC\_SE\_LS\_NOT\_EXECUTABLE  
RC\_SE\_MAX\_NUMBER\_OF\_BUSY\_INDEXES  
RC\_SE\_NOT\_ENOUGH\_MEMORY  
RC\_SE\_REQUEST\_IN\_PROGRESS  
RC\_SE\_UNKNOWN\_INDEX\_NAME  
RC\_SE\_UNEXPECTED\_ERROR

## 制限

この関数は、DesStartBrowseSession 呼び出しによってブラウザ・セッションを開始した後でしか呼び出すことができません。

### DesStartBrowseSession

#### 目的

ブラウザ・セッションを開始し、テキスト文書をブラウザしてその一致語を強調表示するための環境を確立します。この関数は、DesGetBrowseInfo または DesGetSearchResultTable のどちらかからブラウザ情報に対するポインターを受け取り、他のブラウザ関数が使用するためのブラウザ・セッション・ハンドルを戻します。

#### 構文

```

DESRETURN
DesStartBrowseSession
(
    DESBROWSEINFO BrowseInfo,
    char *pUserId,
    DESSMALLINT UserIdLength,
    char *pPassword,
    DESSMALLINT PasswordLength,
    DESBROWSESESSION *pBrowseSession,
    DESMESSAGE *pErrorMessage);
    
```

#### 関数引き数

表 17. DesStartBrowseSession 引き数

データ・タイプ	引き数	用法	説明
DESBROWSEINFO	<i>BrowseInfo</i>	入力	テキスト文書内の一致語をブラウザして強調表示するために必要な情報に対するポインターです。ポインターは DesGetSearchResultTable または DesGetBrowseInfo によって戻されます。
char *	<i>pUserId</i>	入力	データベースのユーザー ID です。
DESSMALLINT	<i>UserIdLength</i>	入力	データベースのユーザー ID の長さです。
char *	<i>pPassword</i>	入力	データベースのパスワードです。
DESSMALLINT	<i>PasswordLength</i>	入力	データベースのパスワードの長さです。
DESBROWSESESSION *	<i>pBrowseSession</i>	出力	他のブラウザ関数が使用するブラウザ・セッションのハンドルです。



表 17. DesStartBrowseSession 引き数 (続き)

データ・タイプ	引き数	用法	説明
DESMESSAGE *	<i>pErrorMessage</i>	出力	処理系定義のメッセージ・テキスト。エラーが発生した場合は、DB2 テキスト・エクステンダーによりエラー・コードとエラー・メッセージが戻されます。アプリケーション・プログラムによってサイズ DES_MAX_MESSAGE_LENGTH のバッファが割り当てられます。 <i>pErrorMessage</i> がヌル・ポインターのとき、エラー・メッセージは返されていません。

## 使用

この関数により、テキスト文書をブラウズするためのブラウズ・セッションがオープンされます。ユーザーがデータベースへのアクセス権限を持っているかをチェックするために、ユーザー ID とパスワードの入力が要求されます。

ブラウズ・セッションのクローズは、DesEndBrowseSession を呼び出して行います。

*BrowseInfo* は、ブラウズ情報の作成に使用される検索引き数と基本テキスト列によって異なります。

## 戻りコード

RC\_SUCCESS

RC\_ALLOCATION\_ERROR

RC\_INVALID\_BROWSE\_INFO

RC\_INVALID\_PARAMETER

RC\_INTERNAL\_ERROR

RC\_SE\_NOT\_ENOUGH\_MEMORY

RC\_SE\_UNEXPECTED\_ERROR

RC\_SQL\_ERROR\_NO\_INFO

RC\_SQL\_ERROR\_WITH\_INFO

## 制限

この関数を呼び出す前に、DesGetBrowseInfo または DesGetSearchResultTable を適切なブラウズ・オプションを指定して呼び出しておかなければなりません。

## DesStartBrowseSession API 関数

ん。

---

## 第14章 サンプル API プログラム

DB2 テキスト・エクステンダーは、SAMPLES ディレクトリー内にサンプル・プログラム DESSAMP1.C を備えています。

DESSAMP1.C は、DesGetSearchResultTable 関数とユーザーのブラウザー・プログラムを使用するプログラムの 1 例です。これは、125ページの図16 に示されている一連の API 関数呼び出しの後に続きます。

使用可能データベースおよび使用可能テキスト列にアクセスする必要があります。このプログラムを実行するには、次のようにします。

1. 任意選択。ソース・ファイル DESSAMP1.C をクライアント・マシンのローカル・ディレクトリーにコピーします。
2. 提供されている makefile (OS/2、Windows NT、および Windows 2000 の場合は `dessamp.mak`、UNIX システムの場合は `dessamp`) を使用して、サンプル・ファイルをコンパイルし、リンクします。
3. DESRESTB ユーティリティーを実行して、サンプル・コードで使用したい結果表をデータベース内に作成します。

DESRESTB データベース名

この表は、検索結果などの情報を保管するために使用されるもので、その構造は次のとおりです。

列	データ・タイプ
HANDLE	DB2TX.DB2TEXTH または DB2TX.DB2TEXTFH
RANK	DOUBLE
MATCHES	INTEGER

## サンプル API プログラム

---

## 第15章 基本語形索引と一致語句索引の言語的な処理

DB2 テキスト・エクステンダーでは、以下のような検索の分野で言語的な処理を行えます。

- **索引付け。** DB2 テキスト・エクステンダーが文書を分析してテキスト索引に保管される用語を抽出する場合、テキストが言語的に処理されてその索引に最適な用語を抽出します。これは、検索をできるだけ簡単かつ早くするために行われます。
- **検索。** DB2 テキスト・エクステンダーが、指定された検索項目のオカレンスを含む文書を検出するために文書索引内を検索するときに、検索項目も言語的に処理されて索引用語と突き合わされます。
- **ブラウズ。** 検索後に検出された文書をブラウズすると、言語的な処理が行われて文書内で検出された用語が強調表示されます。

---

### 索引作成時の言語的な処理

DB2 テキスト・エクステンダーは、文書の索引付けと検索を行う際に、テキストの言語的な分析を行います。以下の表から分かるとおり、言語的な処理の量は、索引のタイプによって異なります。Ngram 索引では、言語的な処理は行われません。

文書の索引付けに使用される言語処理は以下のとおりです。

- 基本テキスト分析
  - 用語の認識 (トークン化)
  - 用語の標準形への正規化
  - 文の認識
- 用語の基本形への変形
- 停止ワードのフィルター処理
- 分解 (複合語を分割)

264ページの表18 は、索引タイプが**基本語形**で、追加の索引特性が要求されていない場合に、用語を索引付けする方法をまとめて示しています。

## 索引作成時の言語的な処理

表 18. 基本語形索引の用語抽出

文書テキスト	索引中の用語	言語的な処理
Mouse Käfer	mouse kaefer	基本テキスト分析 (正規化)
mice swum	mouse swim	基本形への変形
system-based  Wetterbericht	system-based, system base wetterbericht, wetter bericht	分解
a report on animals	report animal	停止ワードのフィルター処 理。停止ワードは a と on。

比較のため、表19 には索引タイプが**一致語句**の場合に用語を索引付けする方法をまとめています。

表 19. 一致語句索引の用語抽出

文書テキスト	索引中の用語	言語的な処理
Mouse Käfer	Mouse Käfer	正規化なし
mice swum	mice swum	基本形への変形なし
a report on animals	report animals	停止ワードのフィルター処理。 停止ワードは a と on。
system-based Wetterbericht	system-based Wetterbericht	分解なし

### 基本テキスト分析

DB2 テキスト・エクステンダーは、電子辞書を使用せずに基本テキスト分析を処理します。

#### 非英数字文字を含む用語の認識

文書を索引付けする場合には、非英数字文字を含んだ用語でも認識されます。たとえば、「\$14,225.23」、「mother-in-law」、および「10/22/90」などです。

以下のものは、用語の一部とみなされます。

アクセント

通貨記号

数値区切り記号 (「/」や「.」など)

E メール・アドレスに含まれる「@」(英語の場合のみ)

「+」記号

以下のものを含む用語の認識には、言語に固有の規則も使用されます。

- ラテン系言語に含まれるアクセントつき接頭部 (フランス語の l'aventure など)。
- 日付、時刻、数字の国別の書式。
- 英語では「/」を使用して示される (mission/responsibility など) 代替語の表記。
- securita' などのイタリア語の後続アポストロフィ。これは通常タイプ印刷のイタリア文に出てきます。アクセント記号付きの文字が文字セットになれば、イタリア語テキストを文字入力する場合は通常はその文字の後にアクセント記号を入力します。たとえば、「à」は「a」と入力されます。

### 用語の標準形への正規化

正規化は、大小文字が混合している用語や、アクセント記号付き文字や特殊文字を含む用語を、標準形に変形します。これは、索引タイプが基本語形である場合に、省略時処理として行われます。(一致語句索引の場合、大文字小文字は変わりません。つまり、検索では大文字小文字を区別します。)

たとえば、Computer という用語は、computer と索引付けされ、大文字は小文字に変形されます。computer という用語を検索すると、computer のオカレンスだけでなく、Computer のオカレンスも検出されます。索引付け中に正規化を用いると、文書内の用語に大文字が含まれているかどうかに関係なく、同じように索引付けされるという効果があります。

正規化は、索引付け中だけでなく検索中にも適用されます。検索項目の中の大文字は、検索が行われる前に小文字に変形されます。たとえば、検索項目が Computer であると、検索に使用される用語は computer です。

アクセント記号付き文字と特殊文字は、同じ方法で正規化されます。École といった、école のどのバリエーションを使用しても、école、Ecole などが検出されます。Bürger で buerger が、Maße で masse が検出されます。

## 索引作成時の言語的な処理

検索項目にマスキング (ワイルドカード) 文字が含まれている場合、正規化が行われてからそのマスキング文字が処理されます。例: Bür\_er は buer\_er となります。

### 文の認識

同じ文中にある用語を検索することができます。このことを可能にするため、個々の文書は索引付け中に分析されてそれぞれの文の終了位置が検出されません。

DB2 テキスト・エクステンダーでは、以下の 2 種類の方式で文の終了位置が認識されます。

- **Universal Unicode Tokenizer** (アラビア語とヘブライ語以外の言語の場合)  
この方式の方が簡単で高速です。トークン化機能は、ピリオド、感嘆符、疑問符などのうち、前にトークン文字 (文字など) があり、後ろに空白、タブ、改行文字があるものを探します。これが本当に文の終了位置であり、単なる省略語の末尾のピリオドではないことを検査するために、言語固有の省略語のリストが検査されます。
- **POE** ベースのトークン化機能 (アラビア語とヘブライ語の場合)  
このトークン化機能の方が、言語の面ではより拡張されていますが、必要な処理能力は高くなります。このトークン化機能では、基本的には句読点を突き合わせて文の終了位置が検索されますが、特殊な入力タイプや語数も手掛かりとして用います。

### POE ベースのトークン化機能による文の認識

POE ベースのトークン化機能は、句読点の規則と言語固有の処理 (省略語の処理を含む) を使用して、文 (または文の一部) の境界を判別します。省略語処理機能のレベルは言語によって大きく異なります。単一バイト・コード・ページを使用する言語のほとんどについては、関連した省略語補足辞書が POE に付属しています。普通 2 バイト言語ではピリオドを含む省略語を使用しないので、省略語補足辞書は使用できません。

基本的には、句読点を突き合わせて文の終了位置が判別されます。以下の表には、終端の句読文字とそのグラフィック文字グローバル ID (GCGID) がリストされています。

SBCS 文字の GCGID	DBCS 文字の GCGID	説明
SP110000	SP110080	ピリオド
SP020000	SP020080	感嘆符
SP150000	SP150080	疑問符



SBCS 文字の GCGID	DBCS 文字の GCGID	説明
SP140000	N/A	セミコロン (ギリシャ語の疑問符)
N/A	JQ730080	2 バイトの丸

以下のいずれも該当しなければ、終端の句読文字 (ピリオド、感嘆符、疑問符など) は文の終了位置を示していると見なされます。

- 以下の表にリストされている終結句読文字 (右小括弧など) が、終端の句読文字の後にある。

SBCS 文字の GCGID	DBCS 文字の GCGID	説明
SP070000	SP070080	右小括弧
SP040000	SP040080	二重引用符
SP050000	SP050080	単一引用符
SP180000	SP070083	角付き二重引用符
N/A	SM140080	右中括弧
N/A	SM080080	右大括弧
N/A	JQ720080	単一角引用符
N/A	JQ720081	二重角引用符
N/A	SP200080	右単一鍵引用符
N/A	SP220080	右二重鍵引用符
N/A	SP070081	右甲括弧
N/A	SP070082	右単一角付き引用符
N/A	SP070084	右角小括弧
N/A	SP370080	縦の右単一角引用符
N/A	SP370081	縦の右二重角引用符
N/A	SP250084	縦の右角小括弧
N/A	SP250080	縦の右小括弧
N/A	SP350080	縦の右中括弧
N/A	SP250081	縦の右甲括弧
N/A	SP250083	縦の右角度付き二重引用符
N/A	SP250082	縦の右単一角付き引用符

## 索引作成時の言語的な処理

注: 「N/A」と示された項目は POE によって終結句読文字と見なされません。縦の終結句読文字は中国語だけでサポートされます。

例:

...this sentence ends with two parentheses.)

上記の例で、2 つ目の小括弧が文の終了位置として検出されます。しかし、ドイツ語では、終結引用符の後にコンマがあると、その引用符は文の終了位置とは見なされません。

- 終端の句読文字の後に別の終端文字がある。

例:

This is a strong exclamation!!!

最後の感嘆符が文の終了位置として検出されます。

- 終端の句読文字の前に数値か句読文字があり、後に数字がある。「1.25」や「.314」などのストリングは文の終了位置とは見なされません。
- 終端の句読文字がピリオドで、文の終了位置として使用できない省略語の一部である。言語別に限定された省略語処理が実行されます。
- 終端の句読文字がピリオドで、後に空白文字 (ブランクや改行文字など) が無い。「III.IV」などのヘッダーは文の終了位置として検出されません。

また POE ベースのトークン化機能は、省略語処理を行って、ピリオドが省略語の一部なのかそれとも文の終了位置を示すのかを判別します。省略語補足辞書に省略語を追加できます。POE ベースのトークン化機能に辞書が渡されない場合は、単一文字の後にピリオドがあるとすべて省略語と見なされます。その他の省略語処理は行われません。

省略語を通常の語の後にピリオドが付いたものと間違える可能性があるため、特定のテキストが省略語かどうか判別しにくい場合がよくあります。たとえば、以下の文の文字「no.」がこれに該当します。

Enter the no. of exemptions you are claiming.  
Answer each question with yes or no.

特定のテキストが省略語であると分かっているにもかかわらず、依然として文の終了位置かどうかの判別は困難です。決して文の終了位置には使用されない省略語もあれば、使用される場合もある省略語もあります。たとえば、以下の文中の省略語「Hwy.」がこれに該当します。

The drive along Hwy. 1 to Santa Cruz was beautiful.  
Many people speak highly of the Pacific Coast Hwy.

省略語は判別しにくいことがあり、文の終了位置でないこともあるので、POE では省略語を検出した後に分類します。文の終了位置に使用されることのある

省略語の一部にピリオドがあると、それ以上の処理は実行されません。POEにより、省略語が文の終了位置ではないと判別されると、ピリオドを表すトークンが省略語のテキストのトークンと結合されます。それ以外の場合、ピリオドを表すトークンは独立した状態のままです。

POE ベースの省略語処理では、以下の 3 種類の基準を使用して、ピリオドが省略語の一部かどうか判別されます。

- 単一文字の後にピリオドがある場合は、すべて省略語と見なされる。単一文字の省略語は、文の終了位置の可能性ありとして分類されます。
- 入力省略語補足辞書に含まれる語は、常に省略語と見なされる。補足中にある省略語が文の終了位置であるかどうかは、補足中のその語に関連した情報によって判別されます。たとえば、米国英語の省略語補足辞書で、「Mr.」は文の終了位置にできない省略語として示され、「etc.」は文の終了位置になることのある省略語として示されています。
- 長さ 2 ~ 6 文字の語の後にピリオドがある場合は、入力辞書や省略語補足辞書になくても、省略語と見なされる。「Jrnl. Comp. Ling.」などの事例を処理する場合はこれに該当します。辞書検索によって判別される省略語は、常に文の終了位置の可能性ありとして扱われます。

省略語が文の終了位置の可能性のあるものとして識別されると、その後のテキストが POE により検査され、その省略語が現行文の終了位置であるかどうか決定されます。その方法は、次の語の先頭が大文字かどうかを検査するというものです。

省略語の後に複数の改行、新しい文、または新しい段落のデータ・エレメントがある場合は、POE では文の終了位置に達したと見なします。また、後のテキストが逆疑問符か逆感嘆符の場合は、文の終了位置を示すマーカーが出力中に挿入されます。

POE によりピリオドが、文の終了位置にない省略語の一部であると判別された場合は、引き続き文の区切り文字が検索されます。それ以外の場合は、他の終端句読文字の例外条件（後に終端句読点や終結句読点があるか）が検査されてから、文の終了位置としてマーキングされます。

## 用語の基本形への変形 (lemmatization)

たとえば、基本語形索引で、mouse を検索し、mice を検出することができます。索引付けの際に用語は基本形に変形されるので、mice という用語は mouse と索引付けされます。その後、mouse という検索項目を使用するとその文書が検出されます。mice で検索してもその文書は検出されます。

## 索引作成時の言語的な処理

mouse という用語のどのようなバリエーションが文書中に現れても、あるいはそのようなバリエーションを検索項目として使用しても、結果として、mice に関する情報を含む文書を検出する効果が得られます。

同様に、動詞の活用形は原形に変形されます。たとえば、bought は buy となります。

### 停止ワードのフィルター処理

停止ワードとは、文書内に頻繁に出てくる前置詞や代名詞などを指します。したがって、検索項目としては適切ではありません。このような語は、それぞれの辞書に関連した停止ワード・リストの中にあり、索引付けのプロセスからは除外されます。

停止ワード処理は、大文字小文字の区別をしません。したがって、about という停止ワードによって、文の最初の語である About も除外されます。各国語で提供される停止ワード・リストは、修正できます。

Ngram 索引には停止ワード・リストはありません。

### 分解 (複合語の分割)

ドイツ語やオランダ語といったゲルマン系諸言語には、メール (Versand) のラベル (Etiketten) を意味する Versandetiketten のような複合語が多くあります。このような複合語は、それぞれの構成要素に分割することができます。

一致語句索引の場合、複合語は変更されず 1 つの語として索引付けされます。基本語形索引では、複合語は索引付け中に分割されます。基本語形索引を使用する場合、複合語は検索時に分割されます。

構成要素は、それらが 1 つの文に含まれていれば、文書中に任意の順序で出現しても検出されます。たとえば、ドイツ語の Wetterbericht (天気予報) を検索する場合は、Bericht über das Wetter (天気に関する予報) という句を含む文章も検出されます。

以下の場合に用語の分割が試みられます。

- 用語の言語に複合語が使用されている場合。
- 用語がある程度の最小の長さで分けられる場合。
- 用語自体が電子辞書の項目ではない場合。ドイツ語の Geschäftsbericht (業務報告書) のようによく使用される複合語はドイツ語辞書に入っています。

用語の分割が不可能であることがわかった場合、用語の構成要素部分はその基本形式に変形されます。以下に、デンマーク語、ドイツ語、オランダ語からい

くつかの例を示します。

複合語	構成要素部分
børsmæglerselskab	børsmæglerselskab børs mægler selskab
Kindersprachen	kindersprache kind sprache
probleemkinderen	probleemkinderen probleemkind kind probleem

## 検索のための言語的な処理

照会処理では、検索の再呼び出し率を上げるように、すなわち、より多くの関連文書が見つかるように、検索項目に融通性をもたせることを目指しています。この目標を達成するために、照会用語に対して、拡張と変形の 2 つの基本操作が適用されます。さらに、用語検索操作には、拡張と変形の両方の操作を含むものがあります。

- 拡張操作では、検索項目から 1 つの語あるいは複数語からなる用語を取り出して、それを代替検索用語セットと関連付けます。これは複数語からなる用語自体の場合もあります。ソース表記とターゲット表記セットは、DB2 テキスト・エクステンダーの照会言語でブール OR 表記を形成します。ソース用語は、拡張されても未変更のまま残されるので、ある程度索引タイプから独立しています。拡張操作は次のとおりです。

同義語の拡張

シソーラスの拡張

- 変形によって、検索項目がユーザーが指定したものより普遍的な形式に変更されます。これは検索項目を変更するため、変形は索引タイプに依存して、変更された用語が必ず一致するようにします。したがって、DB2 テキスト・エクステンダーは、照会の対象となる索引から変形情報を取り出します。変形は次のとおりです。

Lemmatization (269ページの『用語の基本形への変形 (lemmatization)』を参照)

正規化 (265ページの『用語の標準形への正規化』を参照)

停止ワード (270ページの『停止ワードのフィルター処理』を参照)

## 検索のための言語的な処理

- 操作によっては、検索項目の変更および代替用語のセットを使用する拡張の両方を行うものがあります。このような操作は固有の変形のため、やはり索引に含まれている情報に依存します。以下の操作はこのクラスに属していません。

文字と語のマスキング

発音の拡張

## 同義語

同義語は、意味的に関係のある語です。通常、このような語はソース用語と同じ語クラス (名詞、動詞など) をもちます。同義語は、言語ごとの独立したファイルから得られます。これら常に基本形で戻され、若干の例外を除き、複数語の用語ではありません。同義語を探すときには、検索項目語は常に、それらの基本形に変形されます。以下に 3 種の言語による同義語の例を示します。

- 英語

*word:*

comment remark statement utterance term expression  
communication message assurance guarantee warrant bidding command  
charge commandment dictate direction directive injunction instruction  
mandate order news advice intelligence tidings gossip buzz cry  
hearsay murmur report rumor scuttlebutt tattle tittle-tattle  
whispering

- フランス語

*mot:*

expression parole terme vocable lettre billet missive épître  
plaisanterie

- ドイツ語

*Wort:*

Vokabel Bezeichnung Benennung Ausdruck Begriff Terminus  
Ehrenwort Brocken Bekräftigung Versprechen Zusicherung Gelöbnis  
Beteuerung Manneswort Schwur Eid Ausspruch

## シソーラスの拡張

検索項目は、特定の関係からアクセスできるシソーラスの用語を使用して拡張することができます。このような関係は、階層関係 (「用語の限定」関係など)、関連関係 (「関連用語」関係など)、あるいは同義語の関係が考えられます。シソーラス用語は、複数語用語である場合がよくあります。

275ページの『シソーラスの概念』で、シソーラス拡張について詳細に説明してあります。

シソーラス検索を行う際には、検索項目（検索開始時の用語）は正規化されません。シソーラス検索で検出された語は、索引のタイプに従って基本形に変形されます。

### 発音の拡張

発音の拡張では、1つの語を発音の似た語のセットにより拡張します。これは特に、検索する用語の正確なつづりがわからない場合に役に立ちます。

### 文字と語のマスキング

マスキングは、非言語的拡張手法で、これにより、通常の表記が、それを満たすすべての索引付き語の分離で置き換えられます。マスクされた表記もそのいずれの拡張も lemmatization、停止ワードの抽出、またはそれ以外のどの拡張手法にも従いません。その結果、たとえば、マスクされた用語 `swu*` を使用して検索するとき、`swum` のような不規則な動詞形式が一致語句索引で一致しますが、この形式が `swim` とい基本形に変形されている基本語句索引では一致しません。

語のマスキングを使用する場合、特に大きいサイズの索引を検索しているときに、パフォーマンスが低下することがあります。

---

## ブラウズのための言語的な処理

言語的な処理は、検索後に検出された文書をブラウズする際にも使用されます。これは以下の2つの段階で行われます。

1. 基本テキスト分析：正規化と用語拡張
2. 拡張突き合わせ

### ステージ 1: 正規化と用語拡張

最初の段階は、電子辞書を使わずに行われます。

#### 正規化

正規化は、264ページの『基本テキスト分析』で説明されています。

#### 用語拡張

用語拡張は、用語を基本形に変形することの逆の処理です。索引が基本語形索引の場合、検索項目は検索前に基本形に変形されます。

同様に、基本語形索引の場合、文書の利用語は索引に追加される前に基本形に変形されます。したがって、文書は用語の基本形に基づいて検出されます。

## ブラウザのための言語的な処理

しかし、検出された文書をブラウザする場合には、基本形のあらゆる変化形を強調表示することができます。このような変化形を強調表示するために、検出された用語の拡張が行われます。

辞書で検出される各用語の変化形 (語形変化) はすべて作成可能です。ドイツ語 gehen (行く) に対して生成される変化語形は次のとおりです。

```
gegangen geh     gehe     gehen     gehend  gehest  gehet  gehst  
ging       ginge    gingen   gingest  ginget  gingst  gingt  geht
```

### ステージ 2: 拡張突き合わせ

第 2 段階は拡張突き合わせです。この段階は、基本テキストの分析および正規化により検出用語を強調表示できない場合に、まれに使用されます。拡張突き合わせでは、一致度の低い用語を検出します。

DesOpenDocument API 関数にパラメーターとして DES\_EXTENDED を指定することによって、拡張突き合わせを選択します。

拡張突き合わせでは、基本語形索引付けの際に行われる処理と同じ言語的な処理を使用します。

拡張突き合わせにより追加の一致項目を検出できるのは以下の場合です。

- 検索項目は、マスキング文字を含む変化形です。  
マスキング文字が処理され、検索項目の語幹が変形されて、該当する文書が検出されます。拡張突き合わせが行われないと、指定の検索基準に合ったテキストを強調表示することはできません。  
例: 文書に変化形 swam が含まれている場合
  - この用語は、索引付け中に swim に変形される。
  - 検索項目が swi% の場合、語幹が swim に変形されるので上記の文書は検出される。
  - 拡張突き合わせを使用しないと、swi% という語と一致する語しか強調表示されない。拡張突き合わせを使用すると、変化形である swam も強調表示される。
- 複合語が索引付けされている場合  
ゲルマン系諸言語の文書で、中に複合語があり、かつ基本語形索引を使って索引付けされている場合、その文書索引には複合語とその各部分が保存されます。複合語の一部分を検索すると、その複合語を含んだ文書が検出されますが、拡張突き合わせを使用しないとその語は強調表示されません。  
例: 文書にドイツ語 Apfelbaum (りんごの木) が含まれている場合



