

DB2[®] 通用数据库



DB2 Text Extender 管理和程序设计

版本 7

DB2[®] 通用数据库



DB2 Text Extender 管理和程序设计

版本 7

注释!

在使用此信息以及它支持的产品之前，确保阅读第305页的『注意事项』下的一般信息。

第一版，2000年6月

此版本适用于 DB2 通用数据库 Text Extender，（它是版本 7 的 DB2 通用数据库 (DB2 UDB) 的一个功能部件，5645-DB2），以及所有后续的发行版和修订版，直至新版本中指示不再适用为止。

此版本替换并废弃 SC26-9646-00。

本文档包含 IBM 的专利信息。它在许可协议下提供，并受版权法保护。本出版物包含的信息不包括任何产品保证，且本手册提供的任何声明不应作如此解释。

通过您当地的 IBM 代表或 IBM 分部可订购出版物，或者，通过致电 1-800-879-2755（在美国）或 1-800-IBM-4YOU（在加拿大）来订购出版物。

当您发送信息给 IBM 后，即授予 IBM 非专有权，IBM 对于您所提供的任何信息，有权利以任何它认为适当的方式使用或散发，而不必对您负任何责任。

目录

关于本书	vii	使文本可搜索的样本示例	25
谁应使用本书	vii	创建索引和搜索的样本向导	27
使用本书的方法	vii	第4章 计划搜索需要	29
如何阅读语法图	viii	为何要对文本文档创建索引	29
如何发送意见	ix	受支持的文档格式	32
更改摘要	xi	HTML 文档, 特殊考虑事项	32
第1部分 指南.	1	XML 文档, 特殊考虑事项	33
第1章 DB2 Text Extender 概述	3	使用未受支持的文档格式	33
在 DB2 客户机 / 服务器环境中的 DB2 Text		语言	35
Extender	4	CCSID	35
在分区数据库环境中的 DB2 Text Extender	7	EBCDIC	35
第2章 安装、设置和维护	9	ASCII	36
在 AIX、Sun-Solaris 和 HP-UX 系统中安装	9	DBCS	37
安装需求	9	UNICODE	37
对于 AIX, 第 1 步: 安装产品部件	9	在存储和启用文本时避免代码页问题	38
对于 SunXSolaris, 第 1 步: 安装产品部件	10	搜索类型	40
对于 HP-UX, 第 1 步: 安装产品部件	11	语言搜索	41
第 2 步: 建立 DB2 Text Extender 实例	13	精确搜索	42
第 3 步: 启动 DB2 Text Extender 服务器	13	在 DBCS 文档进行模糊搜索或搜索	43
在 Windows 95、Windows 98、Windows NT		更改索引类型	44
和 Windows 2000 中安装	13	为表创建一个或若干个文本索引	44
服务器需求	13	计算索引的大小	44
客户机需求	14	更新索引	45
安装 DB2 Extender	14	维护多节点环境中的索引	46
创建和管理实例	15	使用结构文档 (部分支持)	47
准备样本数据库以进行安装验证	16	属性部分	47
配置	16	平面文件和 HTML 文档	50
环境变量	17	XML 文档	51
文本配置设置	18	词典、停止字列表、缩写列表和语言参数	52
更改文本配置	19	修改停止字和缩写文件	53
设置和维护 DB2 Text Extender 服务器	20	第5章 使文本可搜索	55
创建 DB2 Text Extender 实例	20	在使文本可搜索之前作准备	55
启动和停止 DB2 Text Extender 服务器	21	启动 DB2 Text Extender 命令行处理器	56
备份和复原索引及启用的数据库	21	命令行处理器帮助	58
跟踪故障	22	连接至数据库	58
第3章 入门	25	启用数据库	59
		启用文本表 (可选)	60
		示例	61
		启用文本列	63

添加句柄列	64	维护文本索引.	95
设置文档信息.	65	更新索引	95
创建日志表	65	更新外部文件的索引	96
创建索引	65	更改索引的设置	96
示例.	66	重设索引状态.	97
启用大型表中的文本列.	67	删除索引事件.	98
启用不支持的数据类型的文本列.	68	重组索引	98
启用包含数据类型 DATALINK 的列	68	获取有用信息	100
启用外部文本文件	69	显示已启用状态信息	100
示例.	70	显示环境变量的设置	101
结束对话	71	显示文本配置设置.	101
第6章 如何进行搜索	73	显示索引的状态	102
何处可找到搜索函数的语法示例.	73	显示错误事件	103
样本表 DB2TX.SAMPLE	74	显示索引设置	104
外部文件句柄.	78	显示列的文本设置.	105
设置当前函数路径	78	使用 DB2 Text Extender 目录视图	106
搜索文本	78	反向文本准备过程.	107
执行查询	79	禁用文本列	108
搜索并返回所找到的匹配数	80	禁用文本文件	108
搜索并返回所找到的文本文档的等级	80	禁用文本表	109
指定搜索变元.	81	禁用数据库	109
搜索若干词条.	81	第8章 使用 API 函数进行搜索和浏览	111
用布尔运算符 AND 和 OR 来进行搜索	81	安装应用程序	111
搜索词条变体.	82	链接 UNIX 应用程序.	111
搜索词条的各部分 (字符屏蔽)	82	链接 OS/2 或 Windows 应用程序	111
查找已包含屏蔽字符的词条	83	API 函数概述	112
搜索任何顺序的词条	83	搜索文本.	113
在同一个句子或段落中搜索词条.	84	获取搜索结果表 (DesGetSearchResultTable)	114
在结构化文档的节中搜索词条	84	浏览文本.	114
搜索词条的同义词	84	获取浏览信息 (DesGetBrowseInfo)	115