- この語は、言語の索引付け中に `apfel` と `baum` に変形される。
  - 索引で `baum` という用語を検索すると、`Baum` という用語とそれを含む文書が索引内で見つけられる。
  - 拡張突き合わせを使用しないと、文書には `Apfelbaum` が含まれているが `Baum` は含まれていないため、どの用語も強調表示されない。拡張突き合わせを使用すると、`Apfelbaum` という複合語は分割され、`Baum` 部分が見つけられて強調表示される。
- 行末の語がハイフン付けされている場合
 

ワード・プロセッサでハイフンが自動的に挿入される場合は、ハイフン付けされた語を検出して強調表示することができます。しかし、ユーザーがハイフンを入力する場合は、その語を含んだ文書の検出は行われますが、拡張突き合わせを使用しない限りその語は強調表示されません。

例: 次のように、行の終わりで中断されたためにハイフン付けされている `container` という語が文書に含まれている場合

```
Another name for a folder is a con-
tainer.
```

    - この語は、索引付け時に `container` に正規化される。
    - 索引を使用して `container` という語を検索すると、その語を含んだ用語と文書が検出される。
    - 文書内で `container` と一致するあらゆる語を強調表示する試みが行われる。拡張突き合わせを使用しないと、`con-tainer` 内のハイフンがユーザーによってタイプされたのではなく、テキスト・プロセッサによって挿入された場合にのみ一致します。

---

## シソーラスの概念

シソーラスとは、通常、特定の対象領域を網羅する意味的に関連のある用語の制御可能な語いのことです。これは、各用語をノードによって表したセマンティクス・ネットワークとして視覚化することができます。2つの用語が互いに関連している場合、それらのノードは、関係名ラベルの付いたリンクによって接続されます。特定の用語に直接関連しているすべての用語には、そのノードからでてくるすべての接続をたどることによって達することができます。間接的に関連している用語には、前のステップで到達したノードから出ているすべての接続をさらにたどることによって達することができます。276ページの図18は、ごく小さいシソーラスの構造例を示しています。

## シソーラスの概念

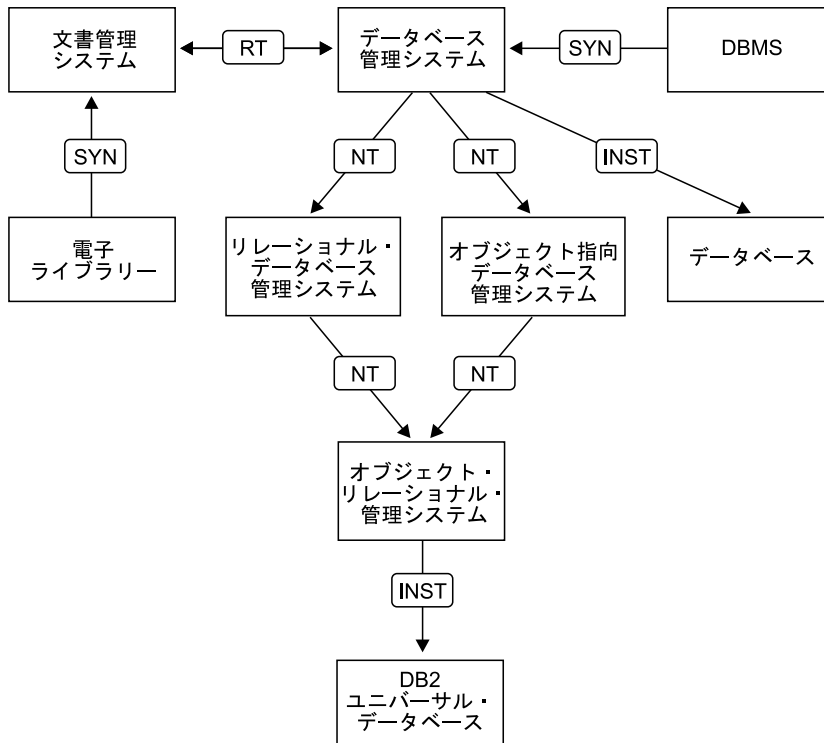


図 18. ネットワークとして表されたシソーラス

DB2 テキスト・エクステンダーにより、事前に作成済みのシソーラスからさらに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。照会でシソーラスの拡張を使用する方法については、221ページの『第12章 検索引き数の構文』を参照してください。

検索アプリケーションで使用するためにシソーラスを作成するには、シソーラス定義ファイルを内部形式（シソーラス辞書）にコンパイルする必要があります。

基本語形索引および一致語句索引で使用される辞書形式は、Ngram 索引で使用されるものとは異なります。したがって、2つの異なるシソーラス・コンパイラーが製品に付属しています。それらは、基礎となる概念が若干異なるだけでなく、必要とするソース形式も異なっています。したがって、検索アプリケーション用のシソーラスを定義し始める前に、まずどの索引タイプを使用するかを決定する必要があります。

シソーラスの基本構成要素は「用語」と「関係」です。

## 用語

用語とは、シソーラスの対象領域内の概念を表す語または表記のことです。たとえば、以下の用語は、1 つまたは複数のシソーラス内に含まれるものです。

data processing  
helicopter  
gross national product

DB2 テキスト・エクステンダー・シソーラスでは、用語は記述用語または非記述用語のいずれかに分類することができます。記述用語とは、同義語クラス内の、索引付けおよび検索に望ましい用語のことです。クラス内のその他の用語は非記述用語と呼ばれます。たとえば、outline と shape は同義語ですが、shape を記述用語にし、outline を非記述用語にすることができます。

Ngram シソーラスでは、記述用語と非記述用語は区別されません。

## 関係

関係は、2 つの用語間の関連を表現するものです。関係には、以下の特性があります。

- 関係の深さ は、関係が波及するレベルの数です。これは、検索構文内で THESDEPTH キーワードを使用することによって指定されます。照会でシソーラスの拡張を使用する方法については、221ページの『第12章 検索引き数の構文』を参照してください。
- 関係の方向性 は、その関係が1つの用語と他の用語間で等しく真である(両方向)か、一方向にのみ真である(単一方向)かを示します。

シソーラスの拡張では、シソーラス内に定義されているどの関係も使用することができます。また、その拡張の深さも指定できます。これが、ソース用語からターゲット用語への変換の最大数です。ただし、深さが増加すると、用語セットが指数関数的に増えることに注意してください。

次の例は、深さが増加されたときに、新たに追加された用語を示しています。

health

health service, paramedical, medicine, illness

allergology, virology, veterinary medicine, toxicology, surgery, stomatology, rheumatology, radiotherapy, psychiatry, preventive medicine, pathology, odontology, nutrition, nuclear medicine, neurology, nephrology, medical check up, industrial medicine, hematology, general medicine, epidemiology, clinical trial, cardiology, cancerology

## シソーラスの概念

### DB2 テキスト・エクステンダー・シソーラスの関係

DB2 テキスト・エクステンダー・シソーラスに備わっている関係には、以下のものがあります。

- 結合
- 同義
- 階層
- その他

DB2 テキスト・エクステンダー・シソーラスでは、事前定義の関係はありません。各関係には、**BROADER TERM** などの名前を付けることができ、これは **BT** のような簡略形にすることができます。シソーラスの設計で一般に使用される関係は以下のとおりです。

- **BT** または **BROADER TERM**
- **NT** または **NARROWER TERM**
- **RT** または **RELATED TERM**
- **SYN** または **SYNONYM**
- **USE**
- **UF** または **USE FOR**

**結合:** 結合関係は記述用語間の両方向関係で、任意の深さに拡張されます。この関係は、同等でも階層的でもないが、意味的に関連した 2 つの用語を結び付けて、その連結が、索引付けまたは検索で使用される追加の用語を想起できるようにします。

結合関係は一般に **RT** (関連用語) として示されます。

例:

```
dog RT security
pet RT veterinarian
```

**同義:** DB2 テキスト・エクステンダー・シソーラスにおけると同様に、記述用語と非記述用語の区別があるときには、同義関係は、同じまたは類似した意味をもつ 2 つの用語間の単一方向関係です。同義語クラスでは、いずれか 1 つの用語が記述用語として指定されます。それ以外の用語は、非記述用語と呼ばれます。記述用語と非記述用語との区別がないときの同義関係の定義については、279ページの『Ngram シソーラスの関係』を参照してください。

一般に指定される **USE** は、特定の非記述用語からその記述用語を示します。

一般に指定される **USE FOR** は、記述用語から各非記述用語を示します。例:

```
feline USE cat
```

lawyer UF advocate

**階層:** 階層関係は、記述用語間の単一方向関係で、ある用語が他の用語より特定化されている、すなわち汎用性がなくなっていることを表します。この差異によって、ある用語が 1 つのクラスを表し、それに従属する用語はそのメンバー部分を示すように用語が階層的に表されます。たとえば、「mouse」という用語は、「rodent」クラスに属します。

BROADER TERM と NARROWER TERM は階層関係を示します。

例:

car NT limousine

equine BT horse

**その他:** その他 のタイプの関係はもっとも汎用性があります。これは、他のカテゴリーに容易に含められない関連を表します。その他 のタイプの関係には、単一方向も両方向もあり、深さに制限はありません。また、記述用語と非記述用語との間に複数の関係があることもあります。

このような関係は、他の用語との適切な関係が決まるまでの間、シソーラスの新しい用語によく使用されます。

もちろん、記述子間の同義関係に結合 の関係タイプを使用するか、または任意の用語間の同義関係にその他 の関係タイプを使用して、独自の両方向同義関係を定義することもできます。

## Ngram シソーラスの関係

Ngram シソーラスは、次の 2 つのタイプをサポートします。

- 結合
- 同義

2 つ存在する事前定義関係のそれぞれは、これら 2 つのタイプのうちの 1 つに基づいています。結合 タイプに基づく独自の関係を定義することもできます。詳しくは、285ページの『Ngram シソーラスの作成』を参照してください。

### 結合

結合関係は、概念は異なっていながらも相互に関連している 2 つの用語の間の両方向関係です。事前定義関係 *RELATED\_TO* およびすべてのユーザー定義の関係は、この関係タイプに基づいています。

例:

## シソーラスの概念

tennis RELATED\_TO racket  
German RELATED\_TO sausage

### 同義

同義関係は、同じまたは類似した意味をもち、互いに代替語として使用できる、2つの用語間の両方向関係です。この関係は、たとえば、用語とその省略語に使用することができます。事前定義関係 `SYNONYM_OF` だけが、このタイプに基づいています。

例:

spot SYNONYM\_OF stain  
US SYNONYM\_OF United States

## シソーラスの作成

285ページの『Ngram シソーラスの作成』も参照してください。

インストール・パスの `samples` ディレクトリーには、英語シソーラス・コンパイル入力ファイルのサンプル `desthes.sgm` が保管されています。OS/2 および Windows システムの辞書ディレクトリーは、次のディレクトリーです。

`drive:¥dmb¥db2tx¥samples`

AIX、HP-UX および SUN-Solaris システムの場合は、次のディレクトリーです。

`DB2TX_INSTOWNER /db2tx/samples`

次の辞書ディレクトリーには、このシソーラスのコンパイル済みバージョンとその SGML 入力ファイルが保管されています。

`drive:¥dmb¥db2tx¥dict`

または

`DB2TX_INSTOWNER /db2tx/dicts`

このシソーラスに属するファイルは、`desthes.th1`、`desthes.th2`、...、および `desthes.th6` と呼ばれます。

シソーラスを作成するためには、まず、ファイルの内容を定義します。それぞれのシソーラスの定義には、分かりやすいディレクトリーを使用することをお勧めします。ファイルの拡張子には、`th1` ~ `th6` (これらはシソーラス辞書に使用される) 以外のものを付けることができます。Ngram シソーラスと同じディレクトリーを使用する場合には、285ページの『Ngram シソーラスの作成』にある、さらに限定されたファイル拡張子を使用してください。

それから、次のコマンドを実行して、ファイルをコンパイルします。

```
txthesc -f filename -c ccsid
```

*filename* に使用できる文字は、 a ~ z、A ~ Z、および 0 ~ 9 だけです。

現時点では、CCSID 850 だけがサポートされています。

txthesc により、定義ファイルと同じディレクトリーに、拡張子のない (th1 ~ th6 の拡張子も付いていない) *filename* という名前のシソーラス・ファイルが作成されます。すでに同じ名前のシソーラスがある場合には、警告なしでそれが上書きされます。

照会でシソーラスを使用する方法については、221ページの『第12章 検索引き数の構文』を参照してください。

シソーラスの内容は、標準汎用印刷指定言語 (SGML) を使用して指定します。次の図は、シソーラスを作成する際の構文規則を示しています。

```
▶<thesaurus><header><thname>thesaurus-name</thname>
```

```
<rldef>relation-definition</rldef></header>
```

```
thesaurus-entry</thesaurus>
```

#### relation-definition:

```
<rls><rlname>relation-name</rlname>
```

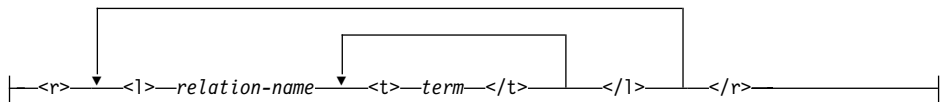
```
<rltype>ASSOCIATIVE</rltype></rls>
      |
      |SYNONYMOUS
      |
      |HIERARCHICAL
      |
      |OTHER
```

#### thesaurus-entry:

```
<en>unique-number, 1 <t>term</t> 2 related-terms </en>
```

#### related-terms:

## シソーラスの概念



`relation-name` に使用できる文字は、 a ~ z、A ~ Z、および 0 ~ 9 だけです。

283ページの図19 は、276ページの図18 に示されているシソーラスの SGML 定義を示しています。



```

<thesaurus>
<header>
<thname>thesc example thesaurus</thname>
<rldef>

<rls>
<rlname>Related Term</rlname>
<rltype>associative</rltype>
</rls>

<rls>
<rlname>Narrower Term</rlname>
<rltype>hierarchical</rltype>
</rls>

<rls>
<rlname>Instance</rlname>
<rltype>hierarchical</rltype>
</rls>

<rls>
<rlname>Synonym</rlname>
<rltype>synonymous</rltype>
</rls>
</rldef>
</header>

<en> 2, 1
<t>database management system</t>
<r>
  <l>Narrower Term
  <t>oo database management system</t>
  <t>relational database management system</t>
  </l>

  <l>Synonym
  <t>DBMS</t>
  </l>

  <l>Related Term
  <t>document management system</t>
  </l>

  <l>Instance
  <t>database</t>
  </l>
</r>
</en>

```

図 19. 簡単なシソーラスの定義 (1/2)

## シソーラスの概念

```
<en> 5, 1
<t> relational database management system </t>
<r>
  <l>Narrower Term
  <t>object relational database management system</t>
  </l>
</r>
</en>

<en> 3, 1
<t>object relational database management system</t>
<r>
  <l>Instance
  <t>DB2 Universal Database</t>
  </l>
</r>
</en>

<en> 6, 1
<t>object oriented database management system</t>
<r>
  <l>Narrower Term
  <t>object relational database management system</t>
  </l>
</r>
</en>

<en> 4, 1
<t>document management system</t>
<r>
  <l>Synonym
  <t>library</t>
  </l>
</r>
</en>

<en> 9, 1
<t>library</t>
</en>

<en> 10, 1
<t>DB2 Universal Database</t>
</en>

<en> 11, 1
<t>database</t>
</en>
</thesaurus>
```

図 19. 簡単なシソーラスの定義 (2/2)

## Ngram シソーラスの作成

インストール・パスの辞書ディレクトリーには、英語の Ngram シソーラス・コンパイラ入力ファイルのサンプル `desnthes.def` が保管されています。OS/2 および Windows システムの辞書ディレクトリーは、次のディレクトリーです。

```
drive:¥dmb¥db2tx¥dict
```

AIX、HP-UX および SUN-Solaris システムの場合は、次のディレクトリーです。

```
DB2TX_INSTOWNER /db2tx/dicts
```

このディレクトリーには、このサンプル・シソーラスのコンパイル済みバージョンも保管されています。このシソーラスに属するファイルは、`desnthes.<拡張子>` という名前になっています。拡張子は次のとおりです (n は数字)。

- 辞書ファイルの場合: `wdf`、`wdv`、`grf`、`grv`、`MEY`、`ROS`、`NEY`、`SOS`、`lkn`
- 一時ファイルの場合: `wnf`、`wnv`、`gnf`、`gnv`、`M!1`、`M!2`、`N!1`、`N!2`、`R!1`、`R!2`、`S!1`、`S!2`、`Mnn`、`Nnn`、`Rnn`、`Snn`、`$00`、`$01`、`$10`、`$11`、`$20`、および `$21`

Ngram シソーラスを作成するには、まず定義ファイルにその内容を定義します。同じディレクトリーに複数のシソーラスを入れることはできますが、それぞれのシソーラスに別個のディレクトリーを使用することをお勧めします。拡張子を除いたファイル名の長さは、8 文字以下でなければなりません。拡張子は任意指定ですが、3 文字までで、上記以外のものでなければなりません。

他の DB2 テキスト・エクステンダー・シソーラス用に同じディレクトリーを使用する場合、280ページの『シソーラスの作成』にリストされた拡張子を使用してはなりません。

それから、次のコマンドを実行して、ファイルをコンパイルします。

```
txthesn -f definition-file-name -ccsid code-page
```

以下に、Ngram シソーラスでサポートされているコード・ページをリストします。

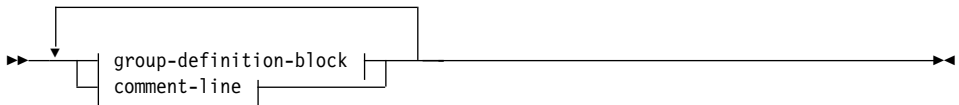
932	AIX、OS/2	日本語
942	OS/2	日本語
943	OS/2、Windows	日本語
949	OS/2	韓国語

## シソーラスの概念

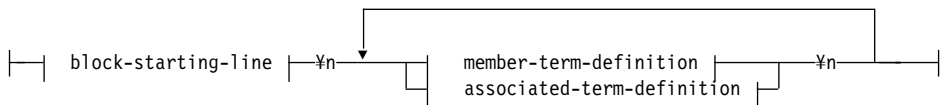
950	AIX、HP-UX、OS/2、SUN-Solaris、Windows	中国語 (繁体字)
970	AIX、HP-UX、SUN-Solaris	韓国語
1381	OS/2、Windows	中国語 (簡体字)
1363	Windows	韓国語
1383	AIX、HP-UX、SUN-Solaris	中国語 (簡体字)
850	AIX、OS/2	ラテン -1
1252	Windows	ラテン -1

txthesn により、上記の拡張子を持ち、*definition-file-name* と同じ名前を持つシソーラス・ファイルが作成されます。このファイルは、定義ファイルと同じディレクトリーに作成されます。すでに同じ名前のシソーラスがある場合には、警告なしでそれが上書きされます。

シソーラスの内容の指定には、次の構文図を使用します。



### group-definition-block:



### block-starting-line:

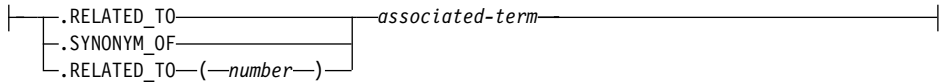
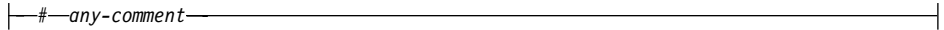


### member-relation:



### member-term-definition:



**associated-term-definition:****comment-line:**

それぞれのメンバー用語は単一行に書き込む必要があります。それぞれの関連用語の前には、関係名を付ける必要があります。メンバー用語が互いに関連している場合には、メンバー関係を指定します。

メンバー用語および関連用語の長さは、164 文字までに制限されています。同一文字の単一バイト文字と 2 バイト文字は、同じとみなされます。大文字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ますが、単一バイト文字のピリオド (.) およびコロン (:) については、いずれか一方だけを使用できます。

ユーザー定義の関係はすべて、結合 タイプに基づいています。これらの関係は 1 ~ 128 の固有の数字で識別されます。

アプリケーションでそのシソーラス関係を表すのに関係名や数値ではなく記号名を使用する場合には、アプリケーション側でマッピングの管理を行う必要があります。たとえば、OPPOSITE\_OF 関係を RELATED\_TO(1) と定義した場合、アプリケーションはこの名前を内部の関係名 RELATED\_TO(1) にマップする必要があります。照会でシソーラスの拡張を使用する方法については、221 ページの『第12章 検索引き数の構文』を参照してください。

## 変更の要約

---

## 第16章 構成ファイル

この章では、構成ファイルについて説明します。構成ファイルは、`txicrt` コマンドを使用して DB2 テキスト・エクステンダー・インスタンスを作成する際や、`descrc1` コマンドを使用してクライアント・プロファイルを作成する際に、自動的に生成されます。構成ファイルはコード・ページ 819 で生成されません。

構成ファイルを編集してご使用のシステムを調整できますが、ファイルを編集する際には正しいコード・ページを使用しているか確認してください。セクション名とオプション名は、大文字と小文字を区別しません。コメント区切り文字としてセミコロンが使用されます。

オプションがブール値の場合は、値 `TRUE`、`YES`、`ON`、および `1` は大文字小文字にかかわらず真と見なされます。他の値はすべて偽と見なされます。

---

### クライアント構成ファイル

ファイル名      `DESCL.INI`

場所              `UNIX: DB2TX_INSTOWNERHOMEDIR/db2tx`

`Windows NT および Windows 2000:`  
`%DMBMPATH%¥instance¥%DB2INSTANCE%¥db2tx`

オプションを更新すると、次回 `StartSession` 関数を実行する際にそれらのオプションがアクティブになります。

セクション	オプション	省略時値	説明
[INSTANCE]	DESWORKCL		一時ファイル用に使用される作業ディレクトリーを指します。
	DESNLPSCl		資源ディレクトリーを指します。
	DESDDPATHSRV		XML 文書の DTD が保管されるディレクトリーを指します。

## クライアント構成ファイル

セクション	オプション	省略時値	説明
	DESDTDPATHCL		DTD ファイルを保管しなければならない場所のパスを示します。XML ファイルの場合に使用されます。
[BUFFER]	BUFFERSEGMENTSIZ	32 000	バッファリングに使用されるブロック・セグメントのサイズ (バイト単位)。
	BUFFERSEGMENTCOUNT	3	バッファが一時的にファイルにスワップされる前に使用されるセグメントの最大数。バッファ・セグメントは <b>BUFFERSEGMENTSIZ</b> によって定義されます。
[DOCUMENTFORMAT]	USEREXIT		サポートされない文書形式の処理に使用されるユーザー出口の名前。ユーザー出口が <b>PATH</b> ステートメントの一部であるディレクトリーに保管されている場合はファイル名を指定し、それ以外の場合は完全修飾ファイル名を指定してください。  形式変換用のユーザー出口に関する詳細は、37ページの『サポートされていない文書形式の使用』を参照してください。
	FORMATRECOGNITION	TRUE	文書形式の形式認識を起動します。  TRUE: 形式認識がオン FALSE: 形式認識がオフ



セクション	オプション	省略時値	説明
	UseExitForAllFormats	FALSE	<p>deslsdef.h にリストされていない文書形式を処理するユーザー出口が、いつ呼び出されるかを判別します。 USEREXIT の値を設定しなければなりません。</p> <p>TRUE: ユーザー出口は常時呼び出されます。この値を設定しないと、 FORMATRECOGNITION は無視されます。</p> <p>FALSE: EHW_USER_FORMATS の値を上回るすべての文書形式の場合に、ユーザー出口を呼び出します。</p>

## サーバー構成ファイル

ファイル名      DESSRV.INI

場所              UNIX: DB2TX\_INSTOWNERHOMEDIR/db2tx/TXINSnnn

Windows NT および Windows 2000:

%DMBMMPATH%¥instance¥%DB2INSTANCE%¥db2tx¥txinsnnn。

このファイルは、サーバー・インスタンスあたり 1 つだけ存在します。

オプションを更新すると、次回サーバー・インスタンスを開始する際にそれらのオプションがアクティブになります。

表 20. サーバー構成ファイルのオプション

セクション	オプション	省略時値	説明
[INSTANCE]	DESWORKSRV		一時ファイル用に使用される作業ディレクトリーを指します。
	DESNLPSSRV		資源ディレクトリーを指します。

## サーバー構成ファイル

表 20. サーバー構成ファイルのオプション (続き)

セクション	オプション	省略時値	説明
[DAEMON]	MaxMtEntries	AIX、 Windows、 OS/390 の場 合は 30  Solaris の場 合は 10	一度に並列使用される索引の最大数。セマフォや共用メモリーなどの資源が足りない場合は、この数値を小さくしてください。使用可能な資源はプラットフォームに応じて異なるので、省略時値もプラットフォームに応じて異なります。
	MaxIndexEntries	1000	使用される索引の最大数。共用メモリーなどの資源が足りない場合は、この数値を小さくしてください。
[BUFFER]	BUFFERSEGMENTSIZ	32 000	バッファリングに使用されるブロック・セグメントのサイズ (バイト単位)。これは EhwUpdate で使用されます。
	BUFFERSEGMENTCOUNT	3	索引の更新処理中の、バッファが一時的にファイルにスワップされる前に使用されるセグメントの最大数。文書のコレクションに大きな文書が含まれている場合は、この数値を大きくしてください。
	BUFFERSORTSIZE	20 000 000	一時作業ファイルのソートに使用されるバッファのサイズ (バイト単位)。

表 20. サーバー構成ファイルのオプション (続き)

セクション	オプション	省略時値	説明
[DOCUMENTFORMAT]	USEREXIT		<p>deslsdef.h にリストされていない文書形式の処理に使用されるユーザー出口の名前。ユーザー出口が PATH ステートメントの一部であるディレクトリーに保管されている場合はファイル名を指定し、それ以外の場合は完全修飾ファイル名を指定してください。</p> <p>形式変換用のユーザー出口に関する詳細は、37ページの『サポートされていない文書形式の使用』を参照してください。</p>
	FORMATRECOGNITION	TRUE	<p>deslsdef.h にリストされていない文書形式の形式認識を起動します。</p> <p>TRUE: 形式認識がオン</p> <p>FALSE: 形式認識がオフ</p>
	UseExitForAllFormats	FALSE	<p>deslsdef.h にリストされていない文書形式を処理するユーザー出口が、いつ呼び出されるかを判別します。</p> <p>USEREXIT の値を設定しなければなりません。</p> <p>TRUE: ユーザー出口は常時呼び出されます。この値を設定しないと、FORMATRECOGNITION は無視されます。</p> <p>FALSE: EHW_USER_FORMATS の値を上回るすべての文書形式の場合に、ユーザー出口を呼び出します。</p>

## サーバー構成ファイル

表 20. サーバー構成ファイルのオプション (続き)

セクション	オプション	省略時値	説明
[LINGPREC] 基本タイプが基本言語または一致語句であるすべての索引	UPDATETHRESHOLD	4 000 000	索引更新処理は、複数の更新処理と再編成処理に内部的に分割されます。この値は、1 つの更新ステップで収集される語の数を指定します。
	UPDATESLICE	1	内部再編成処理が開始される前に行われる更新処理の数。更新処理は UPDATETHRESHOLD によって定義されます。
[NGRAM] DBCS がサポートされているすべての索引	UPDATETHRESHOLD	10 000 000	1 つの更新処理中に索引に追加される文書の合計サイズ (バイト単位)。しきい値を超えると、再編成処理が自動的に開始されます。
	UPDATESLICE	10 000	2 次索引内の文書の最大数。この数値は、更新処理が終わるたびにチェックされます。文書の数がこの値より大きいと、再編成処理が自動的に開始されます。

## テキスト分析構成ファイル

ファイル名 DESTAF.TCF

場所 UNIX: DB2TX\_INSTOWNERHOMEDIR/db2tx/TXINSnnn

Windows NT および Windows 2000:

%DMBMPATH%\¥instance¥%DB2INSTANCE%\¥db2tx¥txinsnnn。

このファイルは、サーバー・インスタンス当たり 1 つだけ存在します。

テキスト分析構成ファイルには、 ngram 索引以外の索引タイプごとに更新処理と照会処理のセクションが含まれています。変更を加えるのは、下記のオプションだけにしてください。その他の項目に変更を加えると、更新処理や照会処理の最中にエラーが起きる可能性があります。

## 共通オプション

以下のオプションは、以下のように適用されます。

- [TAF\_Application:UPDATE:PREC]), PRECISE 索引タイプ
- [TAF\_Application:UPDATE:LING]), LINGUISTIC 索引タイプ
- [TAF\_Application:UPDATE:NORM]), PRECISE NORMALIZED 索引タイプ
- [TAF\_Application:QUERY:PREC])
- [TAF\_Application:QUERY:LING])
- [TAF\_Application:QUERY:NORM])

オプション	省略時値	説明
FilterSet	None (必須)	停止ワードのフィルター処理の規則を含む構成ファイル中のセクションの名前。たとえば、StwUseAllUpCs や StwIgnAllUpCs など。
Normalize	No	索引に対して正規化を実行するかどうかを指定します。  これを可に設定すると、正規化により、すべての文字が小文字に変換され、特殊アクセント文字が基本文字に置き換えられます。  これを不可に設定すると、NormalizeUmlauts と ConvertAllUpperToLower の項目は無視されます。
NormalizeUmlauts	Yes	ウムラウトを置き換える (たとえば、ä を ae に置き換える) かどうかを指定します。これはドイツ語、デンマーク語、スウェーデン語、またはノルウェー語の文書だけに適用されます。
ConvertAllUpperToLower	No	英大文字の語をすべて小文字に変換するかどうかを指定します。

### [TAF\_Filter:STWUSEALLUPCS] [TAF\_Filter:STWIGNALLUPCS]

これらのセクション中のオプションは、停止ワードのフィルターの集合を記述します。 [TAF\_Filter:STWUSEALLUPCS] セクションは、英大文字の語をすべて大文字小文字混合文字の語として扱い、 [TAF\_Filter:STWIGNALLUPCS] セクションは、英大文字の語の特殊な処理方法を定義します。

オプション	省略時値	説明
FilterType	TAF_FILTER_STW	停止ワードのフィルター処理の規則を含む構成ファイル中のセクション。

## テキスト分析構成ファイル

オプション	省略時値	説明
MinTokenLength	3	トークンに必要な文字の最小数。トークンの長さがこの値より小さいと、索引付けされません。これは $\geq 1$ の整数値です。
MinUpperCaseTokenLength	2	すべて英大文字のトークンに必要な文字の最小数。トークンの長さがこの値より小さいと、索引付けされません。これは $\geq 1$ の整数値です。
UseAlternateTerritories	No	言語固有の辞書を検出できない場合に、停止ワード処理用の代替辞書を使用するかどうかを指定します。  これを可に設定すると、言語が一致する最初の停止ワード辞書が使用されます。たとえば、現行の言語が en-gb だが en-gb.tsw 停止ワード・ファイルがない場合、en-us.tsw 停止ワード・ファイルが使用されます。  これを不可に設定すると、停止ワードのフィルター処理は行われません。

### [TAF\_AnnotationType:TAF\_ThesaurusEntry ]

このセクションは、シソーラス拡張が必要な照会の省略時値を設定します。

キーワード	省略時値	説明
ThesBaseName	desthes	照会拡張が必要な照会で使用されるシソーラスの省略時名を設定します。この名前は文字ストリングです。
Depth	1	シソーラス拡張が必要な照会で使用されるシソーラスを走査する省略時パスを設定します。これは $\geq 1$ の整数値です。

### [TAF\_Plugin:TAF\_POE ]

テキスト検索エンジンの開始時にプリロードされる言語環境の省略時値を変更するには、PreloadLanguages オプションで言語を定義できます。

オプション	省略時値	説明
PreloadLanguages	EN-US	始動時にプリロードされる言語固有の資源の集合を指定します。たとえば、イタリア語の文書に対してアプリケーションが実行される場合は、イタリア語の資源がロードされます。このフィールドの値の形式は XX-YY (XX は言語、YY はテリトリ) です。





---

## 第17章 戻りコード

この章では、DB2 テキスト・エクステンダー API が関数呼び出しに対応して戻すコードを記載します。コードはアルファベット順に並べられています。

DB2 テキスト・エクステンダー API 呼び出しはすべて、数値の戻りコードを C 関数値として戻します。戻りコードは、DB2 テキスト・エクステンダーが提供する組み込みファイル DES\_EXT.H で定義されています。

DB2 テキスト・エクステンダー API はエラー状態を代行受信し、そのエラー状態を戻りコードとともに報告します。

DB2 テキスト・エクステンダー API 関数を呼び出すアプリケーションでは、必ず初めに戻りコードをチェックしてから、他の出力パラメーターを処理するようにしてください。それぞれの呼び出しで戻される可能性のある戻りコードが、235ページの『第13章 検索およびブラウズのための API 関数』にパラメーターとともに列挙されています。

ある場合には、DB2 テキスト・エクステンダーが代行受信できない API サービスにおいて、使用されなくなったセッション・ポインターなどを誤って入力したために異常終了することがあります。

---

### RC\_ALLOCATION\_ERROR

**説明:** 内部使用のための記憶域割り当てができません。

**ユーザーの処置:** 使用可能なメモリーが十分にあるか確認してください。

---

### RC\_FILE\_IO\_PROBLEM

**説明:** DB2 テキスト・エクステンダーがファイルの読み取りまたは書き込みを行えませんでした。

**ユーザーの処置:** サーバーに使用可能なディスク・スペースとメモリーが十分にあるかチェックしてください。環境変数とテキスト構成設定値が

正しく設定されているかをチェックしてください。

---

### RC\_INVALID\_BROWSE\_INFO

**説明:** DesGetSearchResultTable または DesGetBrowseInfo によって戻され、かつ DesStartBrowseSession の入力に使用されるブラウズ情報が無効です。

**ユーザーの処置:** プログラミング・エラーによってブラウズ情報が一時変更されていないかチェックしてください。

## 戻りコード

---

### RC\_INVALID\_BROWSE\_OPTION

**説明:** DesGetSearchResultTable 内のブラウズ・オプションが無効です。

**ユーザーの処置:** オプションが BROWSE または NO\_BROWSE であることを確認してください。

---

### RC\_INVALID\_MATCH\_OPTION

**説明:** DesOpenDocument で使用されている突き合わせオプションが無効です。

**ユーザーの処置:** オプションが FAST または EXTENDED であるかチェックしてください。

---

### RC\_INVALID\_PARAMETER

**説明:** 入力パラメーターの 1 つに誤りがあります。

**ユーザーの処置:** DB2 テキスト・エクステンダーによって戻されるエラー・メッセージを読み、原因を判別してください。

---

### RC\_INVALID\_SEARCH\_OPTION

**説明:** DesGetSearchResultTable 内の検索オプションが無効です。

**ユーザーの処置:** オプションが DES\_TEXTHANDLEONLY、DES\_RANK、DES\_MATCH、または DES\_RANKANDMATCH であることを確認してください。

---

### RC\_INVALID\_SESSION

**説明:** 現行のサービス呼び出しに指定されているセッション・ポインターに誤りがあるか、または使用されなくなっています。

**ユーザーの処置:** エラーの原因究明に役立つ情報をすべて保管した後、アプリケーションを終了してください。

---

### RC\_NO\_BROWSE\_INFO

**説明:** DB2 テキスト・エクステンダーはブラウズ情報を何も戻しませんでした。これは、検索引き数が何も検索結果を得られなかったためです。これはエラーではありません。

**ユーザーの処置:** 何もする必要はありません。

---

### RC\_PARSER\_INVALID\_ESCAPE\_CHARACTER

**説明:** 検索基準に誤ったエスケープ文字が含まれています。このエラーは、ブランクをエスケープ文字として使用した場合、または検索基準で 1 つの語あるいは句について 1 つ以上のエスケープ文字を指定した場合などに報告されます。例: ESCAPE " " または ESCAPE "\$"。

**ユーザーの処置:** 検索引き数の構文をチェックし、再試行してください。

---

### RC\_PARSER\_INVALID\_USE\_OF\_ESCAPE\_CHAR

**説明:** 検索基準内のエスケープ文字の構文が解釈できません。

**ユーザーの処置:** エスケープ文字の構文をチェックしてください。たとえば、\$ が指定されたエスケープ文字である場合、語句に含めることができるのは、\$\$、\$\_ または % だけです。ここで、\_ および % は 2 つのマスキング記号を表しています。

---

### RC\_PARSER\_SYNTAX\_ERROR

**説明:** 検索基準構文は解釈することができません。

**ユーザーの処置:** 221ページの『第12章 検索引き数の構文』を参照し、検索引き数の構文をチェックしてください。

---

### RC\_RESULT\_TABLE\_NOT\_EXIST

**説明:** 存在しない表内に検索結果を保管しようとしています。

**ユーザーの処置:** 252ページの図17 に示すように結果表を作成してください。

---

### RC\_SE\_BROWSER\_TIME\_OUT

**説明:** ブラウズ・プロセスが開始されましたが、許容時間内に応答がありませんでした。そこで、DB2 テキスト・エクステンダーは保留状態のプロセスを取り消しました。

このエラーは、システムに十分な記憶域がないか負荷がかかり過ぎる場合に起こる可能性があります。

**ユーザーの処置:** DesEndBrowseSession を呼び出してブラウズ・セッションを終了させ、DesFreeBrowseInfo で割り振り済み記憶域を解放し、それから再試行してください。

---

### RC\_SE\_CAPACITY\_LIMIT\_EXCEEDED

**説明:** 要求された機能は処理できません。メモリーまたはディスク・スペースが不足しています。

**ユーザーの処置:** プログラムを終了し、システムの資源を調べてください。

---

### RC\_SE\_COMMUNICATION\_PROBLEM

**説明:** DB2 テキスト・エクステンダー・サーバーとの通信に失敗しました。記憶域が足りなかったか DB2 テキスト・エクステンダーのインスタ

ールに誤りがあったためにエラーが起こったと考えられます。

**ユーザーの処置:** エラー検出に役立つ情報をすべて保管し、アプリケーションを終了させてください。

---

### RC\_SE\_CONFLICT\_WITH\_INDEX\_TYPE

**説明:** 照会の検索項目の基本語形指定が、索引のタイプに対応していません。たとえば、PRECISE FORM OF を基本語形索引で使用することはできません。省略時の基本語形指定は、229ページの表8 に示すように使用されます。

**ユーザーの処置:** アプリケーションを変更して、その索引のタイプと対立する照会オプションを指定しないようにしてください。

---

### RC\_SE\_DICTIONARY\_NOT\_FOUND

**説明:** DB2 テキスト・エクステンダー基本語形サービスが辞書ファイルを検出できません。照会には言語的なサポートなしに処理されます。指定の言語コード (複数の場合もある) に対応する辞書ファイルが、あるべきパスにありません。

**ユーザーの処置:** API 呼び出しを継続してください。UNIX の場合、必要な辞書がパス {DB2TX\_INSTOWNERHOMEDIR}/db2tx/dicts に入っているかをチェックしてください。OS/2 の場合は、必要な辞書がテキスト構成設定 DB2TX\_DATA で指定されたパス内にあるかをチェックしてください。必要であれば、必要な辞書をインストールしてください。

---

### RC\_SE\_DOCMOD\_READ\_PROBLEM

**説明:** DB2 テキスト・エクステンダー・インスタンスの作成時に、文書モデル・ファイル desmodel.ini がインスタンス・ディレクトリーに保管されます。索引の作成時に、

## 戻りコード

desmodel.ini ファイルは索引ディレクトリー IXnnnnnn にも保管されます。この文書モデル・ファイルを読み取れませんでした。

**ユーザーの処置:** 文書モデル・ファイルが存在し、正しいディレクトリーに入っていることをチェックしてください。

---

### RC\_SE\_DOCUMENT\_NOT\_ACCESSIBLE

**説明:** 要求されたテキスト文書が検出されましたが、現在はアクセスできません。

**ユーザーの処置:** その文書が、他のタスクやユーザーによって独占的に使用されていないかチェックしてください。

---

### RC\_SE\_DOCUMENT\_NOT\_FOUND

**説明:** 要求されたテキスト文書が見つかりませんでした。その原因としては、テキスト文書が記憶域から削除されてしまっているが、DB2 テキスト・エクステンダーの索引からはまだ削除されていないことが一番考えられます。また、損傷を受けたハンドルによって識別された文書をブラウズしようとする場合にも起こることがあります。

**ユーザーの処置:** ほとんどの場合、この戻りコードは無視してもかまいません。この戻りコードは、次の索引更新時以降は表示されなくなります。

もし表示される場合は、アプリケーション・プログラムが検出ハンドルをブラウズ用に正常に渡しているかチェックしてください。

---

### RC\_SE\_EMPTY\_INDEX

**説明:** 検索要求によってアドレス指定されたハンドル列に対応する DB2 テキスト・エクステンダー索引が空です。すなわち、索引にテキスト文書が何も追加されなかったか、またはすべてのテキ

スト文書が索引から削除されたかのどちらかです。

これは、テキスト列を使用可能にしたものの、列内の文書にまだ索引が付けられていない場合に起こる可能性があります。つまり、後で (周期的な索引付け設定により決められたときに) 索引を作成することを ENABLE TEXT COLUMN コマンド内で指定していたというわけです。

これは、テキスト表を使用可能にしてすべてのテキスト列に関する空の共通索引を作成してあるが、テキスト列はどれも使用可能になっていなかった場合にも起こることがあります。

**ユーザーの処置:** ENABLE TEXT TABLE ですべてのテキスト列に関する空の共通索引を作成した場合は、検索するテキストを含むテキスト列の少なくとも 1 つに ENABLE TEXT COLUMN を実行してください。このコマンドでは、索引を即時に作成するか、または定期的な索引付け設定によって指定された時点で作成するかを決めることができます。

GET INDEX STATUS を実行して、索引が正常に作成されたことをチェックしてください。

---

### RC\_SE\_EMPTY\_QUERY

**説明:** 指定の検索基準は、DB2 テキスト・エクステンダーによって言語学的に分析されて処理されました。プログラミング・エラーのために照会に検索項目が含まれなくなったか、またはすべての検索項目が、照会から除去された停止ワード (DB2 テキスト・エクステンダーによる索引付けをされていない語) であるかのどちらかです。結果は検索項目ではありませんでした。

**ユーザーの処置:** 照会を繰り返してください。それでも問題が残る場合には、プログラミング・エラーをチェックしてください。

---

**RC\_SE\_END\_OF\_INFORMATION**

**説明:** これはエラーではありません。文書の終わりに到達しました。DesGetMatches に関する情報をこれ以上得ることはできません。

**ユーザーの処置:** この戻りコードを使用して、DesGetMatches による文書の反復処理を終了させてください。

---

**RC\_SE\_FUNCTION\_DISABLED**

**説明:** 要求された関数が、管理者により禁止されている DB2 テキスト・エクステンダー関数を呼び出しました。

**ユーザーの処置:** 管理担当者に援助を依頼してください。DB2 テキスト・エクステンダーをいったん停止して再始動する必要があるかもしれません (txstop/txstart)。

---

**RC\_SE\_FUNCTION\_IN\_ERROR**

**説明:** DB2 テキスト・エクステンダー・サーバー上のエラーのために、要求された関数がロックされています。API 呼び出しは処理できません。

**ユーザーの処置:** 索引状況をチェックしてください。索引ディレクトリーで使用できるスペースをチェックしてください。索引状況をリセットし、コマンドを再試行してください。

---

**RC\_SE\_INCORRECT\_HANDLE**

**説明:** 入力パラメーターに指定されたハンドル (たとえばブラウザ・セッション・ハンドル) が無効です。このハンドルは、直前の呼び出しで戻されて現在は使用されないものであると推定されます。

**ユーザーの処置:** エラーの原因究明に役立つ情報

をすべて保管した後、DesEndBrowseSession を呼び出してセッションを終了してください。

プログラミング・エラーによって誤ったハンドルが作成されていないかチェックしてください。

---

**RC\_SE\_INDEX\_DELETED**

**説明:** アクセスした DB2 テキスト・エクステンダーの索引が削除されています。

**ユーザーの処置:** DB2 テキスト・エクステンダー管理者に連絡し、索引を作成し直してください。

---

**RC\_SE\_INDEX\_NOT\_ACCESSIBLE**

**説明:** DB2 テキスト・エクステンダーの索引にアクセスできず、現行の呼び出しは処理できません。

**ユーザーの処置:** DB2 テキスト・エクステンダー管理者に連絡を取り、索引へのアクセスが可能かをチェックしてください。

---

**RC\_SE\_INDEX\_SUSPENDED**

**説明:** DB2 テキスト・エクステンダーが、他のセッションまたは現行のセッションを延期している索引に関する要求を受信しました。

**ユーザーの処置:** DB2 テキスト・エクステンダー管理者に連絡を取り、索引の状況をチェックしてください。

---

**RC\_SE\_INSTALLATION\_PROBLEM**

**説明:** DB2 テキスト・エクステンダーで、インストール上の問題が検出されました。

**ユーザーの処置:** 環境変数 DB2INSTANCE、DB2TX\_INSTOWNER、DB2TXINSTOWNERHOMEDIR の現行の設定を手

## 戻りコード

ェックしてください。 `descfgcl -d` および `descfgsv -d -i txinsnnn` を使用して、検索サービス構成をチェックしてください。

---

### RC\_SE\_IO\_PROBLEM

**説明:** サーバーがそのサーバーの索引ファイルの 1 つをオープンして読み取ろうとした際に、エラーが発生しました。この原因としては、以下のどちらかが考えられます。

管理者が意図せずに、DB2 テキスト・エクステンダー索引ファイルの削除などを行ってしまった場合

テキスト構成内の設定に誤りがある場合

**ユーザーの処置:** アプリケーションを終了してください。管理者と一緒に以下の点をチェックしてください。

- 現行の DB2 テキスト・エクステンダー索引の全ファイルが存在している。
- テキスト構成設定が正しい。

---

### RC\_SE\_LS\_FUNCTION\_FAILED

**説明:** ブラウズの目的でテキスト文書を検索するためにデータベースにアクセスした関数が、失敗しました。ユーザーがデータベースにアクセスできなくなったか、またはユーザーにテキスト・テーブルに関する権限が与えられていません。

**ユーザーの処置:** ユーザー ID などの関数への入力が正しいかチェックしてください。データベースがアクセス可能か、またこのタスクに関する権限がユーザーに与えられているかチェックしてください。

---

### RC\_SE\_LS\_NOT\_EXECUTABLE

**説明:** ブラウズの目的でデータベースにアクセスしてテキスト文書を検索しようとしている関数を実行できません。

**ユーザーの処置:** DB2 テキスト・エクステンダーが正常にインストールされているかチェックしてください。問題が解決できない場合は、最寄りの IBM 担当員に連絡してください。

---

### RC\_SE\_MAX\_OUTPUT\_SIZE\_EXCEEDED

**説明:** 異常に多くの一致が検出されました。ブラウズ情報のサイズが、処理可能な最大サイズを超えています。要求は処理できません。

**ユーザーの処置:** 照会をより特定なものに限って行うか、またはシステム・メモリーが拡張可能か確認してください。

---

### RC\_SE\_MAX\_NUMBER\_OF\_BUSY\_INDEXES

**説明:** 要求された関数の実行が検索サービスによって阻止されました。現在活動状態にある索引が最大数に達しているためです。

**ユーザーの処置:** 少し時間がたってから関数呼び出しを再発行してください。通常、問題は一時的なものです。

---

### RC\_SE\_NO\_DATA

**説明:** これはエラーではありません。検索基準に一致するテキスト文書がありません。ブラウズ情報を要求しても、戻される情報はありません。ブラウズ情報用に記憶域が割り当てられていません。

**ユーザーの処置:** 何もする必要はありません。

---

### RC\_SE\_NOT\_ENOUGH\_MEMORY

**説明:** クライアントまたはサーバー・システム上に十分な記憶域がありません。現行の要求は処理できません。

**ユーザーの処置:** 記憶域を解放し、アプリケーション

ョンを終了してください。

---

### RC\_SE\_PROCESSING\_LIMIT\_EXCEEDED

**説明:** 現行の検索要求が、クライアント / サーバー環境用に指定された最大結果サイズまたは最大処理時間を超えています。要求は取り消されました。

**ユーザーの処置:** 検索要求をより特定の範囲に制限してください。最大処理時間の延長を検討してください。

---

### RC\_SE\_QUERY\_TOO\_COMPLEX

**説明:** 指定された照会が複雑過ぎます。

**ユーザーの処置:** アプリケーションを変更して、マスキング文字および同義語を過度に使用しないようにしてください。

マスキング記号または SYNONYM オプションを使い過ぎると、照会の範囲が、DB2 テキスト・エクステンダーで管理できないサイズにまで広がってしまうことがあります。

---

### RC\_SE\_REQUEST\_IN\_PROGRESS

**説明:** DB2 テキスト・エクステンダー・ブラウザ API サービスが呼び出されましたが、同一セッションで別のブラウザ API 要求が活動状態でした。

**ユーザーの処置:** DesEndBrowseSession を呼び出してセッションを終了し、DesFreeBrowseInfo で記憶域を解放してください。

DB2 テキスト・エクステンダー・ブラウザ API は、同一ブラウザ・セッションに対する並行アクセスをサポートしていません。

同一プロセスで並行実行中のアプリケーションはすべて、そのアプリケーション自体のブラウザ・セッションを処理するようにしてください。

---

### RC\_SE\_SERVER\_BUSY

**説明:** DB2 テキスト・エクステンダー・クライアントは、要求された DB2 テキスト・エクステンダー・サーバーとのセッションを現在確立できません。または DB2 テキスト・エクステンダー・サーバー通信リンクに割り込みが行われて確立し直すことができません。

DB2 テキスト・エクステンダー・サーバーは正常に開始されましたが、並列サーバー処理の数が最大値に達しました。

**ユーザーの処置:** これが一時的な問題で済まない場合は、サーバー上で通信の構成を変更してください。

---

### RC\_SE\_SERVER\_CONNECTION\_LOST

**説明:** クライアントとサーバーとの間の通信に割り込みが行われ、確立し直すことができません。

管理者が DB2 テキスト・エクステンダー・サーバー・タスクを停止したか、またはサーバー・ワークステーションが遮断した可能性があります。

**ユーザーの処置:** これらの条件のどちらかが発生していないかチェックし、訂正を行ってください。

---

### RC\_SE\_SERVER\_NOT\_AVAILABLE

**説明:** DB2 テキスト・エクステンダー API サービスは、要求された DB2 テキスト・エクステンダー・サーバーとのセッションを確立できませんでした。

DB2 テキスト・エクステンダー・サーバーが開始されていない可能性があります。

**ユーザーの処置:** DB2 テキスト・エクステンダー・サーバーが正常に開始されているかチェックしてください。エラーが解決しない場合は、インストールに問題があります。

## 戻りコード

---

### RC\_SE\_STOPWORD\_IGNORED

**説明:** この情報コードは、指定された照会に停止ワードだけから成る検索項目が少なくとも 1 つ含まれていた場合に戻されます。検索項目は照会の処理の間無視されました。

**ユーザーの処置:** API 呼び出しを継続してかまいません。DB2 テキスト・エクステンダー照会では停止ワードを使用しないでください。

---

### RC\_SE\_UNEXPECTED\_ERROR

**説明:** DB2 テキスト・エクステンダーのインストール方法に誤りがあったことが原因と思われるエラーが発生しました。

**ユーザーの処置:** エラーの原因究明に役立つと思われる情報を保管し、アプリケーションを終了させてください。

---

### RC\_SE\_UNKNOWN\_INDEX\_NAME

**説明:** テキスト列に関連したテキスト索引の名前がハンドルの一部になっています。

**ユーザーの処置:** DesGetBrowseInfo への入力として使用するハンドルが正しいかどうかを確認してください。

---

### RC\_SE\_UNKNOWN\_SECTION\_NAME

**説明:** 指定したセクション名は、文書モデル・ファイル中に指定されているモデルや、使用されている省略時モデルの一部ではありません。

**ユーザーの処置:** 指定されているモデルまたは省略時モデルの一部であるセクション名を指定してください。

---

### RC\_SE\_UNKNOWN\_SERVER\_NAME

**説明:** DB2 テキスト・エクステンダー・サーバーの名前がハンドルの一部になっています。

**ユーザーの処置:** DesGetBrowseInfo への入力として使用するハンドルが正しいかどうかを確認してください。

---

### RC\_SE\_WRITE\_TO\_DISK\_ERROR

**説明:** DB2 テキスト・エクステンダー・サーバー・ワークステーション上でディスクがいっぱいになったため、または DB2 テキスト・エクステンダーのインストール方法に誤りがあったために発生した書き込みエラーです。

**ユーザーの処置:** エラーの原因究明に役立つと思われる情報を保管し、アプリケーションを終了させてください。サーバーのディスク・スペースに十分空きがあるかチェックしてください。

---

### RC\_SQL\_ERROR\_WITH\_INFO

**説明:** SQL エラーが起きました。エラー・メッセージが戻されます。

**ユーザーの処置:** 詳しくは、SQL エラー・メッセージや SQLState およびネイティブの SQL エラー・コードなどの、DB2 テキスト・エクステンダーから戻されるエラー・メッセージをチェックしてください。

---

### RC\_SQL\_ERROR\_NO\_INFO

**説明:** SQL エラーが起きました。エラー・メッセージは戻されません。



---

**RC\_TEXT\_COLUMN\_NOT\_ENABLED**

**説明:** 指定されたハンドル列は、指定の表内の列ではありません。

**ユーザーの処置:** 指定したハンドル列名が正しいかどうかチェックしてください。その表内のテキスト列が使用可能になっているか確認してください。

戻りコード

---

## 第18章 メッセージ

この章では、以下について説明します。

- **DB2 テキスト・エクステンダーの関数により戻される SQL の状態:** これらのメッセージは、DB2 テキスト・エクステンダーの関数を使用しているときに、表示されることがあります。
- **DB2TX コマンド行プロセッサからのメッセージ:** これらのメッセージは、コマンド行プロセッサ DB2TX を使用して管理コマンドを入力したときに、表示されることがあります。個々のメッセージ番号の接頭部は DES です。

---

### DB2 テキスト・エクステンダーの関数により戻される SQL の状態

DB2 テキスト・エクステンダーが備えている SQL 関数は、エラー状態に戻すことができます。例:

```
SQL0443N User-defined function
"DB2TX.CONTAINS" (specific name "DES5A")
has returned an error SQLSTATE with
diagnostic text "Cannot open message file".
SQLSTATE=38702
```

この節のメッセージは、SQLSTATE 番号順に記載されています。

---

#### 01H10 ファイル *file-name* を開けません。

**ユーザーの処置:** ファイルが存在するか、DB2 のインスタンス名にそのファイルをオープンする許可が与えられているかどうかを確認してください。

---

#### 01H11 テキスト・ハンドルが完了していません。

**説明:** 初期化されているハンドルを使用しようとしたのですが、完了しませんでした。文書の言語と書式についての事前設定値が入っている INIT\_TEXT\_HANDLE を使って、部分ハンドルが作成されました。しかし、そのハンドルはトリガーにより完了しませんでした。

**ユーザーの処置:** 完了したハンドルだけを使用してください。そのハンドルをハンドル列に保管する場合は、これに対応するテキスト列を使用可能にするか再使用可能にしてください。

---

#### 01H12 検索引き数が長過ぎます。2番目の検索引き数は無視されました。

**説明:** REFINE 関数を使用して2つの検索引き数が結合されましたが、結合された検索引き数の長さが LONG VARCHAR の最大値を超えています。REFINE 関数は、結合された検索引き数の代わりに最初の検索引き数を戻します。

**ユーザーの処置:** 検索引き数のどちらか一方または両方の長さを短縮してから、照会を繰り返してください。

## DB2 テキスト・エクステンダーの関数により戻される SQL の状態

---

**01H13**      検索引き数は停止ワードを含みません。

**説明:** 指定された照会には、停止ワードだけからなる少なくとも 1 つの検索用語が含まれていません。検索項目は照会の処理の間無視されました。

**ユーザーの処置:** DB2 テキスト・エクステンダー照会では停止ワードを使用しないでください。

---

**01H14**      言語処理のための言語辞書がありません。

**説明:** DB2 テキスト・エクステンダー言語サービスが辞書ファイルを検出できません。照会は言語的なサポートなしに処理されます。指定の言語コード (複数の場合もある) に対応する辞書ファイルが、あるべきパスにありません。

**ユーザーの処置:** UNIX の場合、必要な辞書がパス {DB2TX\_INSTOWNERHOMEDIR}/db2tx/dicts に入っているかどうかをチェックしてください。

OS/2 の場合、必要な辞書がテキスト構成設定内に指定されたパスにあるかどうかを調べてください。必要であれば、必要な辞書をインストールしてください。

---

**01H15**      言語検索項目の指定が索引タイプと一致しません。

**説明:** 照会の検索項目の基本語形指定が、索引のタイプに対応していません。たとえば、PRECISE FORM OF は基本語形索引と一緒に使用できません。省略時の基本語形指定は、229ページの表 8 に示すように使用されます。

**ユーザーの処置:** アプリケーションを変更して、その索引のタイプと対立する照会オプションを指定しないようにしてください。

---

**38700**      テキスト・エクステンダーのライブラリーが現行のものではありません。

**説明:** 後続バージョンの DB2 テキスト・エクステンダーでしか解釈できないハンドルを使用しようとしてしました。

**ユーザーの処置:** 現行ライブラリーのバージョンに対するパスが正しく設定されているか、また、このライブラリーへのアクセス許可が与えられているか確認してください。

この問題が起こる関数については、IMPLEMENTATION 列にある DB2 カタログ表示 SYSCAT.FUNCTIONS を参照してください。

---

**38701**      トレース・ファイル名 このトレース・ファイルをオープンすることができません。

**説明:** ディレクトリー DB2TX\_TRACEDIR 内のファイル DB2TX\_TRACEFILE に書き込みを行うトレース機能を使用しようとしてしました。ファイルが存在しないか見つからない、またはファイルに必要な許可が与えられていないかのいずれかです。

---

**38702**      メッセージ・ファイル *message-file* をオープンすることができません。

**説明:** DB2 テキスト・エクステンダーがメッセージを戻そうとするような状態が発生しました。メッセージを格納しているファイルが存在しないか見つからない、または使用が許可されていないかのどちらかです。

**ユーザーの処置:** ファイルが存在するか、パスの設定が正しいか、さらにファイルをオープンする許可が与えられているかどうかを確認してください。

**38704**      テキスト・ハンドルの形式が誤りです。

**説明:** DB2 テキスト・エクステンダーの関数の引き数として、形式が正しくないハンドルが使用されました。

**ユーザーの処置:** ハンドルが INIT\_TEXT\_HANDLE によって生成されたものではないことを確認してください。

**38705**      UDF 名 誤った UDF 宣言です。

**説明:** DB2 テキスト・エクステンダーの関数の特定の名前が、関数を宣言しているスクリプト内で変更されています。DB2 テキスト・エクステンダーの関数の名前は変更可能ですが、関数の特定の名前は変更できません。

**ユーザーの処置:** DB2 テキスト・エクステンダーの関数の宣言が入っているスクリプト DESCVDF.DDL をチェックして、まだ正しい名前が使用されているかどうかを確認します。名前が元の配布媒体内に保管されているものと同じかチェックしてください。

**38706**      属性 この属性値を認識できません。

**説明:** CCSID、書式、または言語に無効な値を設定しようとしてしました。

**ユーザーの処置:** 正しい値については、33ページの『第4章 必要に応じた検索の計画』を参照してください。

**38707**      要求された機能はまだサポートされていません。

**説明:** 指定された機能はまだサポートされていません。

**ユーザーの処置:** 指定された機能をチェックしてください。

**38708**      戻りコード

**説明:** 検索要求の処理中にエラーが発生しました。

**ユーザーの処置:** 299ページの『第17章 戻りコード』の戻りコードに関する説明を参照してください。

**38709**      使用可能なメモリーが不十分です。

**説明:** メモリー不足のため、DB2 テキスト・エクステンダーの関数を実行できません。

**ユーザーの処置:** 不要なアプリケーションをクローズしてメモリーを解放してから、再度実行してください。

**38710**      エラー番号 検索結果にアクセスできません。

**説明:** 検索サービスにより戻された検出文書のリスト（結果リスト）を読み取ろうとした場合に発生するエラーです。

**ユーザーの処置:** 検索を繰り返してください。それでもだめな場合は、検索サービスを再始動してください。それでもエラーが起こる場合は、最寄りの IBM 担当員にエラー番号を添えてご連絡ください。

**38711**      重大な内部エラーです。

**説明:** 重大エラーが発生しました。

**ユーザーの処置:** 最寄りの IBM 担当員に連絡し、重大エラーが発生したときの状況をお知らせください。

**38712**      索引名 このテキスト索引の中のテキスト・ハンドルは誤りです。

**説明:** ハンドルが損傷しています。

**ユーザーの処置:** UPDATE INDEX を使用して索引を作成し直してください。

## DB2 テキスト・エクステンダーの関数により戻される SQL の状態

---

### 38714 DB2TX\_INSTOWNERHOMEDIR 環境変数を短くしてください。

**説明:** インスタンス所有者のホーム・ディレクトリー名は、256 文字以下でなければなりません。

**ユーザーの処置:** リンクによりディレクトリー名の長さを短縮してください。

---

### 38717 指定されたシソーラス (類語辞典) が見つかりませんでした。

**説明:** 指定されたシソーラスが見つかりません。

**ユーザーの処置:** 指定されたシソーラス名をチェックしてください。

---

### 38718 指定された関係名をシソーラスの中 で見つけることができませんでした。

**説明:** 指定された関係が指定されたシソーラスの中に存在しません。

**ユーザーの処置:** 指定した関係が存在することを確認してください。

---

### 38719 検索処理エラーが発生しました。理 由コード: rc

**説明:** 指定された理由のために検索ができませんでした。

**ユーザーの処置:** 理由コードで報告される問題の解決を試みてください。指定された理由が役に立たず、それ以上の情報が `desdiag.log` ファイルに見つからない場合、トレースを作成してその情報を最寄りの IBM 担当員に報告してください。

---

### 38720 共用メモリーの接続エラーが発生し ました。

**説明:** システムは共用メモリーにアクセスできません。

**ユーザーの処置:** システム構成をチェックして共

用資源を増やすか、または現行の共用資源の使用 (ipcs) をチェックして必要のない資源をクリーンアップしてください。

---

### 38721 セマフォの作成またはアクセス・ エラーが発生しました。

**説明:** システムはセマフォの作成またはアクセスができません。

**ユーザーの処置:** システム構成をチェックして共用資源を増やすか、または現行の共用資源の使用 (ipcs) をチェックして必要のない資源をクリーンアップしてください。

---

### 38722 検索処理が戻りませんでした。

**説明:** 検索要求の処理中にエラーが発生しました。

**ユーザーの処置:** システム構成 `descfgcl` を検査してすべてのノードが起動して、稼働中であるかチェックしてください。

---

### 38723 索引および照会 CCSID が一致しま せん。

**説明:** 照会ストリングで使用されるデータベース CCSID がテキスト索引と同じではありません。

**ユーザーの処置:** テキスト索引を使用不可にした後データベースの CCSID を使用して再作成してください。

---

### 38724 セクションまたはモデル名が正しく ありません。

**説明:** 照会で指定されたセクションまたはモデル名が正しくありません。

**ユーザーの処置:** セクションまたはモデル名をチェックしてください。

**38726** モデル・ファイルの読み取りエラーが発生しました。

**説明:** モデル定義ファイルが存在しないか、またはオープンできません。

**ユーザーの処置:** モデル定義ファイルが索引ディレクトリーに存在するかチェックしてください。

## DB2 テキスト・エクステンダーからのメッセージ

各メッセージには、接頭部 (DES)、メッセージ番号、および接尾部文字から成るメッセージ識別コードが付いています。接尾部の文字は、そのメッセージを出したエラーの重大度を表します。

- I** 通知メッセージ
- W** 警告メッセージ
- N** エラー (または「否定」) メッセージ
- C** 重大エラー・メッセージ

**DES0001N db2txinstance** 引き数の数値が無効です。

**説明:** db2txinstance コマンドには 2 つの引き数が必要です。

**ユーザーの処置:** これらの引き数と共にコマンドを再入力してください。

db2txinstance instanceName db2InstanceName

ここで、*instanceName* はこのインスタンスの所有者として割り当てられている既存の UNIX ユーザーのログイン名で、*db2InstanceName* は対応する DB2 インスタンスの所有者のログイン名です。

**DES0002N 無効なインスタンス名です。**

**説明:** 指定するインスタンス名は既存の UNIX ユーザーのログイン名でなければなりません。

**ユーザーの処置:** インスタンス名を訂正するか、既存の UNIX ユーザーを選択するか、またはインスタンスの所有者となる UNIX ユーザーを作成してください。

db2txinstance コマンドを次のように再入力してください。

db2txinstance instanceName

ここで、*instanceName* は、選択された UNIX ユーザーのログイン名です。

**DES0004N 指定されたインスタンスは、すでに存在します。コマンドを処理できません。**

**説明:** *instanceName* は、インスタンスの所有者である UNIX ユーザーのログイン名を示します。このインスタンス所有者は、ホーム・ディレクトリーに db2tx ディレクトリーをすでに持っています。

**ユーザーの処置:** インスタンスを作成するには、既存のインスタンスを削除してからコマンドを再実行してください。

**DES0005N インストール・メッセージのカタログを検出できません。**

**説明:** インストール・スクリプトに必要なメッセージ・カタログがシステムにありません。削除されてしまったか、またはデータベース・プロダクトのロードが正しくなかったと思われます。

**ユーザーの処置:** db2tx\_01\_01\_0000.client プロダ

## DB2 テキスト・エクステンダーからのメッセージ

クト・オプションが正しくインストールされていることを確認してください。検査エラーが見つかった場合は、そのオプションをインストールし直してください。

---

### DES0015W 言語検索項目の指定が索引タイプと一致しません。

**説明:** 照会の検索項目の基本語形指定が、索引のタイプに対応していません。たとえば、PRECISE FORM OF は基本語形索引と一緒に使用できません。省略時の基本語形指定は、229ページの表8に示すように使用されます。

**ユーザーの処置:** アプリケーションを変更して、その索引のタイプと対立する照会オプションを指定しないようにしてください。

---

### DES0016W 現行索引タイプに対してサポートされていない言語が指定されています。

**説明:** ユーザーが指定した言語は、指定された索引タイプに対してサポートされていません。

**ユーザーの処置:** 索引タイプに対してサポートされている言語のリストについては資料を参照してください。

---

### DES0017W 特徴抽出が使用可能ではありません。

**説明:** 照会でフィーチャー検索引き数が使用されましたが、索引が、索引オプション FEATURE\_EXTRACTION を指定して作成されていません。

**ユーザーの処置:** 索引オプションを FEATURE\_EXTRACTION に変更してください。

---

### DES0018W 現行索引タイプに対してサポートされていないオプションが指定されています。

**説明:** ユーザーが要求した検索オプションは、現行索引タイプおよび索引オプションに対してサポートされていません。

**ユーザーの処置:** 要求した検索オプションをサポートする索引タイプまたは索引オプションをチェックしてください。229ページの表8を参照してください。

---

### DES0121N メモリーを割り振りができませんでした。

**説明:** アプリケーションをインストールするための記憶域を予約できませんでした。

**ユーザーの処置:** ページ・スペースを増やしてください。

---

### DES0333N テキスト索引ファイルの入出力に問題が発生しました。

**説明:** DB2 テキスト・エクステンダー・クライアントは、要求されたサーバーとのセッションを確立できませんでした。

**ユーザーの処置:** DB2 テキスト・エクステンダー・サーバーが開始されているかを確認してください。開始していない場合は、TXSTART を実行します。

---

### DES0709W 言語の辞書が未導入です。

**説明:** DB2 テキスト・エクステンダーが、辞書ファイルを見つけることができません。

**ユーザーの処置:** 指定した言語の辞書をインストールするか再インストールしてください。



---

**DES0377N** テキスト索引ファイルの入出力に問題が発生しました。

**説明:** DB2 テキスト・エクステンダーがテキスト索引にアクセスできません。これは、テキスト構成内の DIRECTORY 設定が無効な辞書を指し示す場合に起きる可能性があります。

**ユーザーの処置:** テキスト構成設定をチェックしてください。

---

**DES0700N** ノード番号の値 '*node*' はノード・グループの定義に含まれていません。

**説明:** 指定されたノード番号は無効です。

**ユーザーの処置:** DB2 ノード番号をチェックしてください。

---

**DES0701N** ノード番号の値 '*node*' は範囲外です。

**説明:** 指定されたノード番号は無効です。

**ユーザーの処置:** DB2 ノード番号をチェックしてください。

---

**DES0704N** フォーマット '*format*' は索引プロパティの指定が必要です。

**説明:** 文書の書式は索引タイプの情報と互換性がありません。

**ユーザーの処置:** 文書の書式と互換性のある索引プロパティを指定してください。

---

**DES0705N** 指定された文書のモデル名 '*model*' がモデル定義ファイルに見つかりませんでした。

**説明:** モデル定義ファイル中に文書モデル名がありませんでした。モデル名には大文字小文字の区別があるので注意してください。

**ユーザーの処置:** モデル定義ファイルで指定されているモデル名を使用してください。

---

---

**DES0706N** DB2 テキスト・エクステンダーのサーバーでモデル定義ファイルにアクセスできません。

**説明:** モデル定義ファイルが存在しないか、またはオープンできません。

**ユーザーの処置:** モデル定義ファイルが存在することをチェックしてください。

---

**DES0707N** フォーマット '*format*' は指定された索引プロパティをサポートしません。

**説明:** 文書の書式ではこの索引プロパティがサポートされていません。

**ユーザーの処置:** 文書の書式と互換性のある索引プロパティを指定してください。

---

**DES0710N** ヌル・ポインターはパラメーター '*parameter*' には許可されていません。

**説明:** *parameter* に値が指定されていません。

**ユーザーの処置:** パラメーターに値を指定してください。

---

**DES0711N** テキスト・エクステンダーの内部エラーが発生しました。診断情報:  
*message*。

**説明:** 内部処理エラーが発生しました。

**ユーザーの処置:** 診断メッセージをチェックして、問題を解決してください。内部エラーの原因がインストールに関する問題でない場合は、desdiag.log ファイルまたは作成されたトレース・ファイルに追加情報がある可能性があります。それでも解決できない場合は、利用可能な情報を収集して IBM 技術員に連絡してください。

---

## DB2 テキスト・エクステンダーからのメッセージ

---

**DES0712N** パラメーター '*parameter*' は長すぎます。

**説明:** 指定されたパラメーターは範囲外です。

**ユーザーの処置:** 有効な長さを使用してパラメーターを指定してください。

---

**DES0713N** パラメーター '*parameter*' の長さは誤りです: %d1。

**説明:** 指定されたパラメーターは範囲外です。

**ユーザーの処置:** 有効な長さを使用してパラメーターを指定してください。

---

**DES0714N** パラメーター *parameter* を直接指定するか、または構成表内に指定してください。

**説明:** CCSID、書式、または言語が指定されておらず、この値に対する構成設定がありません。

**ユーザーの処置:** 欠落しているパラメーターを直接 ENABLE TEXT COLUMN コマンドに指定するか、またはテキスト構成設定に値を設定してください。

---

**DES0715N** データ・タイプ *schema.type* はテキスト・データにはサポートされていません。

**説明:** *schema.type* は、テキスト列のスキーマ名およびタイプ名、またはアクセス関数の結果です。テキスト列のデータ・タイプは DB2 テキスト・エクステンダーではサポートされていません。これは、CHAR、GRAPHIC、VARGRAPHIC、LONG VARGRAPHIC、DBCLOB、VARCHAR、LONG VARCHAR または CLOB のいずれかでなければなりません。上記のデータ・タイプ以外の場合は、入力がテキスト列のデータ・タイプで出力が

VARCHAR、LONG VARCHAR、または LOB であるアクセス関数を使用しなければなりません。

**ユーザーの処置:** *schema.type* がテキスト列タイ

プの場合は、結果がタイプ VARCHAR、LONG VARCHAR、または LOB のアクセス関数を登録しなければなりません。 *schema.type* がアクセス関数の結果である場合、これは使用できません。結果が必須タイプのアクセス関数を使用してください。

---

**DES0716N** フォーマット *format* はサポートされていません。

**説明:** *format* は、テキスト・エクステンダーがサポートしていない書式です。

**ユーザーの処置:** 35ページの『サポートされる文書形式』のサポートされている書式のリストをチェックしてください。

---

**DES0717N** 言語 *language* はサポートされていません。

**説明:** *language* は DB2 テキスト・エクステンダーがサポートしていない言語であるか、または選択された言語が指定した索引タイプによってサポートされていないかのいずれかです。

**ユーザーの処置:** 59ページの表5 のサポートされている言語のリストをチェックしてください。

---

**DES0718N** CCSID *ccsid\_value* はサポートされていません。

**説明:** 無効な CCSID 値が指定されました。

**ユーザーの処置:** サポートされている CCSID のリストについては、資料を参照してください。

---

**DES0719N** DB2 テキスト・エクステンダーのプログラム *program* へのコールはリターン・コード *rc* で失敗しました。

**説明:** インストールの最中にエラーが発生したと考えられます。DES\_EXT.H ファイル内に戻りコードがリストされています。

**ユーザーの処置:** インストールが正常に終了した

かどうかをチェックしてください。

DB2TX\_INSTOWNER や

DB2TX\_INSTOWNERHOMEDIR などの環境変数が正しく設定されているかどうかを調べてください。

---

**DES0720N** アクセス関数 *schema.function* はデータベースに登録されていません。

**説明:** 関数名が間違っているか、またはデータベースに登録されていません。

**ユーザーの処置:** アクセス関数の名前をチェックしてください。正しい場合は、その関数がデータベース・システムに認識されているかどうかチェックします。アクセス関数をデータベースに登録するには、CREATE FUNCTION を使用します。

---

**DES0721N** データベースが不整合です。DB2 テキスト・エクステンダー・カタログ表示がありません。

**説明:** DB2 テキスト・エクステンダー・カタログ表示の 1 つがデータベース内に存在しません。

**ユーザーの処置:** DISABLE DATABASE コマンドで残りのカタログ表示を除去してから、再び ENABLE DATABASE コマンドを入力してください。索引データが失われています。テキスト文書に索引を付け直してください。

---

**DES0722N** 表 *schema.table* はデータベースの基本表ではありません。

**説明:** 表がデータベース内に存在しないか、または表が結果表か視点となっています。テキスト列は、基礎表内になければ DB2 テキスト・エクステンダーで使用できるようになりません。

**ユーザーの処置:** 表名が正しいかどうか、またそれが基礎表であるか確認してください。

---

**DES0723N** 表 *schema.table* のハンドル列 *handlecolumnname* のテキスト索引の (再) 作成に失敗しました。

**説明:** ハンドル列のテキスト索引を作成できませんでした。

**ユーザーの処置:** txstatus を使ってサーバーの状態を調べてください。サーバー上で保守が正常に行われている場合は、DISABLE TEXT COLUMN または DISABLE TEXT TABLE を使用して、一貫性のある状態に戻してください。その後、ENABLE TEXT COLUMN または ENABLE TEXT TABLE を使用して、テキスト列を再び使用可能にします。

---

**DES0724N** 表 *schema.table* のハンドル列 *handlecolumn* のテキスト索引のカタログ表示への入力が誤りです。

**説明:** TextIndices カatalog表示が損傷しています。

**ユーザーの処置:** DISABLE TEXT COLUMN または DISABLE TEXT TABLE を使用して、一貫性のある状態に戻してください。その後、ENABLE TEXT COLUMN または ENABLE TEXT TABLE を使用してテキスト列を使用可能にします。

---

**DES0727N** 列 *column* は表 *schema.table* で既に使用可能です。

**説明:** このメッセージは、最初に列を使用不能にしないで、同じテキスト列を使って表を消去してから作成し直した場合に発生することがあります。

**ユーザーの処置:** 列を使用不能にしてから再試行してください。

## DB2 テキスト・エクステンダーからのメッセージ

---

**DES0728N** 列 *column* は表 *schema.table* に存在しません。

**説明:** 存在しないテキスト列を使用可能にしようとしています。

**ユーザーの処置:** 表名または列名を変更してから再試行してください。

---

**DES0729N** ハンドル列 *handlecolumn* は表 *schema.table* に存在しません。

**説明:** 存在しないハンドル列を使用しようとしています。

**ユーザーの処置:** GET STATUS コマンドを使用して、ハンドル列が存在するかどうか、さらにその名前が正しく指定されているかチェックしてください。

---

**DES0730N** 表 *schema.table* は既に共通索引表として使用可能です。

**説明:** すでに共通索引表として使用可能になっている表を使用可能にしようとしています。

**ユーザーの処置:** 表を使用可能にせずに継続するか、または DISABLE TEXT TABLE コマンドを実行して使用不能にしてから再び使用可能にしてください。

---

**DES0731N** 表 *schema.table* は DB2 テキスト・エクステンダーには使用できず、使用不能にすることもできません。

**説明:** 使用可能になっていない表を使用不能にしようとしています。

**ユーザーの処置:** 表名をチェックしてください。

---

**DES0732N** ロケーション *location* 付近の更新頻度は誤りです。期待されるのは *parameter* です。

**説明:** 更新頻度の *parameter* 指定が正しくありませんでした。

**ユーザーの処置:** 更新頻度パラメーターをチェックし、コマンドを再入力してください。

---

**DES0733N** 表 *schema.table* には使用可能な列が含まれています。それを共通索引表として使用可能にすることはできません。

**説明:** この表には、すでに独自の索引を持っているテキスト列が含まれています。この固有の索引が存在する限り、すべてのテキスト列に共通の索引を作成することはできません。

**ユーザーの処置:** DISABLE TEXT COLUMN を使用して使用可能な列を使用不能にし、その後再び ENABLE TEXT TABLE コマンドを入力してください。

---

**DES0734N** ハンドル列 *handlecolumn* は部分テキスト表 *schema.table* に属しているため、別個に使用不能にすることはできません。

**説明:** 部分テキスト表として使用可能だった表内で、単一のテキスト列を使用不能にすることはできません。

**ユーザーの処置:** 部分テキスト表全体を使用不能にしてください。

---

**DES0736N** *handlecolumn* は既に表 *schema.table* のハンドル列です。

**説明:** 既存のハンドル列名を使用しようとしています。

**ユーザーの処置:** 別のハンドル列名を使用してコマンドを再入力してください。

---

**DES0737N** 表 *schema.tablename* は **STORAGE** オプション *storage\_option* を使用して共通索引表として使用可能です。

**説明:** 外部ファイルに対して共通索引表を使用可能にすることはできません。

**ユーザーの処置:** 外部ファイルに対して表を使用可能にしたい場合は、複数索引表を使用してください。

**DES0738N** アクセス関数 *schema.function* は誤ったパラメーターを持っています。

**説明:** *schema.function* の入力パラメーターまたは出力パラメーターに誤りがあります。

- 入力パラメーターは 1 つしか使用できず、そのデータ・タイプは使用可能にするテキスト列のデータ・タイプでなければなりません。
- 出力パラメーターのタイプは CHAR、GRAPHIC、VARGRAPHIC、LONG VARGRAPHIC、DBCLOB、VARCHAR、LONG VARCHAR または CLOB のいずれかでなければなりません。

**DES0739W** 表 *schema.table* ハンドル列 *handlecolumn* の索引の更新プログラムが開始できませんでした。

**説明:** 索引を更新するプログラムが開始できませんでした。インストールの最中にエラーが発生したと考えられます。

**ユーザーの処置:** インストールが正常に終了したかどうかをチェックしてください。

DB2TX\_INSTOWNER や

DB2TX\_INSTOWNERHOMEDIR などの環境変数が正しく設定されているかどうかを調べてください。

**DES0740N** ハンドル列 '*column*' はタイプ *columnntype* のテキスト列にのみ、再使用できます。

**説明:** このハンドル列は、すでに別のハンドル列タイプで使用されています。

**ユーザーの処置:** 新しいハンドル列名を指定してください。

**DES0741N** プログラムまたはファイル *parameter* が見つからないか、開始できませんでした。

**説明:** ENABLE DATABASE または DISABLE DATABASE コマンドでファイル *parameter* をオープンすることができませんでした。インストールの最中にエラーが発生したと考えられます。

**ユーザーの処置:** インストールが正常に終了したかどうかをチェックしてください。

**DES0742N** この表は列索引を使用します。コマンドにハンドル列を指定してください。

**説明:** 指定された表は、複数索引表として使用できます。特定の列を処理するには、関連するハンドル列を指定してください。

**ユーザーの処置:** ハンドル列を指定してください。

**DES0745N** **DB2TX** インスタンスの所有者 *instance-owner* は有効なユーザー ID ではありません。

**説明:** 環境変数 DB2TX\_INSTOWNER に、正しいユーザー ID が含まれていません。

**ユーザーの処置:** 環境変数を訂正してください。

## DB2 テキスト・エクステンダーからのメッセージ

---

**DES0747N** 現行の **CCSID** は索引タイプ *index\_type* にはサポートされていません。

**説明:** 要求される索引タイプに対してサポートされない **CCSID** が指定されました。

**ユーザーの処置:** サポートされている **CCSID** のリストについては、資料を参照してください。

---

**DES0748N** コミット・カウント '*commitcount*' は **DB2** テキスト・エクステンダーではサポートされていません。

**説明:** 指定されたコミット・カウント値はサポートされていません。

**ユーザーの処置:** 有効なコミット・カウント値を指定してください。

---

**DES0749N** 更新索引値 '*indexvalue*' は **DB2** テキスト・エクステンダーでは認識されません。

**説明:** 指定された '更新索引' 値が無効です。

**ユーザーの処置:** 有効な更新索引値を指定してください。

---

**DES0750N** 索引タイプ値 '*indextype*' は **DB2** テキスト・エクステンダーでは認識されません。

**説明:** 指定された索引タイプ値が無効です。

**ユーザーの処置:** 有効な索引タイプ値を指定してください。

---

**DES0751N** 指定された操作を実行する権限がありません。

**説明:** この操作を実行するために必要なデータベース管理者権限が与えられていません。

**ユーザーの処置:** この操作はデータベース管理者が行ってください。

---

**DES0756N** そのデータベースは **DB2** テキスト・エクステンダーには使用できません。

**説明:** このコマンドを実行する前にデータベースを使用可能にしなければなりません。

**ユーザーの処置:** **ENABLE DATABASE** を実行してからコマンドを再実行してください。

---

**DES0763N** 更新頻度が誤っています。

**説明:** 指定された '更新頻度' 値が無効です。

**ユーザーの処置:** 有効な '更新頻度' を指定してください。

---

**DES0765N** そのデータベースは、既に **DB2** テキスト・エクステンダーには使用可能です。

**説明:** すでに使用可能となっているデータベースを使用可能にしようとしています。

**ユーザーの処置:** データベースを使用可能とせずに継続するか、または **DISABLE DATABASE** で使用不能にしてからこれを再び使用可能にしてください。

---

**DES0766N** 処理によって表または一時表の最大行サイズを越えました。

**説明:** **ENABLE TEXT COLUMN** コマンドは、ハンドル列を表に追加します。表がもともと大きい場合は、表の行サイズが最大値 4005 を超える場合があります。

**ENABLE TEXT COLUMN** コマンドは、すでに使用可能となっているテキスト列の数に釣り合ったサイズの一時表も作成します。多くのテキスト列がすでに使用可能になっている場合は、一時表のサイズが最大値を超えることがあります。

**ユーザーの処置:** **ENABLE TEXT COLUMN** は、この制限を超えない表に限り使用してください。

---

**DES0769W** 警告: 索引の特性が指定されましたが、無視されます。表 '*tablename*' は共通索引表です。

説明: 指定された表は共通索引表なので、索引特性を指定できません。

ユーザーの処置: 何もする必要はありません。

---

**DES0770N** 環境変数 *env-variable* が定義されていません。

説明: コマンドにパラメーターが指定されていないため、システムは環境変数 *env-variable* から省略時の値を読み取ろうとしましたが、この環境変数は定義されていませんでした。

ユーザーの処置: 必要な環境変数を定義してください。

---

**DES0774N** 変数 '*variable*' の値の長さ *length* は範囲外です。

説明: パラメーターの値の長さが範囲外です。

ユーザーの処置: 有効な長さを使用してパラメーターを指定してください。

---

**DES0775N** 索引ディレクトリーの値 '*directory*' は誤りです。

説明: 索引ディレクトリーの値が誤っています。ディレクトリーの長さが誤っている可能性があります。

ユーザーの処置: 有効な索引ディレクトリーの値を指定してください。

---

**DES0776N** 表スペース名 '*tablespace*' は誤りです。

説明: 指定された表スペース名が誤っています。表スペースの値の長さが誤っている可能性があります。

ユーザーの処置: 有効な表スペースの値を指定してください。

---

**DES0777N** 表スペース *tablespace* はデータベース管理システムには認識されません。

説明: 指定された表スペースはデータベース・システムに認識されていません。

ユーザーの処置: 指定した表スペースがデータベース中に存在することをチェックしてください。

---

**DES0778N** '*tablespace*' は正規の表スペースではありません。キーワード '*keyword*' で作成されました。

説明: 指定された表スペースのデータ・タイプはサポートされていません。

ユーザーの処置: 正規の表スペースを指定してください。

---

**DES0779I** 索引の作成は正常に開始されました。索引の作成状況をチェックするには '**GET INDEX STATUS**' を使用してください。

説明: 索引付けプログラムが開始しています。'**GET INDEX STATUS**' コマンドを使用して、索引付けプロセスの状況をチェックできます。

ユーザーの処置: **GET INDEX STATUS** コマンドの出力をチェックしてください。

---

**DES0780I** 索引の再編成は正常に開始されました。索引の状況をチェックするには '**GET INDEX STATUS**' を使用してください。

説明: 再編成プログラムが開始しています。'**GET INDEX STATUS**' コマンドを使用して、再編成プロセスの状況をチェックできます。

ユーザーの処置: **GET INDEX STATUS** コマンドの出力をチェックしてください。

---

## DB2 テキスト・エクステンダーからのメッセージ

---

**DES0789W** 警告: 現在のノードグループの区分化マップが更新されました。  
**TXNCHECK** ユーティリティを呼び出してください。

説明: 現行ノードグループの区分化マップが更新されています。

ユーザーの処置: TXNCHECK コマンドを使用してください。

---

**DES0800I** *command* コマンドが正常に完了しました。

説明: 指定されたコマンドは正常に完了しました。

ユーザーの処置: 何もする必要はありません。

---

**DES0810N** 閉じの引用符がありません。

説明: 引用符が検出されましたが、後ろの引用符が欠落しています。

ユーザーの処置: コマンド構文をチェックして再試行してください。

---

**DES0811N** "*token*" は予期されません。索引特性またはテキスト情報をチェックしてください。

説明: 索引の特性またはテキスト情報に誤りがあります。

ユーザーの処置: 構文をチェックして再試行してください。

---

**DES0812N** 表 *schema.table* が存在しないか、または **DB2** テキスト・エクステンダーには使用できません。

説明: GET コマンドの実行中に、データベース表の名前が間違っているか存在しない、またはまだ使用可能になっていないかのいずれかです。

ユーザーの処置: 表名が正しい場合は、GET STATUS コマンドを使用してこれが使用可能にな

っているかチェックしてください。表を使用可能にし、再試行してください。

---

**DES0813N** 表 *schema.table* が存在しないか、**DB2** テキスト・エクステンダーには使用できないか、またはハンドル列 *column* を含んでいません。

説明: GET コマンドの実行中に、ハンドル列の項目が表内に見つかりません。表が存在する場合は、これが使用可能になっていないかハンドル列が含まれていません。

ユーザーの処置: 表名が正しい場合は、GET STATUS コマンドを使用してこれが使用可能になっているかチェックしてください。表を使用可能にし、再試行してください。

---

**DES0814N** 表 *tablename* が存在しないか、**DB2** テキスト・エクステンダーでは使用可能でないか、またはこの表内で使用可能なテキスト列がありません。

説明: 指定された表は **DB2** テキスト・エクステンダーには使用できません。

ユーザーの処置: 表が **DB2** テキスト・エクステンダーで使用できるようにしてください。

---

**DES0815N** 空の引用符 "" が検出されました。名前を引用符の中に入れてください。

説明: 2 つの引用符の間にテキストが入力されていないものが検出されました。

ユーザーの処置: 構文をチェックして再試行してください。



---

**DES0816N** ワード *"token"* は予期されません。キーワード *keyword* または *keyword* のいずれかを使用してください。

説明: 予想外のトークンが検出されました。

ユーザーの処置: メッセージに示されているキーワードのどちらかを、コマンド内に指定してください。

---

**DES0817N** *"token"* は予期されません。キーワード *keyword* を使用してください。

説明: 予想外のトークンが検出されました。

ユーザーの処置: メッセージに示されているキーワードをコマンド内に指定してください。

---

**DES0818N** 予期しないコマンドの終了です。キーワード *keyword* が見込まれます。

説明: キーワードが欠落しています。

ユーザーの処置: メッセージに示されているキーワードをコマンド内に指定してください。

---

**DES0819N** 予期しないコマンドの終了です。次のキーワードのいずれかが見込まれます。 *keyword* または *keyword*。

説明: キーワードが欠落しています。

ユーザーの処置: メッセージに示されているキーワードのどちらかを、コマンド内に指定してください。

---

**DES0820N** 索引オプション *index\_option* は索引タイプ *index\_type* にはサポートされていません。

説明: 要求された索引タイプに対してサポートされない索引オプションが指定されました。

ユーザーの処置: 要求する索引タイプに対してサ

ポートされている索引オプションについて資料を参照してください。

---

**DES0821N** 名前 *"token"* が長すぎます。 *variable* の名前に許可されるのは、 *nn* 文字だけです。

説明: 名前の長さが指定範囲を超えています。

ユーザーの処置: 有効な長さの名前を指定してください。

---

**DES0822N** コマンドに認識されないトークン *"token"* が含まれています。コマンドの終了が見込まれます。

説明: コマンドの終わりを検出しましたが、キーワードが必要です。

ユーザーの処置: コマンド構文をチェックして再試行してください。

---

**DES0823N** *"schema"* に続いて表名が見込まれます。

説明: “.” の後に、表名または関数名が欠落しています。

ユーザーの処置: コマンド構文をチェックして再試行してください。

---

**DES0824N** 予期しないコマンドの終了です。 *keyword* の値は必須です。

説明: メッセージ内のキーワードが構文から欠落しています。

ユーザーの処置: コマンド構文をチェックして再試行してください。

---

**DES0825N** *'alias'* は認識されているデータベースの別名ではありません。許可されるのは *nn* 文字だけです。

説明: 指定されたデータベース別名は、データベース・システムに認識されていません。

## DB2 テキスト・エクステンダーからのメッセージ

**ユーザーの処置:** 指定した別名が有効であることをチェックしてください。

---

**DES0826N** データベース別名 *alias* を引用符で囲んではいけません。

**説明:** メッセージ内の名前がデータベース別名として解釈されています。これは引用符で囲んではいけません。

**ユーザーの処置:** コマンド構文をチェックして再試行してください。

---

**DES0827N** CCSID "*ccsid*" はサポートされていません。

**説明:** CCSID は DB2 テキスト・エクステンダーではサポートされていません。

**ユーザーの処置:** サポートされている CCSID のリストについては、資料を参照してください。

---

**DES0829N** ユーザー名 *userid* は引用符で囲む必要はありません。

**説明:** ユーザー名が引用符で囲んで入力されました。

**ユーザーの処置:** 引用符を取り除いてください。

---

**DES0830N** パラメーター "*parameter*" は *enable/disable* **DATABASE** コマンドで認識されません。コマンドの終了が見込まれます。

**説明:** コマンド ENABLE DATABASE および DISABLE DATABASE には、パラメーターを指定できません。

**ユーザーの処置:** パラメーターを指定せずにコマンドを入力し直してください。

---

**DES0831N** 予期しないコマンドの終了です。表名がありません。

**説明:** コマンドには表名が必要です。

**ユーザーの処置:** 適切な表名を入力してください。

---

**DES0832N** 予期しないコマンドの終了です。データベース名がありません。

**説明:** コマンドにはデータベース名が必要です。

**ユーザーの処置:** 適切なデータベース名を入力してください。

---

**DES0833N** 予期しないコマンドの終了です。列名がありません。

**説明:** コマンドには列名が必要です。

**ユーザーの処置:** 適切な列名を入力してください。

---

**DES0899N** 未知の DB2TX コマンド: *command*

**説明:** 指定されたコマンドは、DB2 テキスト・エクステンダーではサポートされていません。

**ユーザーの処置:** **db2tx ?** と入力して、コマンドのリストを入手してください。

---

**DES0971N** 索引ディレクトリーはノード指定なしで一度、ノード指定ありで複数回指定することができます。

**説明:** 索引ディレクトリーの指定が正しくありません。

**ユーザーの処置:** 索引ディレクトリーの指定をチェックしてください。ノードを指定せずに 1 つの索引ディレクトリーを指定するか、またはノードを指定して複数の索引ディレクトリーを指定することができます。

---

**DES0972N** ノード指定が誤りです。符号なしの数値が要求されています。

**説明:** ノード番号として数字でない値が指定されています。

**ユーザーの処置:** ノード番号には符号なしの数値を指定してください。

---

**DES0973N** ノード指定が誤りです。右括弧の前に 1 文字以上の予期しない文字が検出されました。

**説明:** ノードを指定した構文が誤っています。

**ユーザーの処置:** ノードを指定した構文をチェックして再試行してください。

---

**DES0974N** TO 文節のノード番号の順序が誤りです (2 番目のノードが最初のノードより小さい)。

**説明:** ノードを指定した構文が誤っています。

**ユーザーの処置:** ノードを指定した構文をチェックして再試行してください。

---

**DES0975N** ノード情報の指定構文が誤りです。

**説明:** ノードを指定した構文が誤っています。

**ユーザーの処置:** ノードを指定した構文をチェックして再試行してください。

---

**DES0976N** ノード情報が不完全です。

**説明:** ノードの指定が不完全です。一部の情報が欠落しています。

**ユーザーの処置:** ノードを指定した構文をチェックして再試行してください。

---

**DES0977N** ノード情報が不完全です。左括弧がありません。

**説明:** ノードの指定が不完全です。左括弧が欠落しています。

**ユーザーの処置:** ノードの指定を訂正して再試行してください。

---

**DES0998N** 文書モデル名の長さが正しくありません。

**説明:** '文書モデル名' 値の長さが誤っています。

**ユーザーの処置:** モデル名の値をチェックして再試行してください。

---

**DES0999N** 文書モデルの指定構文が正しくありません。

**説明:** モデル名を指定した構文が誤っています。

**ユーザーの処置:** モデルを指定した構文をチェックして再試行してください。

---

**DES9994N** テキスト検索エンジン・エラーが発生しました。理由コード:  
**reason-code**

**説明:** DB2 テキスト・エクステンダーが使用するテキスト検索エンジンにエラーが発生しました。

**ユーザーの処置:** 327ページの『第19章 検索エンジンの理由コード』を参照して検索エンジンの理由コードをチェックしてください。報告された理由が問題の解決に役立たなかった場合は、desdiag.log ファイルまたは作成されたトレース・ファイルに追加情報がある可能性があります。それでも解決できない場合は、利用可能な情報を収集して IBM 技術員に連絡してください。

---

**DES9995N** テキスト・エクステンダー・エラーが発生しました。メッセージ・テキスト: **message-text**

**説明:** DB2 テキスト・エクステンダーのエラーが発生しました。

**ユーザーの処置:** DB2 テキスト・エクステンダーからのメッセージを使用して、問題を解決してください。報告されたメッセージが問題の解決に

## DB2 テキスト・エクステンダーからのメッセージ

役立たなかった場合は、`desdiag.log` ファイルまたは作成されたトレース・ファイルに追加情報がある可能性があります。それでも解決できない場合は、利用可能な情報を収集して IBM 技術員に連絡してください。

---

**DES9996N DB2 テキスト・エクステンダーの内部エラーが発生しました。理由コード: *reason\_code***

**説明:** 内部処理エラーが発生しました。

**ユーザーの処置:** DB2 テキスト・エクステンダーのインストールが正常に完了しているかをチェックしてください。正常に完了している場合は、理由コードを書き留めて、IBM 技術員に連絡してください。

---

**DES9997N SQL エラーが発生しました。SqlState: *state* QL エラー・コード: *rc*; SqlErrorMessage: *message***

**説明:** SQL エラーが発生しました。

**ユーザーの処置:** メッセージと共に表示される SQL エラー・メッセージに基づいて処置してください。

---

**DES9998N SQL エラーが発生しました。それ以上の情報はありません。**

---

**DES9999N 対応するエラー・メッセージがありません。**

**説明:** 内部処理エラーが発生しました。

**ユーザーの処置:** 診断メッセージをチェックして、問題を解決してください。インストールに関する問題が内部エラーの原因でない場合は、`desdiag.log` ファイルまたは作成されたトレース・ファイルに追加情報がある可能性があります。それでも解決できない場合は、利用可能な情報を収集して IBM 技術員に連絡してください。

## 第19章 検索エンジンの理由コード

この章では、DB2 テキスト・エクステンダーの検索エンジンによって戻される理由コードを記載します。

表 21. 検索エンジンの理由コード

理由コード	値
RC_DONE	0
RC_CONTINUATION_MODE_ENTERED	1
RC_END_OF_INFORMATION	2
RC_EMPTY_LIST	3
RC_MORE_INFORMATION	4
RC_INDEX_GROUP_SEARCH_ERROR	7
RC_INDEX_SPECIFIC_ERROR	8
RC_DICTIONARY_NOT_FOUND	9
RC_PROCESSING_LIMIT_EXCEEDED	12
RC_UNKNOWN_SERVER_NAME	16
RC_INCORRECT_AUTHENTICATION	17
RC_DATASTREAM_SYNTAX_ERROR	18
RC_QUERY_SCOPE_TOO_COMPLEX	20
RC_QUERY_TOO_COMPLEX	22
RC_MEMBER_OF_INDEX_GROUP	23
RC_UNKNOWN_INDEX_NAME	24
RC_INCORRECT_HANDLE	25
RC_INDEX_NOT_MEMBER_OF_GROUP	26
RC_UNKNOWN_SESSION_POINTER	27
RC_UNKNOWN_COMMUNICATION_TYPE	29
RC_UNKNOWN_SERVER_INFORMATION	30
RC_INVALID_MASKING_SYMBOL	31
RC_UNEXPECTED_ERROR	32
RC_SERVER_NOT_AVAILABLE	33
RC_INDEX_ALREADY_OPENED	35
RC_MAX_NUMBER_OF_OPEN_INDEXES	36
RC_MAX_NUMBER_OF_RESULTS	37

## 検索エンジンの理由コード

表 21. 検索エンジンの理由コード (続き)

理由コード	値
RC_CCS_NOT_SUPPORTED	41
RC_LANGUAGE_NOT_SUPPORTED	42
RC_CONFLICT_WITH_INDEX_TYPE	43
RC_MAX_INPUT_SIZE_EXCEEDED	46
RC_SERVER_BUSY	47
RC_SERVER_CONNECTION_LOST	48
RC_SERVER_IN_ERROR	49
RC_REQUEST_IN_PROGRESS	50
RC_UNKNOWN_INDEX_TYPE	51
RC_INCORRECT_INDEX_NAME	52
RC_INCORRECT_LS_EXECUTABLES	53
RC_INCORRECT_LIBRARY_ID	54
RC_INDEX_ALREADY_EXISTS	55
RC_MAX_NUMBER_OF_INDEXES	56
RC_INCORRECT_LOCATION	57
RC_LOCATION_IN_USE	58
RC_UNKNOWN_CONDITION	59
RC_INDEX_DELETED	60
RC_INDEX_SUSPENDED	61
RC_INDEX_NOT_ACCESSIBLE	62
RC_MAX_NUMBER_OF_BUSY_INDEXES	63
RC_CONFLICTING_TASK_RUNNING	64
RC_NOT_ENOUGH_MEMORY	65
RC_MAX_OUTPUT_SIZE_EXCEEDED	68
RC_COMMUNICATION_PROBLEM	70
RC_NO_ACTION_TAKEN	71
RC_EMPTY_INDEX	72
RC_EMPTY_QUERY	73
RC_INSTALLATION_PROBLEM	74
RC_FUNCTION_DISABLED	75
RC_FUNCTION_IN_ERROR	76
RC_IO_PROBLEM	77
RC_WRITE_TO_DISK_ERROR	78
RC_SERVER_VERSION_NOT_CURRENT	79

表 21. 検索エンジンの理由コード (続き)

理由コード	値
RC_FUNCTION_NOT_SUPPORTED	80
RC_RESULT_ALREADY_RANKED	81
RC_RESULT_VIEW_EXISTS	82
RC_INDEX_NOT_OPEN	83
RC_NO_RANKING_DATA_AVAILABLE	84
RC_LINGUISTIC_SERVICE_FAILED	85
RC_THESAURUS_PROBLEM	86
RC_INVALID_IDENTIFIER	88
RC_DOCUMENT_MODEL_ALREADY_EXISTS	89
RC_UNKNOWN_DOCUMENT_SECTION_NAME	90
RC_DOCMOD_READ_PROBLEM	91
RC_UNKNOWN_DOCUMENT_MODEL_NAME	92
RC_SECTION_NAME_ALREADY_EXISTS	94
RC_SECTION_TAG_ALREADY_EXISTS	95
RC_MAX_NUMBER_OF_TASKS	96
RC_LS_NOT_EXECUTABLE	97
RC_LS_FUNCTION_FAILED	98
RC_CAPACITY_LIMIT_EXCEEDED	99
RC_DOCUMENT_NOT_ACCESSIBLE	100
RC_DOCUMENT_CURR_NOT_ACCESSIBLE	101
RC_DOCUMENT_NOT_TO_INDEX	102
RC_DOCUMENT_NOT_FOUND	103
RC_DOCUMENT_IN_ERROR	104
RC_DOCUMENT_NOT_SUPPORTED	105
RC_CROSSIDX_SEARCH_NOT_ALLOWED	110
RC_DOCUMENT_GROUP_NOT_FOUND	111
RC_INVALID_ATTRIBUTE_VALUE	112
RC_INVALID_SECTION_TYPE	113
RC_INCORRECT_RELEVANCE_VALUE	120
RC_NO_RAT_EXPANSION	130
RC_DOCUMENT_NOT_IN_VIEW	131

## 検索エンジンの理由コード



---

## 第20章 エラー事象理由コード

この章では、DB2 テキスト・エクステンダーが文書を索引付けするときに発生するおそれのあるエラー事象を列記します。これは、たとえば、次のような場合に起きることがあります。

- 文書を索引付けすることができない。
- 文書は索引付けされたが、問題が発生した。
- 言語辞書が見つからない。

### ヒント

理由コードが文書化されない場合

1. ディスク・スペースに十分空きがあるかチェックしてください。
2. できる限りエラー情報を収集してください。
  - desdiag.log ファイル
  - 事象メッセージ
3. IBM サービス担当員に連絡してください。

- 1 記憶域が足りません。サーバーにメモリーが十分ではありません。作業負荷を減らしてください。

### 116

データ・ストリーム構文エラー

### 280

文書は索引付けされていません。索引ファイルの 1 つをオープンできませんでした。

### 281

文書は索引付けされていません。索引ファイルの 1 つを読み取れませんでした。

### 441

文書は索引付けされていません。このメッセージが生じるのは、Ngram 索引だけです。文書のコード・ページと索引が作成されたコード・ページとが異なります。これは、索引が UTF8 で作成されていない場合に HTML および XML 文書で生じることがあります。

## エラー事象理由コード

### 500

文書は索引付けされていません。ライブラリー・サービスをロードできませんでした。DLL が使用可能であること、およびリソース・パスが有効であることを確認してください。

### 501

文書は索引付けされていません。ライブラリー・サービス内の Lib\_Init はフラット・ファイル・システムで失敗しました。DIT ファイルが見つからないか、有効なディレクトリーにないか、または DIT の内容が正しくありません。

### 502

文書は索引付けされていません。ライブラリー・サービス LIB\_read\_doc\_content 内の文書の内容を読み取っているときに、エラーが発生しました。

### 503

文書は索引付けされていません。ライブラリー・サービス LIB\_access\_doc 内でエラーが発生しました。

### 504

文書は索引付けされていません。ライブラリー・サービス LIB\_doc\_index\_status は、エラーを返しました。

### 505

文書のクローズに失敗しました。ライブラリー・サービス LIB\_close\_doc は、エラーを返しました。

### 506

ライブラリー・サービスの終了に失敗しました。ライブラリー・サービス LIB\_end はエラーを返しました。

### 507

ライブラリー・サービス呼び出し LIB\_read\_doc\_content が、予期しないリターン・コードを戻して失敗しました。

### 508

ライブラリー・サービス呼び出し LIB\_close\_doc が、RC\_TERMINATION エラーを戻しました。

### 545

文書は索引付けされていません。一時索引ファイルの 1 つをオープンできませんでした。

**546**

文書は索引付けされていません。一時索引ファイルの 1 つをクローズできませんでした。

**548**

内部エラー。診断ログ内の情報を IBM 担当員に送ってください。

**549、550**

記憶域がありません (割り当てに失敗しました)。サーバーにメモリーが十分ではありません。作業負荷を減らしてください。

**551-564**

文書は索引付けされていません。索引ファイルの 1 つのオープン、読み取り、書き込み、またはクローズができませんでした。索引に使用されているディスクのスペースに十分空きがあるか、およびアクセス権限が適正であるかをチェックしてください。

**565**

記憶域がありません (割り当てに失敗しました)。サーバーにメモリーが十分ではありません。作業負荷を減らしてください。

**566-587**

文書は索引付けされていません。索引ファイルの 1 つのオープン、読み取り、書き込み、またはクローズができませんでした。

**588-590**

内部エラー。診断ログ内の情報を IBM 担当員に送ってください。

**591-604**

文書は索引付けされていません。索引ファイルの 1 つのオープン、読み取り、書き込み、またはクローズができませんでした。

**605**

記憶域がありません (割り当てに失敗しました)。サーバーにメモリーが十分ではありません。作業負荷を減らしてください。

**606-623**

文書は索引付けされていません。索引ファイルの 1 つのオープン、読み取り、書き込み、またはクローズができませんでした。索引に使用されているディスクのスペースに十分空きがあるか、およびアクセス権限が適正であるかをチェックしてください。

**624**

記憶域がありません (割り当てに失敗しました)。サーバーにメモリーが十分ではありません。作業負荷を減らしてください。

## エラー事象理由コード

### 625-631

文書は索引付けされていません。索引ファイルの 1 つのオープン、読み取り、書き込み、またはクローズができませんでした。索引に使用されているディスクのスペースに十分空きがあるか、およびアクセス権限が適正であるかをチェックしてください。

### 632

索引付けで作成された一時ファイルの 1 つを、書き込みアクセスによってオープンできませんでした。アクセス権限を調べてください。

### 633

索引付けで作成された一時ファイルの 1 つをクローズできませんでした。

### 634

索引付けで作成された一時ファイルの 1 つに書き込めませんでした。索引作業ディレクトリーに十分なディスク・スペースがあるかを調べてください。

### 635

索引付けで作成された一時ファイルの 1 つを読み取ることができませんでした。

### 636

索引付けで作成された一時ファイルの 1 つを、読み取りアクセスによってオープンできませんでした。アクセス権限を調べてください。

### 659

索引付けで作成された一時ファイルの 1 つをオープンできませんでした。

### 660

索引付けで作成された一時ファイルの 1 つに書き込めませんでした。

### 661

索引付けで作成された一時ファイルの 1 つをクローズできませんでした。

### 662

索引付けで作成された一時ファイルの 1 つをオープンできませんでした。

### 663

索引付けで作成された一時ファイルの 1 つに書き込めませんでした。

### 664

索引付けで作成された一時ファイルの 1 つをクローズできませんでした。

### 665

索引付けで作成された一時ファイルの 1 つをオープンできませんでした。

**667**

索引付けで作成された一時ファイルの 1 つに書き込めませんでした。

**668-669**

文書が索引付けされていませんでした。文書内に存在するセクション・タグと文書モデル・ファイル内に定義されているセクション・タグが一致しないという問題が生じています。

**670-672**

文書は索引付けされていません。索引ファイルの 1 つのオープン、読み取り、書き込み、またはクローズができませんでした。索引に使用されているディスクのスペースに十分空きがあるか、およびアクセス権限が適正であるかをチェックしてください。

**673**

記憶域がありません (割り当てに失敗しました)。サーバーにメモリーが十分ではありません。作業負荷を減らしてください。

**674**

内部エラー。診断ログ内の情報を IBM 担当員に送ってください。

**675-687**

文書は索引付けされていません。索引ファイルの 1 つのオープン、読み取り、書き込み、またはクローズができませんでした。索引に使用されているディスクのスペースに十分空きがあるか、およびアクセス権限が適正であるかをチェックしてください。

**688、690**

記憶域がありません (割り当てに失敗しました)。サーバーにメモリーが十分ではありません。作業負荷を減らしてください。構成ファイル内の値を小さくしてみてください。

**689**

内部エラー。診断ログ内の情報を IBM 担当員に送ってください。

**691-695**

文書は索引付けされていません。索引ファイルの 1 つのオープン、読み取り、書き込み、またはクローズができませんでした。索引に使用されているディスクのスペースに十分空きがあるか、およびアクセス権限が適正であるかをチェックしてください。

**696-707**

内部エラー。診断ログ内の情報を IBM 担当員に送ってください。

### 708

記憶域がありません (割り当てに失敗しました)。サーバーにメモリーが十分ではありません。作業負荷を減らしてください。構成ファイル内の値を小さくしてみてください。

### 709-718

文書は索引付けされていません。索引ファイルの 1 つのオープン、読み取り、書き込み、またはクローズができませんでした。索引に使用されているディスクのスペースに十分空きがあるか、およびアクセス権限が適正であるかをチェックしてください。

### 719-721

内部エラー。診断ログ内の情報を IBM 担当員に送ってください。

### 722、729

記憶域がありません (割り当てに失敗しました)。サーバーにメモリーが十分ではありません。作業負荷を減らしてください。構成ファイル内の値を小さくしてみてください。

### 730、732、733、735-738

文書は索引付けされていません。索引ファイルの 1 つのオープン、読み取り、書き込み、またはクローズができませんでした。索引に使用されているディスクのスペースに十分空きがあるか、およびアクセス権限が適正であるかをチェックしてください。

### 731、739-742、744-746、749、755-758、760-761、767

内部エラー。診断ログ内の情報を IBM 担当員に送ってください。

### 743、748、750-754、759、765-766、768-770

文書は索引付けされていません。索引ファイルの 1 つのオープン、読み取り、書き込み、またはクローズができませんでした。索引に使用されているディスクのスペースに十分空きがあるか、およびアクセス権限が適正であるかをチェックしてください。

### 747、763、764

記憶域がありません (割り当てに失敗しました)。サーバーにメモリーが十分ではありません。作業負荷を減らしてください。構成ファイル内の値を小さくしてみてください。

### 815

次の 2 つの原因が考えられます。

- 障害の元となった文書に使用される言語のサポートに必要なリソース・ファイルの 1 つが見つからない

- その文書が要求する言語を DB2 テキスト・エクステンダーがサポートしていない

**831**

文書は索引付けされていません。テキストが見つかりません。文書の長さが 0 バイトです。

**860**

ファイル・オープン・エラー。一部の辞書またはシソーラス・ファイルが見つかりません。リソース・パス内に辞書ファイルがあるかを調べてください。検索時にシソーラス・ファイルのパス情報を指定した場合、そのセッションおよびファイル名を調べてください。

**954-956**

内部エラー。診断ログ内の情報を IBM 担当員に送ってください。

**957-967**

文書は索引付けされていません。索引ファイルの 1 つのオープン、読み取り、書き込み、またはクローズができませんでした。索引に使用されているディスクのスペースに十分空きがあるか、およびアクセス権限が適正であるかをチェックしてください。

**1000**

ファイルをオープンするときにエラーが発生しました。アクセス権限をチェックしてください。

**1001**

ファイルを追加するときにエラーが発生しました。アクセス権限をチェックしてください。

**1002**

ファイルを読み取るときにエラーが発生しました。ファイルが壊れている可能性があります。

**1003**

ファイルに書き込むときにエラーが発生しました。ディスク・スペースおよびアクセス権限をチェックしてください。

**1005**

ファイルを読み取るときにエラーが発生しました (ファイル内に位置決めされています)。ファイルが壊れている可能性があります。

**1006**

一時ファイルの名前を変更するときにエラーが発生しました。アクセス権限を調べてください。

## エラー事象理由コード

### 1007

ファイルを作成するときにエラーが発生しました。アクセス権限をチェックしてください。

### 1008

ファイルを圧縮するときにエラーが発生しました。アクセス権限を調べてください。

### 1009

ファイルをクローズするときにエラーが発生しました。ファイルが壊れている可能性があります。

### 1010

指定された索引名は、すでに使用されています。他の索引名を使用してください。

### 1011

指定されたパスは、すでに使用されています。他の場所を使用してください。

### 1012

データおよび作業ディレクトリーに同一のパスが使用されています。他の場所を使用してください。

### 1013

指定された索引名は、無効です。索引名は大文字または数字でなければならず、8文字以下でなければなりません。

### 1014

ファイルをコピーするときにエラーが発生しました。アクセス権限およびディスク・スペースを調べてください。

### 1017

索引名が不明です。正しいつづりをチェックしてください。

### 1019

ファイルを削除するときにエラーが発生しました。アクセス権限をチェックしてください。このエラー・メッセージは「2次エラー」として発生することがあります。診断ファイルを表示して、直前のエラー項目から追加の情報が得られないかを調べてください。

### 1020

一般ファイル・エラー。アクセス権限をチェックしてください。

### 1070-1074

文書は索引付けされていません。指定されたこのコード・ページは、全般に無効であるか、またはアクセス中の索引に関して無効です。



**1085**

文書は索引付けされていません。索引ページの読み取り中にエラーが発生しました。

**1086**

文書は索引付けされていません。索引待ち行列が空です。

**1116-1117**

文書は索引付けされていません。サーバー・インスタンスの初期設定ファイルから得られる情報は処理できません。初期設定ファイル内の項目が有効であること、およびファイルがアプリケーションからアクセス可能であることを確認してください。

**1129**

どの文書も索引付けされていません。バックグラウンド処理の開始に失敗しました。

**1158**

ファイルの名前を変更するときにエラーが発生しました。アクセス権限およびディスク・スペースを調べてください。

**1162**

Ngram 索引の索引ファイルが破壊された可能性があります。

**1163、1164**

文書は予期しないエラーのために、索引付けされていません。

**1165**

文書は予期しないファイル終わり (EOF) 状態のために、索引付けされていません。

**1176**

Ngram 索引にこれ以上索引付けすることはできません。文書数に関してオーバーフロー状態になっています (項目の多いことによるオーバーフロー)。多数の削除を行っていたり、同一の文書を繰り返し更新している場合、EhwReorg を呼び出して問題の解決を試みてください。それ以外の場合、新規の文書に 2 番目の索引を使用することを検討してください。

**1177**

文書は索引付けされていません。Ngram インデクサーはその文書が大き過ぎると判断しました。

**1189**

文書は索引付けされていません。境界シーケンスに問題が発生しました (韓国語に特定)。

## エラー事象理由コード

### 1198-1200

文書は索引付けされていません。索引のアクセスに問題が発生しました。索引が破壊された可能性があります。

### 1201

文書は索引付けされていません。文書のコード・ページを索引に特定のコード・ページに変換することができません。このエラーが発生するのは、UTF8 コード・ページの Ngram 索引だけです。

### 1202

文書は索引付けされていません。文書内に無効なデータがあるため、文書のコード・ページを索引に特定のコード・ページに変換することができません。このエラーが発生するのは、UTF8 コード・ページの Ngram 索引だけです。

### 1500-1505

文書分析コンポーネントに問題があります。初期化できないか (LIBPATH および構成ファイルの内容を調べてください)、または内部問題に起因する障害が生じています。詳しくは診断ファイルを参照してください。

### 1904

文書は索引付けされていません。セクションが使用可能な索引のために、文書モデルにアクセスするときに問題が発生しました。アクセス権限およびファイルの存在を調べてください。

### 2000

文書は索引付けされていません。文書タイプがサポートされていません。ライブラリー・サービス Lib\_access\_doc は、無効な文書タイプを戻しました。

### 2001

文書は索引付けされていません。文書のデータ・ストリーム中で、一連の誤りフィールドが検出されました。

### 2002

文書は索引付けされていません。文書のデータ・ストリーム中で、誤りの構造化フィールドが検出されました。

### 2003

文書は索引付けされていません。DB2 テキスト・エクステンダー・テキスト書式の文書に許されるテキスト・セクションは 1 つだけです。

### 2005

文書は索引付けされていません。文書のデータ・ストリーム中に指定されている言語がサポートされていません。

**2006**

文書は索引付けされていません。文書のデータ・ストリーム中に指定されている CCSID がサポートされていません。

**2007**

ライブラリーまたは省略時規則が指定した予期された文書形式が正しくありません。文書のヘッダーの形式が正しくありません。省略時規則が特殊文書ヘッダーのついた文書であるかどうかをチェックし、その規則が正しくない場合には変更してください。

**2008**

文書にアクセスできなかったため、索引付けされませんでした。

**2009**

文書は使用中でアクセスできなかったため、索引付けされませんでした。

**2010**

文書は索引付けされていません。指定された CCSID が誤りです。

**2011**

文書が正しい IBM DCA RFT または FFT 文書でないため、索引付けされていません。ページの終わりは、文書の本体テキストの最後の制御ワードでなければなりません。

**2012**

文書が正しい IBM DCA RFT または FFT 文書でないため、索引付けされていません。構造化フィールドに含まれている長さ指定が誤りです。

**2013**

文書が正しい IBM DCA RFT または FFT 文書でないため、索引付けされていません。文書中に、誤りの制御ワードが検出されました。

**2014**

文書が正しい IBM DCA RFT または FFT 文書でないため、索引付けされていません。文書中に、誤りのマルチバイト制御ワードまたは構造化フィールドが検出されました。

**2015**

文書が正しい IBM DCA RFT または FFT 文書でないため、索引付けされていません。文書のパラメーターが重複して検出されました。

**2016**

文書が正しい IBM DCA RFT または FFT 文書でないため、索引付けされていません。空のテキスト単位が検出されました。

**2018**

文書が、サポートされていない形式になっているか、または文書の拡張子

## エラー事象理由コード

について DIT に “exclude” 項目があります。文書に索引付けを許可する拡張子があることをチェックしてください。

### 2020

文書は索引付けされていません。これは WordPerfect 文書でも WordPerfect ファイルでもありません。

### 2021

文書は索引付けされていません。これは WordPerfect ファイルですが、WordPerfect 文書ではありません。

### 2022

文書は索引付けされていません。このバージョンの WordPerfect はサポートされていません。

### 2023

文書は索引付けされていません。これは暗号化された WordPerfect ファイルです。文書は、暗号化しないで保管してください。

### 2026

フットノートまたはエンド・ノートで END\_TXT が起きました。WordPerfect ファイルに損傷がある可能性がありますので、チェックしてください。

### 2028

パーサーは非文書のテキストを戻しました。ファイルの内容、特に形式指定ワードに関する部分を調べてください。文書形式がサポートされているかを調べてください。自動形式認識が失敗した場合、正しいパーサーが呼び出されていることを確認してください。

### 2030

文書は索引付けされていません。これは、Microsoft Word ファイルでないか、サポートされていないバージョンの Word です。

### 2031

文書は索引付けされていません。Microsoft Word 文書で、予期しないファイル終わりが検出されました。

### 2032

文書は索引付けされていません。Microsoft Word 文書で、誤りの制御ワードが検出されました。

### 2033

文書は索引付けされていません。これは、*fastsave* オプションを指定して、複合書式で保管されています。*fastsave* オプションを除去して保管してください。

**2034**

文書は索引付けされていません。Microsoft Word 文書で、必要なフィールド終わりマークが欠落しています。

**2035**

文書は暗号化されています。Microsoft Word で暗号化なしで保管してください。

**2036**

これは Word Macintosh 版の文書です。処理できません。Word Windows 版形式で文書を保管してください。

**2037**

この Word 文書には、組み込み OLE オブジェクトが含まれています。

**2040**

文書が有効な ECTF ファイルでないため、索引付けされませんでした。

**2041**

文書は索引付けされていません。文書に含まれている .SO LEN 制御ワードには、数値の指定が続いていません。

**2042**

文書は索引付けされていません。文書に含まれている .SO LEN 制御ワードには正しくない数値の指定が続いています。この数値は 1 ~ 79 です。

**2043**

文書は索引付けされていません。 .SO DOC 制御ワードは 1 つしか指定できません。 ECTF 文書はそれぞれ別々のファイルに保管してください。

**2044**

文書は索引付けされていません。 .SO HDE 制御ワードには、開始タグと終了タグを続けなければなりません。

**2046**

文書は索引付けされていません。この文書では、.SO DOC 制御ワードの前にテキストが含まれています。

**2047**

文書は索引付けされていません。この文書では、.SO PID 制御ワードの前にテキストが含まれています。

**2048**

文書は索引付けされていません。開始タグの後の終了タグが欠落しています。

## エラー事象理由コード

### 2050

文書は索引付けされていません。 .SO HDE 制御ワードの後で、誤りのタグが検出されました。

### 2051

文書は索引付けされていません。 .SO 制御ワードの後で、行の終わりが検出されました。

### 2052

文書は索引付けされていません。 予期しないテキスト終結が検出されました。

### 2060

文書は索引付けされていません。 この文書は AmiPro 文書でないか、サポートされていないバージョンの AmiPro 文書です。

### 2061

文書は索引付けされていません。 AmiPro 文書中の制御ワードが長過ぎます。

### 2062

文書は索引付けされていません。 このバージョンの AmiPro はサポートされていません。 サポートされているのは、AmiPro Architecture 第 4 版だけです。

### 2063

AmiPro 様式用紙が索引付けされていません。

### 2064

文書は索引付けされていません。 誤りの文字セットが検出されました。 サポートされるのは Lotus 文字セット 82 (Windows ANSI) だけです。

### 2065

文書は索引付けされていません。 AmiPro 文書で、予期しないファイル終わりが検出されました。

### 2072

文書は暗号化されているため、走査できません。

### 2073

文書形式が不整合です。

### 2074

文書に “bad file” フラグ・ビット・セットがあります。

**2080**

文書は索引付けされていません。この文書は RTF 文書でないか、サポートされていないバージョンの RTF 文書です。

**2081**

文書は索引付けされていません。長過ぎる RTF 制御ワードが検出されました。

**2083**

文書は索引付けされていません。Macintosh コード・ページはサポートされていません。

**2084**

文書は索引付けされていません。この文書は RTF 文書ですが、このバージョンの RTF 文書はサポートされていません。サポートされているのは RTF 第 1 版だけです。

**2090**

文書は索引付けされていません。この HTML 文書に含まれるタグは、パーサーによって長すぎるとみなされます。

**2093**

文書は索引付けされていません。この XML 文書は、XML パーサーによって拒否されました。

**2100**

文書に損傷があるか、または他の理由で読み取れません。新しい共通解析プログラムが、問題を解決できませんでした。

**2101**

文書は空であるか、テキストが入っていないため、索引付けできません。文書に図形しか含まれていないかどうかをチェックしてください。

**2102**

文書は、パスワード保護されているか暗号化されているため、索引付けできません。

**2105**

文書タイプは認識されていますが、フィルターが使用できません。

**2106**

文書が空のため、索引付けできません。

**2107**

文書がオープンできないため、索引付けできません。文書アクセスをチェックしてください。

## エラー事象理由コード

### 2112

文書は実行可能ファイルであるため、索引付けできません。

### 2113

文書は圧縮されているため、索引付けできません。

### 2114

文書は図形であるため、索引付けできません。図形文書形式が受け入れ可能なテキストを戻す場合には、この文書形式を索引付け処理に組み込むように要求を出してください。

### 2120

ユーザー出口の出力ファイルが存在しないか、またはアクセスできません。新しい共通解析プログラムのバージョンが、問題を解決できませんでした。

### 2121

出力ファイルはオープンして読み取ることはできません、または空です。新しい共通解析プログラムのバージョンが、問題を解決できませんでした。

### 2122

ユーザー出口出力ファイルを使用しようとしたが、オブジェクトにファイル名が指定または設定されていません。

### 2130

ユーザー出口プログラムを実行することはできませんでした。PATH 環境変数により設定されたパスに、実行可能プログラムが見つかるかどうかを確認してください。トレースおよびダンプを作成して、環境 (errno) 戻りコードについての追加情報を入手してください。

### 2131

戻りコードが正しくなかったために、ユーザー出口プログラムは失敗しました。トレースおよびダンプを作成して、環境 (errno) 戻りコードについての追加情報を入手してください。



---

## 第3部 付録および後付け



---

## 特記事項

本書において、日本では発表されていない IBM 製品 (機械およびプログラム)、プログラミングまたはサービスについて言及または説明する場合があります。しかし、このことは、弊社がこのような IBM 製品、プログラミングまたはサービスを、日本で発表する意図があることを必ずしも示すものではありません。本書で、IBM ライセンス・プログラムまたは他の IBM 製品に言及している部分があっても、このことは当該プログラムまたは製品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。これらのプログラムまたは製品に代えて、IBM の知的所有権を侵害することのない機能的に同等な他社のプログラム、製品またはサービスを使用することができます。ただし、IBM によって明示的に指定されたものを除き、これらのプログラムまたは製品に関連する稼働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 および他社は、本書で説明する主題に関する特許権 (特許出願を含む)、商標権、または著作権を所有している場合があります。本書の提供は、これらの特許権、商標権、および著作権について、本書で明示されている場合を除き、実施権、使用权等を許諾することを意味するものではありません。実施権、使用权等の許諾については、下記の宛先に、書面にてご照会ください。

〒106-0032 東京都港区六本木 3 丁目 2-31  
AP 事業所  
IBM World Trade Asia Corporation  
Intellectual Property Law & Licensing

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。

本書に含まれる情報には、技術的に不正確なもの、または誤植が含まれる場合があります。これらに対する変更は、定期的に行われます。これらの変更は、資料の改訂版に含まれます。IBM は、本書で説明している製品、プログラムに対して、予告なく改良、変更を加える場合があります。

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、お客様になんら義務も負わせない適切な方法で、使用もしくは配布することがあります。

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  
Office of the Lab Director  
1150 Eglinton Ave. East  
North York, Ontario  
M3C 1H7  
CANADA

本プログラムに関する上記の情報は、適切な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。

本書に含まれるパフォーマンス・データは、制御された環境下で決定されています。したがって、その他の稼働環境で得られる結果とは、かなり異なる可能性もあります。一部の測定値は、開発中のシステムを使用している場合があり、これらの測定値が一般的に提供可能なシステムで同様の数値になることを保証するものではありません。さらに、一部の測定値が推定されたものもあります。実測値と異なる場合があります。本書のユーザーは、使用される特定の環境での該当データを確認してください。

IBM 以外の製品については、当該製品の提供者から直接、出版されている資料または一般公開されている情報から入手しました。IBM は、これらの製品についてはテストを行っておらず、これらの IBM 以外の製品に関する性能、互換性またはその他の主張について確認することはできません。IBM 以外の製品の機能に対する質問は、それぞれの製品提供者にお問い合わせください。

IBM の将来の方向性または意図については、予告なしに変更または中止する場合があります。IBM の目的および目標のみ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が、これは説明に具体性を与えるために記載されたものであり、それらの例には、個人、企業、ブランドの、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。それらの名前はすべて架空のものであり、また名称や住所が類似する企業が実在しても、それは偶然に過ぎません。

#### 著作権：

本書に含まれる情報には、サンプル・アプリケーション・プログラムがソース言語の形式で含まれており、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技法を示しています。お客様は、これらのサンプル・プログラムが書かれているオペレーティング・プラットフォームでアプリケーション・プログラミング・インターフェースが実行可能となるためのアプリケーション・プログラムを開発、使用、販売または配布もしくは転送する目的のためだけに、サンプル・プログラムを、IBM に対する別途料金を支払うことなく、複製、変更、配布または転送することができます。これらのサンプルは、すべての条件下で十分にテストを行っていません。したがって、IBM は、これらのプログラムの信頼性、実用性または機能について、いかなる保証も負いません。

サンプル・プログラムまたはその改変版の複製物には、全部複製か部分複製かを問わず、次の著作権表示を必ず行うものとします。

© (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\_. All rights reserved.

---

## 商標

以下は、IBM Corporation の商標です。

ACF/VTAM	IBM
AISPO	IMS/ESA
AIX	LAN Distance
AIXwindows	MVS
AnyNet	MVS/ESA
APPN	MVS/XA
AS/400	Net.Data
BookManager	OS/2
CICS	OS/390
C Set++	OS/400
C/370	PowerPC
DATABASE 2	QBIC
DataHub	QMF
DataJoiner	RACF
DataPropagator	RISC System/6000
DataRefresher	RS/6000
DB2	S/370
DB2 Connect	SP
DB2 Extenders	SQL/400
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView
DRDA	VisualAge
Extended Services	VM/ESA
FFST	VSE/ESA
First Failure Support Technology	VTAM
	WebExplorer
	WIN-OS/2

次のものは、他社の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。

Java、およびすべての Java 関連の商標およびロゴは Sun Microsystems, Inc. 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 がライセンスしている米国およびその他の国における登録商標です。

他の会社名、製品名およびサービス名等は、それぞれ各社の商標または登録商標です。





## 用語集

この用語集では、本書で使用するさまざまな用語と略語を定義します。探している用語が見つからない場合は、索引か、あるいは *Dictionary of Computing*, New York: McGraw-Hill, 1994 を参照してください。

### A

**アクセス関数 (access function).** 列に保管されるテキストのデータ・タイプを、DB2 テキスト・エクステンダーで処理できるタイプに変換する、ユーザー定義の関数。

**API.** アプリケーション・プログラミング・インターフェース (Application programming interface)。

**アプリケーション・プログラミング・インターフェース (application programming interface (API)).** アプリケーション・プログラムと DB2 テキスト・エクステンダー情報検索サービス間の汎用インターフェース。

### B

**ブール検索 (Boolean search).** 1 つまたは複数の言葉をブール演算子を使って結合する検索。

**境界検索 (bound search).** 韓国語文書における、ワード境界を区別した検索。

**ブラウズする (browse).** コンピューター・モニター上にテキストを表示すること。

**ブラウザー (browser).** コンピューターのモニター上にテキストを表示できるようにする DB2 テキスト・エクステンダー機能。

### C

**カタログ視点 (catalog view).** 管理目的のために DB2 テキスト・エクステンダーが作成するシステム表の表示。カタログ表示には、DB2 テキスト・エクステンダーが使用できる表と列に関する情報が含まれる。

**CCSID.** コード化文字セット識別子 (Coded Character Set Identifier)。

**コード・ページ (code page).** すべてのコード・ポイントに図形文字および制御機能の意味を割り当てたもの。たとえば、8 ビット・コードの場合、文字および意味を 256 のコード・ポイントへ割り当てたもの。

**コマンド行プロセッサ (command line processor).** DB2TX というプログラムで、次のような機能がある:

DB2 テキスト・エクステンダー・コマンドを入力できるようにする。

コマンドを処理する

結果を表示する

**共通索引表 (common-index table).** テキスト列が共通のテキスト索引を共有する DB2 表。複数索引表 (*multi-index table*) を参照。

**カウント (count).** シソーラスの中の用語のレベル数 (深さ) を指定するキーワード。このレベルは、指定された関係に基づく検索項目を拡張するために使用される。

### D

**データ・ストリーム (data stream).** API 機能から戻される情報で、(少なくとも 1 つの段落からなる) テキストで構成される。テキストには検索

対象の用語、および見付かった用語をそのテキスト内で強調表示するための情報が入っている。

**DB2 エクステンダー (DB2 Extender).** 従来の数値データや文字データ以外のデータ・タイプ (たとえばイメージ、オーディオ、ビデオ・データ) および複雑な構造の文書の保管や検索を可能にする、複数のプログラムからなるグループ。

**DBCS.** 2 バイト文字サポート (Double-byte character support)。

**辞書 (dictionary).** DB2 テキスト・エクステンダーがテキストの分析、索引付け、検索を行った、特定言語の文書を強調表示したりするのに使用する、言語関連の情報のコレクション。

**使用不可にする (disable).** 使用可能プロセス中に作成された項目を除去することにより、データベース、テキスト表、またはテキスト列を DB2 テキスト・エクステンダーで使用可能になる前の条件に復元すること。

**特殊タイプ (distinct type).** ユーザー定義特殊タイプ (*user-defined distinct type*) を参照。

**文書 (document).** テキスト文書 (*text document*) を参照。

**文書ハンドル (document handle).** ハンドル (*handle*) を参照。

**文書モデル (document model).** 文書が含むセクションに関する、文書の構造の定義。DB2 テキスト・エクステンダーは索引付けの際に、文書モデルにより文書中のセクションを認識する。文書モデルでは、各セクションを識別するマークアップ・タグをリストする。セクション照会に使用するためのセクション記述名を、それぞれのタグごとに指定できる。1 つの文書モデル・ファイルには、1 つまたは複数の文書モデルを指定できる。

## E

**使用可能にする (enable).** データベース、テキスト表、またはテキスト列を DB2 テキスト・エクステンダーで使用できるようにすること。

**環境変数 (environment variable).** DB2 テキスト・エクステンダー環境の値に対する省略時値を提供するために使用する変数。

**環境プロファイル (environment profile).** DB2 テキスト・エクステンダーの提供するスクリプト。環境変数の設定を含む。

**エスケープ文字 (escape character).** それ以降の文字が、解釈されないマスク文字 (*masking character*) であることを示す文字。

**拡張する (expand).** シソーラスから導出された用語を検索項目に追加すること。

**拡張突き合わせ (extended matching).** 検索項目と完全には一致していない用語を強調表示する辞書を使用するプロセス。

**エクステンダー (extender).** DB2 エクステンダー (*DB2 Extender*) を参照。

**外部ファイル (external file).** DB2 に制御される表中のセルの形式ではなく、オペレーティング・システムのファイル・システムに保管されるファイル形式のテキスト文書。

## F

**フィーチャー検索 (feature search).** 人名、地名、または団体名などの用語検索で、FEATURE\_EXTRACTION 索引付けオプションを使って作成される基本語形索引 (*linguistic index*) で行われる。

**ファイル・ハンドル (file handle).** ハンドル (*handle*) を参照。

**書式 (format).** 文書のタイプ、たとえば ASCII や WordPerfect。

**フリー・テキスト検索 (free-text search).** 検索項目が自由形式のテキストとして表される検索 (自由形式テキストとは、検索対象の事柄を自然言語で記述した句または文)。

**関数 (function).** アクセス関数 (*access function*) を参照。

**ファジー検索 (fuzzy search).** 検索項目と似たつづりの語を検出する検索。

## H

**ハンドル (handle).** テキスト文書を識別する 2 進値。次のものを含む:

文書 ID

関連する索引の名前と場所

その文書のテキスト情報

DB2 が制御しない外部ファイルに保管されている文書の場合、そのファイルのパスと名前。

ハンドルは、テキスト列が DB2 テキスト・エクステンダーで使用可能な場合に、そのテキスト列内のテキスト文書ごとに作成される。

**情報の強調表示 (highlighting information).** データ・ストリーム (*data stream*) を参照。

**混成検索 (hybrid search).** ブール検索 とフリー・テキスト検索 との組み合わせ。

## I

**索引付け (index).** テキストから有効な用語を抽出し、テキスト索引 に保管すること。

**索引特性 (index characteristics).** 以下のことを判別するテキスト索引 の特性。

索引を保管するディレクトリー

索引のタイプ

索引を更新する頻度

索引を最初に更新する時点

**索引タイプ (index type).** テキスト索引 の特性で、その索引が文書内の用語を実際形で含むか、それとも言語処理した形で含むかを決定する。一致語句索引 (*precise index*)、基本語形索引 (*linguistic index*)、および Ngram 索引 (*Ngram index*) も参照。

**初期化ハンドル (initialized handle).** 前もって準備され、テキスト形式かテキスト言語 (またはその両方) だけを含むハンドル。

**インスタンス (instance).** DB2 テキスト・エクステンダーの論理的な環境。同一ワークステーション上に DB2 テキスト・エクステンダーのインスタンスを複数持つことができる。ただし、個々の DB2 インスタンスには 1 つのインスタンスしか持てない。これらのインスタンスを使用すると、次のことが可能である:

開発環境と実稼働環境を分ける

機密情報へのアクセスを特定の人々のみに制限する

**インスタンス変数 (instance variable).** インスタンス所有者 の名前、またはインスタンス所有者のホーム・ディレクトリーの名前の省略時値を提供する変数。

## L

**言語 (language).** 索引付け、検索、およびブラウズの際に使用されるディクショナリー (*dictionary*) の名前。

**基本語形索引 (linguistic index).** 言語的な処理により基本語形になった用語を含むテキスト索引。たとえば “Mice” (ねずみの複数形) は、“mouse” (単数形) として索引付けされる。一致語句索引 (*precise index*)、および Ngram 索引 (*Ngram index*) も参照。

**論理ノード (logical node).** 同じ物理マシンに他のノードと共に割り当てられるノード。物理ノード (*physical node*) も参照。

**ログ表 (log table).** テキスト文書の索引付けに関する情報を含む、DB2 テキスト・エクステンダーが作成する表。使用可能なテキスト列内の文書が追加されるたびに、この情報をログ表に保管するには、トリガー を使用する。

## M

**マスキング文字 (masking character).** 検索項目の前部、中央部、および後部で任意指定文字を表すのに使用する文字。マスキング文字は通常、一致語句索引でさまざまな用語を検索するのに使用される。

**突き合わせ、一致 (match).** テキスト文書内の検索項目のオカレンス。

**複数索引表 (multi-index table).** テキスト列が個別のテキスト索引を持つ DB2 表。共通索引表 (*common-index table*) を参照。

## N

**Ngram 索引 (Ngram index).** テキスト索引 (*text index*) の 1 つで、DBCS 文書をサポートし、SBCS 文書のあいまい検索をサポートする。基本語形索引 (*linguistic index*)、および一致語句索引 (*precise index*) も参照。

**ノード (node).** 区分データベース 環境のサーバー。論理ノード (*logical node*)、物理ノード (*physical node*)、およびノードグループ (*nodegroup*) も参照。

**ノードグループ (nodegroup).** 1 つまたは複数のデータベース区画サーバーのサブセット。物理的に別個のマシンに割り当てられるノード。論理ノード (*logical node*) も参照。

## O

**オカレンス (occurrence).** 突き合わせ、一致 (*match*) の同義語。

## P

**区分データベース (partitioned database).** 複数の部分で構成されるデータベース。それぞれの部分は別個のデータベース区画サーバーにより保守される。

**周期的索引付け (periodic indexing).** 日付、時間、分で表される既定の時間間隔、および索引付けを行う前にそのログ表 にリストする必要のある文書名の最小数を指定した索引付け。

**物理ノード (physical node).** 物理的に別個のマシンに割り当てられるノード。論理ノード (*logical node*) も参照。

**一致語句索引 (precise index).** 抽出されたテキスト文書内にあった用語とまったく同じ用語が入っているテキスト索引。基本語形索引 (*linguistic index*)、および Ngram 索引 (*Ngram index*) も参照。

**プロフィール (profile).** 環境プロフィール (*environment profile*) を参照。

## R

**ランク (rank).** タイプ DOUBLE の 0 と 1 の間の絶対値で、ある文書が、見つかった他の文書と比べてどの程度検索基準に合致しているかを示す。この値は文書内で検出された一致数と、その文書サイズを示す。

**詳細化 (refine).** 前の検索の検索基準を後の検索基準に追加して、一致 数を減らすこと。

**検索する (retrieve).** DB2 テキスト・エクステンダーの検索関数の 1 つに検索引き数を使用してテキスト文書を検出すること。

## S

**SBCS.** 1 バイト文字サポート (Single-byte character support)。

**検索引き数 (search argument).** 検索の際に指定される条件で、1 つまたは複数の検索項目と検索パラメーターから成る。

**シェル・プロファイル (shell profile).** 環境プロフィール (*environment profile*) を参照。

**停止ワード (stop word).** テキスト文書 (*text document*) において、テキスト索引 (*text index*) から除外され、検索引き数 (*search argument*) に含まれていても無視される一般的な語 (たとえば “before”)。

## T

**テキスト列 (text column).** テキスト文書 を含んでいる列。

**テキスト構成 (text configuration).** 索引、テキスト、および処理の値の省略時設定。

**テキスト文書 (text document).** DB2 表の中に保管される、タイプ CHAR、GRAPHIC、VARGRAPHIC、LONG VARGRAPHIC、DBCLOB、VARCHAR、LONG VARCHAR、または CLOB のテキスト。

**テキスト索引 (text index).** テキスト文書から抽出された、重要な用語の集まり。それぞれの用語は、抽出元の文書と関連付けられる。文書そのものではなくこの索引を検索することによって、検索時間を大幅に改善することができる。一致語句索引 (*precise index*)、および基本語形索引 (*linguistic index*) も参照。

**テキスト情報 (text information).** 以下のことを記述するテキスト文書 の特性。

CCSID  
書式

## 言語

**テキスト表 (text table).** テキスト列 を含む DB2 表。

**トレース処理 (tracing).** エラーの原因を追及するために後で使用される情報をファイルに保管する処置。

**トリガー (trigger).** テキスト列から文書が追加、変更、削除されるたびに、ログ表 への索引付けに必要な文書に関する情報を自動的に追加する機構。

## U

**UDF.** ユーザー定義関数 (User-defined function)。

**UDT.** ユーザー定義特殊タイプ (user-defined distinct type)。

**更新頻度 (update frequency).** テキスト索引が更新される頻度。日付、時間、分、および索引付けを行う前にログ表 にリストする必要のある文書名の最小数で表される。

**ユーザー定義特殊タイプ (user-defined distinct type (UDT)).** DB2 のユーザーが作成するデータ・タイプ。LONG VARCHAR など、DB2 に備えられているデータ・タイプと対比される。

**ユーザー定義関数 (user-defined function (UDF)).** DB2 のユーザーが作成する SQL 関数。DB2 に備えられている SQL 関数と対比される。DB2 テキスト・エクステンダーに備えられている、CONTAINS などの管理および検索関数は、UDF の書式である。

## W

**ワイルドカード文字 (wildcard character).** マスキング文字 (*masking character*) を参照。



## 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]

アクセス関数

説明 77

ENABLE TEXT COLUMN 中の  
149

アプリケーション・プログラミング  
インターフェース (API)

(Application programming interface  
(API))

一致情報の入手 128

解説 235

概説 124

概要 235

強調表示情報 244

検索関数 123

検索結果表の入手 126

情報強調表示のためにポインター  
を入手する 244

テキストの検索 126

ブラウズ関数 123

ブラウズ情報記憶域の解放 130

ブラウズ情報の入手 127

ブラウズする文書のオープン  
128

ブラウズ・セッションの開始  
127

ブラウズ・セッションの終了  
129

プログラムの例 261

文書のクローズ 129

メッセージ 309

戻りコード 299

DesCloseDocument 関数 237

DesEndBrowseSession 関数 238

アプリケーション・プログラミング  
インターフェース (API)

(Application programming interface  
(API)) (続き)

DesFreeBrowseInfo 関数 239

DesGetBrowseInfo 関数 240

DesGetMatches 関数 244

DesGetSearchResultTable 関数  
250

DesOpenDocument 関数 255

DesStartBrowseSession 関数 258

一致語句検索 48

一致語句索引

検索オプションの省略時値 228

説明 48

インスタンス

環境変数 19

作成 23

除去 23

リスト用のコマンド、

TXILIST 182

インストール検査 18

インターフェース・サンプル 31

ウィザード・サンプル

説明 31

TXWIZARD コマンド 201

エクステンダー

使用例 4

エスケープ文字

構文 232

使用法 95

エラー事象

記録 74

削除 110

表示 115

理由コード 331

DELETE INDEX EVENTS 142

GET INDEX STATUS コマンド  
168

オープン、文書の、API 関数

使用法 128

オープン、文書の、API 関数 (続き)  
説明 255

オカレンス、検索項目 216

### [力行]

開始、ブラウズ・セッションの、API  
関数

使用法 127

説明 258

開始、DB2 テキスト・エクステンダー  
・サーバーの 188

外部ファイル

索引更新の考慮事項 108

使用可能化 79

使用不可にする 121

ハンドル 88

ハンドルからのパス / 名前の抽出  
104

ハンドルでの初期化 212

ハンドル内でのパス / 名前の変更  
104

ハンドル内のファイル名を入手ま  
たは変更する 208

DISABLE TEXT FILES コマンド  
145

ENABLE TEXT FILES コマンド  
157

FILE 関数 208

解放、ブラウズ情報用の記憶域の、  
API 関数

使用法 130

説明 239

拡張、強調表示のための用語 273

拡張突き合わせ 274

カタログ表示

削除 122

作成 68

内容 118

各国語サポート 42

環境、クライアント / サーバー 4

環境変数 19

環境変数 19 (続き)

説明 19

表示 112

GET ENVIRONMENT コマンド

165

関数

解説 203

概説 204

関数パス 89

検索関数 83

検索パフォーマンスの向上 105

検索引き数の指定 92

説明 83

直前の検索の詳細化 101

データ・タイプの変換用 77

テキストの検索 89

により戻される SQL の状態

309

ハンドル内の情報の設定と抽出

103

API 関数 123

CCSID 206

CONTAINS 207

DB2 テキスト・エクステンダー

関数のパスの設定 89

FILE 208

FORMAT 209

HANDLE 210

HANDLE\_LIST 211

INIT\_TEXT\_HANDLE 212

LANGUAGE 214

NO\_OF\_DOCUMENTS 215

NUMBER\_OF\_MATCHES 216

RANK 217

REFINE 218

SEARCH\_RESULT 219

SET CURRENT FUNCTION

PATH ステートメント 89

管理

外部ファイルの索引の更新 108

環境変数 19

クライアント・コマンドの要約

133

コマンド行プロセッサ 134

サーバー状況の表示 24

サーバー・コマンドの要約 177

管理 (続き)

索引事象の削除 110

索引状況の表示 113

索引状況のリセット 109

索引設定の表示 116

索引設定の変更 108

索引の再編成 110

索引の即時更新 107

作成、DB2 テキスト・エクステ

ンダー・インスタンスの 23

サンプル・データベース、作成用

ユーティリティ 18

サンプル・データベースの作成

200

シソーラス定義ファイルのコンパ

イル 191

状況情報の入手 111

省略語ファイル、編集 60

データベース、表、および列の状

況の表示 111

データベースを使用不可にする

122

停止ワードと省略語ファイルの変

更 60

停止ワードファイル、変更 60

テキスト構成の変更 22

テキスト索引の保守 107

テキスト情報設定の表示 117

テキスト表を使用不可にする

121

テキスト列を使用不可にする

120

テキスト・ファイルを使用不可に

する 121

トレース障害 26

バックアップおよび復元 25

CHANGE INDEX SETTINGS コ

マンド 135

CHANGE TEXT

CONFIGURATION コマンド

137

CONNECT コマンド 140

DB2 テキスト・エクステンダ

ー・インスタンスの除去 23

DB2 テキスト・エクステンダ

ー・サーバーの開始 188

管理 (続き)

DB2 テキスト・エクステンダ

ー・サーバーの停止 190

DB2TX コマンド 134

DELETE INDEX EVENTS 142

DELETE INDEX EVENTS コマン

ド 142

DISABLE DATABASE コマンド

143

DISABLE TEXT COLUMN コマ

ンド 144

DISABLE TEXT FILES コマンド

145

DISABLE TEXT TABLE コマン

ド 146

GET ENVIRONMENT コマンド

165

GET INDEX SETTINGS コマンド

166

GET INDEX STATUS コマンド

168

GET STATUS コマンド 169

GET TEXT CONFIGURATION コ

マンド 170

GET TEXT INFO コマンド 171

Ngram シソーラス定義ファイルの

コンパイル 193

QUIT コマンド 172

REORGANIZE INDEX コマンド

173

RESET INDEX STATUS コマン

ド 174

TXICRT コマンド 179

TXIDROP コマンド 181

TXILIST コマンド 182

TXIPCLEAN コマンド 183

TXNADD コマンド 184

TXNCHECK コマンド 185

TXNDROP コマンド 186

TXSAMPLE コマンド 187

TXSTART コマンド 188

TXSTATUS コマンド 189

TXSTOP コマンド 190

TXTHESC コマンド 191

TXTHESN コマンド 193

TXTRACE コマンド 195



管理 (続き)

- TXVERIFY コマンド 200
- TXWIZARD コマンド 201
- UPDATE INDEX コマンド 175

規則、検索引き数の 232

基本形への変形、用語の 269

基本テキスト分析

- 強調表示 273
- 正規化 265
- 非英数字文字を含む用語 264
- 文の認識 266
- 用語の索引付け 264

境界検索、例 99

強調表示情報

- データ・ストリーム 128
- データ・ストリーム構文 244

共通索引テーブル

- 作成 69
- 説明 49, 50
- ENABLE TEXT TABLE コマンド 160

区分データベース・サポート 7

組み込みファイル des\_ext.h 123

クライアント / サーバー環境 4

クローズ、文書の、API 関数

- 使用法 129
- 説明 237

結果表 252

言語、テキスト文書の

- 検索引き数 97
- 説明 39
- テキスト構成内の省略時値 21
- ハンドルからの抽出 104
- ハンドルでの初期化 212
- ハンドル内での変更 104
- リスト 39
- GET TEXT INFO コマンド 171
- LANGUAGE 関数 214

言語索引

- 検索オプションの省略時値 228
- 説明 46

言語的な処理

- 拡張突き合わせ 274
- 基本テキスト分析 264
- 検索のための 271
- 語のマスクング 273

言語的な処理 (続き)

- 索引作成時 263
- 説明 263
- 停止ワードのフィルター処理 270
- 同義語 272
- 発音の拡張 273
- 複合語の分割 270
- ブラウズ 273
- マスクング 273
- 文字のマスクング 273
- 用語拡張 273
- 用語の基本形への変形 269

言語パラメーター、リスト 59

検索、言語的な処理 271

検索、テキストの

- 概説 89
- 検出された一致数の入手 91
- 検出文書のランクの入手 91
- 検出文書のリスト作成 211
- 構文 223
- 照会の作成 90
- 直前の検索の詳細化 101
- パフォーマンスの向上 105
- プログラムの例 261
- API の使用 126
- REFINE 関数 218
- SEARCH\_RESULT 関数 219
- 検索エンジンからの理由コード 327
- 検索エンジンの理由コード 327

検索関数 123

検索状況、表示

- 構文 168
- 例および出力 113

検索状況、リセット

- 構文 174
- 例 109

検索引き数

- 同じ段落中の用語の検索 95
- 同じ文中の用語の検索 95
- 規則と制約事項の要約 232
- 境界検索 99
- 構文 223
- 混成検索 100
- さまざまな言語での用語の検索 97

検索引き数 (続き)

- シソーラスの検索 99
- 指定 92
- 説明 221
- 属性値 225
- 同義語の検索 96
- 任意の順序での用語の検索 95
- 発音の似た語の検索 99
- ファジー検索 98, 228
- 複数の用語の検索 92
- フリー・テキスト検索 100
- 文書セクションでの用語の検索 96
- マスクング文字の使用 93
- 用語の一部の検索 93
- 用語の変化形の検索 93
- AND キーワード 228
- BOUND キーワード 228
- COUNT キーワード 226
- EXPAND キーワード 227
- FUZZY FORM OF キーワード 228
- IN SAME PARAGRAPH AS 228
- IN SAME SENTENCE AS 228
- MODEL キーワード 225
- NOT キーワード 227
- NOT を使用した検索 98
- PRECISE FORM OF キーワード 228
- RESULT LIMIT キーワード 226
- SECTION キーワード 225
- STEMMED FORM OF キーワード 228
- SYNONYM FORM OF キーワード 228
- TERM OF キーワード 227
- THESAURUS キーワード 226
- & (AND) 演算子 227
- & および | を使用した検索 92
- | (OR) 演算子 227

更新、テキスト索引の

- 定期的に行う 51
- 頻度の変更 109
- CHANGE INDEX SETTINGS 中の UPDATEFREQ 136

更新、テキスト索引の (続き)

CHANGE TEXT  
CONFIGURATION 中の  
UPDATEFREQ 138

更新状況、表示

構文 168  
例および出力 113

更新状況、リセット

構文 174  
例 109

更新頻度

構文 51  
説明 51  
テキスト構成設定内の省略時値  
22  
変更 109

CHANGE INDEX SETTINGS 中  
の UPDATEFREQ 136

CHANGE TEXT  
CONFIGURATION 中の  
UPDATEFREQ 138

GET INDEX SETTINGS コマンド  
166

構成 19

構成テーブル

作成 68  
表示 113

CHANGE TEXT  
CONFIGURATION コマンド  
137

GET TEXT CONFIGURATION コ  
マンド 170

構成ファイル 289

語の分割 48

コマンド

クライアント・コマンドの要約  
133

サーバー・コマンドの要約 177

CHANGE INDEX  
SETTINGS 135

CHANGE TEXT  
CONFIGURATION 137

CONNECT 140

DB2TX 134

DELETE INDEX EVENTS 142

DISABLE DATABASE 143

コマンド (続き)

DISABLE TEXT COLUMN 144

DISABLE TEXT FILES 145

DISABLE TEXT TABLE 146

ENABLE DATABASE 147

ENABLE TEXT COLUMN 148

ENABLE TEXT FILES 157

ENABLE TEXT TABLE 160

GET ENVIRONMENT 165

GET INDEX SETTINGS 166

GET INDEX STATUS 168

GET STATUS 169

GET TEXT  
CONFIGURATION 170

GET TEXT INFO 171

QUIT 172

REORGANIZE INDEX 173

RESET INDEX STATUS 174

TXICRT 179

TXIDROP 181

TXILIST 182

TXIPCLEAN 183

TXNADD 184

TXNCHECK 185

TXNDROP 186

TXSAMPLE 187

TXSTART 188

TXSTATUS 189

TXSTOP 190

TXTHESC 191

TXTHESN 193

TXTRACE 195

TXVERIFY 200

TXWIZARD 201

UPDATE INDEX 175

コマンド行プロセッサ

開始 65

ヘルプ 66

DB2TX コマンド 134

QUIT コマンド 172

混成検索、例 100

コンパイル、シソーラス定義ファ  
イルの 191

コンパイル、Ngram シソーラス定義  
ファイルの 193

## [サ行]

サーバー

開始 188

状況の表示 189

設定および保守 23

停止 190

トレース障害 26

TXICRT コマンド 179

TXIDROP コマンド 181

TXILIST コマンド 182

TXIPCLEAN コマンド 183

TXNADD コマンド 184

TXNCHECK コマンド 185

TXNDROP コマンド 186

TXSAMPLE コマンド 187

TXSTART コマンド 188

TXSTATUS コマンド 189

TXSTOP コマンド 190

TXTRACE コマンド 195

TXVERIFY コマンド 200

TXWIZARD コマンド 201

索引

一致語句 48

概説 33

外部ファイルの更新 108

基本語形 46

共通索引表 49

計画 33

現行設定値の表示 116

現行設定値の変更 108

更新頻度 51

更新頻度の変更 109

サイズの計算 50

再編成 110

索引タイプの変更 49

作成、テキスト列に対するさまざ  
まな索引タイプ 75

周期的な索引更新 51

即時の索引更新 107

タイプ 45

テキスト構成設定内の省略時タイ  
プ 22

テキスト構成の変更 22

バックアップおよび復元 25

複数、使用 49

索引 (続き)

保守 107  
CASE\_ENABLED オプション 49  
CHANGE INDEX SETTINGS コマンド 135  
CHANGE TEXT  
  CONFIGURATION コマンド 137  
CHANGE TEXT  
  CONFIGURATION 中の INDEXTYPE 138  
CHANGE TEXT  
  CONFIGURATION 中の TABLESPACE 138  
ENABLE TEXT COLUMN 中の INDEXOPTION 152  
ENABLE TEXT COLUMN 中の INDEXTYPE 152  
ENABLE TEXT TABLE 中の INDEXOPTION 161  
ENABLE TEXT TABLE 中の INDEXTYPE 161  
GET INDEX SETTINGS コマンド 166  
GET INDEX STATUS コマンド 168  
GET TEXT CONFIGURATION コマンド 170  
Ngram 48  
UPDATE INDEX コマンド 175  
索引状況、表示  
  構文 168  
  例および出力 113  
索引状況、リセット  
  構文 174  
  例 109  
索引タイプ、変更  
  作成、テキスト列に対するさまざまな索引タイプ 75  
  変更 49  
索引付け、言語的な処理 263  
索引付け事象  
  理由コード 331  
索引付け事象、削除  
  構文 142  
  例 110

索引特性

テキスト構成設定内の省略時値 21  
表示 116  
ENABLE TEXT COLUMN 中の 148  
ENABLE TEXT FILES 中の 157  
ENABLE TEXT TABLE 中の 160  
作成、サンプル・データベースの 200  
作成、サンプル・テーブルの TXSAMPLE コマンド 187  
作成、DB2 テキスト・エクステンダー・インスタンスの TXICRT コマンド 179  
サポートされる浮動形式 55  
サンプル API プログラム 261  
サンプルの DB2 テキスト・エクステンダー関数  
  実行 84  
  サンプル・ウィザード 31  
  サンプル・データベース作成 (TXVERIFY) 18  
  サンプル・テーブル削除 122  
  説明 84  
  TXSAMPLE コマンド 187  
シェル・プロファイル 19  
時刻形式、セクション・サポート 55  
事象理由コード 331  
辞書ファイル名 59  
シソーラスの検索  
  概念 275  
  構文 226  
  シソーラス定義ファイルのコンパイル 191  
  シソーラスの作成 280  
  例 99  
Ngram シソーラス定義ファイルのコンパイル 193  
Ngram シソーラスの作成 285  
THESAURUS キーワード 226  
TXTTHESC コマンド 191  
TXTTHESN コマンド 193

終了、ブラウズ・セッションの、API 関数

  使用法 129  
  説明 238  
障害の検出 26  
状況、索引の  
  エラー後のリセット 109  
  現在状況の表示 168  
  表示 113  
  リセット 109  
状況、DB2 テキスト・エクステンダー・サーバーの 189  
情報、テキスト文書に関する  
  言語 39  
  現行設定の表示 117  
  書式 35  
  タイプ 35  
  CCSID 39  
  GET TEXT INFO コマンド 171  
省略語  
  省略語ファイルの編集 60  
  リスト 59  
初期設定、ハンドルの  
  方法 103  
  INIT\_TEXT\_HANDLE 関数 212  
除去、インスタンスの  
  方法 23  
  TXIDROP コマンド 181  
書式、テキスト文書の 35  
  サポートされていないものを変換 37  
  サポートされるもののリスト 35  
  説明 35  
  テキスト構成内の省略時値 21  
  ハンドルからの抽出 104  
  ハンドルでの初期化 212  
  ハンドル内での変更 104  
CHANGE TEXT  
  CONFIGURATION 中の 139  
ENABLE TEXT COLUMN 中の 151  
FORMAT 関数 209  
FORMAT キーワード 139, 151  
GET TEXT INFO コマンド 171

## 処理特性

- テキスト構成設定内の省略時値 22
- スペース所要量、索引の 50
- 正規化、用語の 265
- 整数形式、セクション・サポート 56
- 制約事項、検索引き数の 232
- セクション、文書中の
  - 検索構文中の MODEL キーワード 225
  - 検索構文中の SECTION キーワード 225
  - 検索構文中の属性値 225
  - 検索例 96
  - サポートされる時刻形式 55
  - サポートされる整数形式 56
  - サポートされる日付形式 54
  - サポートされる浮動形式 55
  - セクション・サポートの使用可能化 53
  - 属性セクション 54
  - フラット・ファイル文書 56
  - 文書モデル・ファイル、内容 57
  - DESMODEL.INI 53
  - HTML 文書 56
  - XML 文書 58
- 接続、データベースへの
  - 方法 67
  - CONNECT コマンド 140

## [タ行]

- タイプ、テキスト索引の
  - 一致語句 48
  - 基本語形 46
  - 検索オプションの省略時値 228
  - テキスト構成設定内の省略時値 22
- CASE\_ENABLED オプション 49
- CHANGE TEXT
  - CONFIGURATION コマンド 138
- CHANGE TEXT
  - CONFIGURATION 中の INDEXTYPE 138

- タイプ、テキスト索引の (続き)
  - ENABLE TEXT COLUMN 中の INDEXTYPE 152
  - ENABLE TEXT TABLE 中の INDEXTYPE 161
  - GET INDEX SETTINGS コマンド 166
  - Ngram 48
- 多重区画ノードグループ 7
- 直前の検索の詳細化 101
- 突き合わせ、一致
  - 検索結果 91
  - DesGetMatches 関数 244
  - DesGetSearchResultTable 126
  - NUMBER\_OF\_MATCHES 関数 216
- 突き合わせ、拡張 274
- データベース
  - 使用可能化 68
  - 状況情報の表示 111
  - 使用不可にする 122
  - 接続 67
  - バックアップおよび復元 25
  - CONNECT コマンド 140
  - DISABLE DATABASE コマンド 143
  - ENABLE DATABASE コマンド 147
  - GET STATUS コマンド 169
- データ・ストリーム構文 245
- データ・タイプ、テキスト文書の
  - サポートされる 149
  - 変換用の関数 77, 78
- DATALINK データ・タイプを含む列の使用可能化 78
- テーブル・スペース 74
- テーブル・スペースおよびノードグループ 74
- 停止、DB2 テキスト・エクステンダー・サーバーの 190
- 停止ワード
  - 基本テキスト分析の一部として 270
  - 説明 33
  - 停止ワードファイルの編集 60
  - リスト 59

- ディスク・スペース、索引の 50
  - ディレクトリー、索引用の
    - GET INDEX SETTINGS コマンド 166
  - テキスト構成設定値
    - インストール時の省略時値 21
    - 表示 113
    - 変更 22
  - テキスト特性
    - 言語 39
    - 書式 35
    - テキスト構成内の省略時値 21
    - CCSID 39
    - ENABLE TEXT COLUMN 中の 148
    - ENABLE TEXT FILES 中の 157
  - テキスト・テーブル
    - バックアップおよび復元 25
    - ラージ表における列の使用可能化 76
  - DISABLE TEXT TABLE コマンド 146
  - ENABLE TEXT TABLE コマンド 160
  - 同義語
    - 検索引き数 96
    - 説明 272
    - SYNONYM FORM OF キーワード 228
  - 特殊タイプ 203
  - トリガー
    - 作成 74
    - 説明 34
  - トレース障害
    - セットアップ 26
    - TXTRACE コマンド 195
- ## [ナ行]
- 入手、一致語の、API 関数
    - 使用法 128
    - 説明 244
  - 入手、検索結果テーブルの、API 関数
    - 使用法 126
    - 説明 250

入手、ブラウザ情報の、API 関数

    使用法 127

    説明 240

認識、文の 266

ノード

    ノードグループおよびテーブル・  
    スペース 74

    ノードの除去コマンド

        TXNDROP 186

    ノードのチェック・コマンド

        TXNCHECK 185

    ノードの追加コマンド

        TXNADD 184

    マルチノード環境 52

    割り当て 7

## [ハ行]

はじめに 29

発音の拡張

    説明 273

    例 99

バックアップおよび復元 25

パフォーマンスの向上 105

ハンドル

    外部ファイルの 88

    情報の設定と抽出 103

    初期化 103

    書式と言語の変更 104

    説明 87

    特殊タイプ DB2TEXTFH 203

    特殊タイプ DB2TEXTH 203

    ハンドルのリストからの入手  
    210

    リストを使用したパフォーマンス  
    の向上 105

    CCSID 関数 206

    CCSID、書式、および言語の抽出  
    104

    FORMAT 関数 209

    LANGUAGE 関数 214

ハンドル・リストのポインター (特殊  
タイプ DB2TEXTFHLISTP) 204

ハンドル・リストのポインター (特殊  
タイプ DB2TEXTHLISTP) 204

日付形式、セクション・サポート  
54

ブール演算子

    NOT 98

    & (AND) および | (OR) 92

ブール検索引き数 227

ファジー検索 46

ファジー検索、例 98

深さを指定する、シソーラスに含ま  
れる用語の 226

複合語の分割 270

複数索引の使用 49

物理ノード、割り当て 7

ブラウザ

    言語的な処理 273

    独自のブラウザーの使用 127

    プログラムの例 261

ブラウザ関数 123

フラットな ASCII、文書形式 35

フラット・ファイル文書、セクショ  
ン・サポート 56

フリー・テキスト検索

    例 100

分解、複合語の 270

文書

    関連情報 117

    形式の変換 37

    言語 39

    構造 53

    索引付け 33

    サポートされる形式 35

    サポートされるデータ・タイプ  
    149

    書式、説明 35

    設定値の表示 117

    データ・タイプの変換 77

    CCSID 39

    CHANGE TEXT

        CONFIGURATION 中の形式  
        139

    ENABLE TEXT COLUMN 中の書  
    式 151

    GET TEXT INFO コマンド 171

文書の構造

    検索構文中の MODEL キーワー  
    ド 225

文書の構造 (続き)

    検索構文中の SECTION キーワー  
    ド 225

    検索構文中の属性値 225

    検索例 96

    セクション・サポートの使用可能  
    化 53

文書モデル

    検索構文中の MODEL キーワー  
    ド 225

    検索構文中の SECTION キーワー  
    ド 225

    検索構文中の属性値 225

    説明 53

    文書モデル・ファイルの変更 53

文書モデル・ファイル、内容 57

分析、テキストの

    索引付け 264

    ブラウザ 273

文の認識 266

文の分割 48

並行処理概念 7

ヘッダー・ファイル des\_ext.h 123

ヘルプ、コマンドの 66

変形、用語の基本形への 269

変数

    環境変数の説明 19

    環境変数の表示 112

    GET ENVIRONMENT コマンド

    165

## [マ行]

マスキング

    言語的な処理 273

    検索項目内での 93

マルチノード環境 52

メッセージ 309

文字のマスキング 273

戻りコード 299

## [ヤ行]

ユーザー出口、文書形式の変換 37

用語拡張、強調表示のための 273

## [ラ行]

ラージ表、使用可能化 76

ランク

検索結果 91

DesGetSearchResultTable 126

列

さまざまな索引タイプの使用可能化 75

使用可能化 72

使用不可にする 120

ラージ表における使用可能化 76

DISABLE TEXT COLUMN コマンド 144

ENABLE TEXT COLUMN コマンド 148

ログ用スペース、使い尽くす 76

ログ・テーブル

エラー事象の抽出 115

作成 74

説明 34

テーブル・スペースへの割り当て 74

論理ノード、割り当て 7

## [ワ行]

ワイルドカード文字

検索項目内での 93

語のマスキング 273

## A

AmiPro、文書形式 35

AND

検索引き数内のキーワード 228

ブール演算子 92

ASCII、文書形式 35

## B

BOUND キーワード 228

## C

CASE\_ENABLED キーワード

ENABLE TEXT COLUMN 中の  
152

CASE\_ENABLED キーワード (続き)

ENABLE TEXT TABLE 中の  
161

CCSID

関数 206

コード・ページの問題の回避 42  
説明 39

テキスト構成内の省略時値 21

ハンドルからの抽出 104

ハンドルでの初期化 212

リスト 39

CHANGE TEXT

CONFIGURATION 中の 139

ENABLE TEXT COLUMN 中の  
150

GET TEXT INFO コマンド 171

CHANGE INDEX SETTINGS コマンド

構文 135

使用法 108

CHANGE TEXT CONFIGURATION  
コマンド

構文 137

使用法 22

COMMITCOUNT 構成パラメーター

説明 76

テキスト構成設定内の省略時値  
22

ログ用スペースの確保 77

CHANGE TEXT

CONFIGURATION 中の 139

ENABLE TEXT COLUMN 中の  
155

ENABLE TEXT TABLE 中の  
176

CONNECT コマンド

構文 140

使用法 67

CONTAINS 関数

構文 207

例 90

COUNT キーワード 226

## D

DATALINK データ・タイプ 78

DB2 エクステンダー

使用例 4

DB2 テキスト・エクステンダーの概  
説 3

DB2 テキスト・エクステンダーの関  
数により戻される SQL の状態  
309

DB2DBDFT、環境変数 19

DB2INSTANCE、環境変数 19

DB2TEXTFH 特殊タイプ 203

DB2TEXTFHLISTP 特殊タイプ 204

DB2TEXTH 特殊タイプ 203

DB2TEXTHLISTP 特殊タイプ 204

DB2TX、コマンド行プロセッサ

構文 134

使用法 65

DB2TX.SAMPLE テーブル

削除 122

作成用ユーティリティ 18

説明 84

DB2TX\_ 環境変数

説明 19

表示 112

DB2TX\_INSTOWNERHOMEDIR、環  
境変数 20

DB2TX\_INSTOWNER、環境変数 19

DBCS 文書、検索 48

DELETE INDEX EVENTS コマンド

構文 142

例 110

DesCloseDocument 関数

使用法 129

説明 237

DESCL.INI 289

DesEndBrowseSession 関数

使用法 129

説明 238

DesFreeBrowseInfo 関数

使用法 130

説明 239

DesGetBrowseInfo 関数

使用法 127

説明 240

DesGetMatches 関数

使用法 128

説明 244

DesGetSearchResultTable 関数  
 使用法 126  
 説明 250

DESMODEL.INI 53

DesOpenDocument 関数  
 使用法 128  
 説明 255

DESRESTB、結果表の作成 252

DESSAMP1、サンプル・プログラム  
 261

DESSRV.INI 291

DesStartBrowseSession 関数  
 使用法 127  
 説明 258

DES\_BROWSE、  
 DesGetSearchResultTable 中のオプション 252

DES\_EXTENDED、DesOpenDocument  
 中のオプション 255

DES\_EXT.H ヘッダー・ファイル  
 123

DES\_FAST、DesOpenDocument 中の  
 オプション 255

DES\_MATCH、DesGetSearchResultTable  
 中のオプション 251

DES\_NOBROWSE、  
 DesGetSearchResultTable 中のオプション 252

DES\_RANKANDMATCH、  
 DesGetSearchResultTable 中のオプション 251

DES\_RANK、DesGetSearchResultTable  
 中のオプション 251

DES\_TEXTHANDLEONLY、  
 DesGetSearchResultTable 中のオプション 251

DIRECTORY キーワード  
 現行設定の表示 116  
 テキスト構成設定内の省略時値  
 21

CHANGE TEXT  
 CONFIGURATION 中の 139

ENABLE TEXT COLUMN 中の  
 154

ENABLE TEXT TABLE 中の  
 163

DISABLE DATABASE コマンド  
 構文 143  
 使用法 122

DISABLE TEXT COLUMN コマンド  
 構文 144  
 使用法 120

DISABLE TEXT FILES コマンド  
 構文 145  
 使用法 121

DISABLE TEXT TABLE コマンド  
 構文 146  
 使用法 121

## E

ENABLE DATABASE コマンド  
 構文 147  
 使用法 68

ENABLE TEXT COLUMN コマンド  
 構文 148  
 使用法 72

ENABLE TEXT FILES コマンド  
 構文 157  
 使用法 79

ENABLE TEXT TABLE コマンド  
 構文 160  
 使用法 69

EXPAND キーワード 227

## F

FFT、文書形式 35

FILE 関数  
 構文 208  
 例 104

FORMAT 関数  
 構文 209  
 例 104

FUNCTION キーワード  
 説明 77

ENABLE TEXT COLUMN 中の  
 149

FUZZY FORM OF キーワード 228

## G

GET ENVIRONMENT コマンド  
 構文 165

GET ENVIRONMENT コマンド (続  
 き)  
 例および出力 112

GET INDEX SETTINGS コマンド  
 構文 166  
 例および出力 116

GET INDEX STATUS コマンド  
 構文 168  
 例および出力 113

GET STATUS コマンド  
 構文 169  
 例および出力 111

GET TEXT CONFIGURATION コマ  
 ンド  
 構文 170  
 例および出力 113

GET TEXT INFO コマンド  
 構文 171  
 例および出力 117

GUI サンプル 31

## H

HANDLE 関数  
 構文 210  
 使用法 105

HANDLE\_LIST 関数  
 構文 211  
 使用法 105

HTML 構造化文書 53

HTML 文書、セクション・サポート  
 56

HTML、文書形式 35

## I

IN SAME PARAGRAPH AS キーワ  
 ード 228

IN SAME SENTENCE AS キーワ  
 ード 228

INDEXOPTION キーワード  
 CHANGE TEXT  
 CONFIGURATION 中の 138

ENABLE TEXT COLUMN 中の  
 152

ENABLE TEXT TABLE 中の  
 161

INDEXPROPERTY キーワード  
ENABLE TEXT COLUMN 中の  
153  
ENABLE TEXT TABLE 中の  
162

INDEXTYPE キーワード  
CHANGE TEXT  
CONFIGURATION 中の 138  
ENABLE TEXT COLUMN 中の  
152  
ENABLE TEXT TABLE 中の  
161

INIT\_TEXT\_HANDLE 関数  
構文 212  
例 103

## L

LANGUAGE 関数  
構文 214  
例 104

LANGUAGE キーワード 139, 150  
LOGPRIMARY、LOGSECOND、および  
LOGFILSIZ パラメーター、DB2  
内の 76

## M

Microsoft、文書形式 35

## N

Ngram 索引  
検索オプションの省略時値 228  
説明 48  
CASE\_ENABLED オプション 49  
NODE キーワード  
ENABLE TEXT COLUMN 中の  
154  
ENABLE TEXT TABLE 中の  
163

NORMALIZED キーワード  
ENABLE TEXT COLUMN 中の  
153

NOT  
検索引き数内のキーワード 227  
ブール演算子 98

NO\_OF\_DOCUMENTS 関数  
構文 215  
NUMBER\_OF\_MATCHES 関数、構文  
216

## O

ON NODE キーワード  
ENABLE TEXT COLUMN 中の  
154  
ENABLE TEXT TABLE 中の  
163

OR ブール演算子 92

## P

PRECISE FORM OF キーワード  
228

## Q

QUIT コマンド  
構文 172  
使用法 81

## R

RANK 関数  
構文 217  
例 91

REFINE 関数  
構文 218  
例 101

REORGANIZE INDEX コマンド  
構文 173  
例 110

RESET INDEX STATUS コマンド  
構文 174  
例 109

RESULT LIMIT キーワード 226

## S

SEARCH\_RESULT 関数  
構文 219  
例 105

SET CURRENT FUNCTION PATH  
ステートメント 89  
STEMMED FORM OF キーワード  
228

## T

TERM OF キーワード 227  
TEXTINDEXES カタログ表示

削除 122  
作成 68  
内容 118

TXICRT コマンド  
構文 179  
作成、DB2 テキスト・エクステ  
ンダー・インスタンスの 23

TXIDROP コマンド  
構文 181

TXILIST コマンド  
構文 182

TXIPCLEAN コマンド  
構文 183

TXNADD コマンド  
構文 184

TXNCHECK コマンド  
構文 185

TXNDROP コマンド  
構文 186

TXSAMPLE コマンド  
構文 187  
使用法 18

TXSAMPLE.UDF  
実行 84

TXSTART コマンド  
構文 188  
使用法 24

TXSTATUS コマンド  
構文 189  
使用法 24

TXSTOP コマンド  
構文 190  
使用法 25

TXTHESC コマンド  
構文 191

TXTHESN コマンド  
構文 193



TXTRACE コマンド

構文 195

使用方法 26

TXVERIFY

サンプル・データベースの作成

18

TXVERIFY コマンド

構文 200

TXWIZARD コマンド

構文 201

## [特殊文字]

&(AND) 演算子、検索引き数内の

検索引き数の構文 227

使用方法 92

|(OR) 演算子、検索引き数内の

検索引き数の構文 227

使用方法 92

## U

UDT 203

UPDATE INDEX コマンド

構文 175

例 107

UPDATEFREQ キーワード

CHANGE INDEX SETTINGS 中の  
136

CHANGE TEXT

CONFIGURATION 中の 138

ENABLE TEXT COLUMN 中の  
153

ENABLE TEXT TABLE 中の  
162

UPDATEINDEX キーワード

現行設定の表示 116

テキスト構成設定内の省略時値  
22

CHANGE TEXT

CONFIGURATION 中の 139

ENABLE TEXT COLUMN 中の  
154

GET INDEX SETTINGS コマンド  
166

## W

WordPerfect、文書形式 35

## X

XML 構造化文書 53

XML 文書、セクション・サポート  
58

XML、文書形式 35



---

## IBM と連絡をとる

技術上の問題がある場合は、時間をとって**問題判別の手引き** に定義されている処置を検討し、それらの提案を実行した後で、DB2 顧客サービスに連絡をとってください。この資料には、DB2 顧客サービスがお客さまを支援するために必要とする情報が説明されています。

---

### 製品情報

以下の情報は英語で提供されます。内容は英語版製品に関する情報です。

#### **<http://www.ibm.com/software/data/>**

DB2 World Wide Web ページには、ニュース、製品説明、研修スケジュールなどの DB2 に関する最新情報が提供されています。ただし、提供されている情報は英語です。

#### **<http://www.ibm.com/software/data/db2/library/>**

「DB2 Product and Service Technical Library」では、よくされる質問 (FAQ)、修正内容、資料、および最新の DB2 技術情報などの情報へのアクセスが提供されています。

**注:** この情報のご提供は英語のみとなりますのでご注意ください。

#### **<http://www.elink.ibm.com/pbl/pbl/>**

「International Publications」注文用 Web サイトでは、マニュアルの注文方法についての情報を提供しています。ただし、提供されている情報は英語です。

#### **<http://www.ibm.com/education/certify/>**

IBM の「Professional Certification Program」Web サイトでは、DB2 を含むさまざまな IBM 製品の認証テストの情報を提供しています。ただし、提供されている情報は英語です。

#### **<ftp.software.ibm.com>**

匿名でログオンしてください。ディレクトリー /ps/products/db2 には、DB2 および多数の他製品に関連したデモ、修正プログラム、情報、およびツールがあります。ただし、提供されている情報は英語です。

**comp.databases.ibm-db2, bit.listserv.db2-l**

これらのインターネット・ニュースグループは、ユーザーが DB2 製品に関する自分の経験について話し合うために利用できます。ただし、提供されている情報は英語です。

**CompuServe: GO IBMDB2**

このコマンドを入力すると、IBM DB2 Family forum にアクセスできます。すべての DB2 製品が、このフォーラムでサポートされています。ただし、提供されている情報は英語です。

米国以外の国で IBM に連絡する方法については、*IBM Software Support Handbook* の Appendix A を参照してください。この資料にアクセスするには、Web ページ: <http://www.ibm.com/support/> にアクセスし、ページの最下部にある「IBM Software Support Handbook」リンク・ボタンを選択します。

**注:** 国によっては、IBM が承認している販売業者が、IBM サポート・センターの代わりにそれら販売業者のサポート・センターに連絡する場合があります。





Printed in Japan

SC88-8610-00



日本アイ・ビー・エム株式会社

〒106-8711 東京都港区六本木3-2-12

Spine information:



DB2<sup>®</sup> ユニバーサル・デー  
タベース

DB2 テキスト・エクステンダー 管理お  
よびプログラミング

バージョン 7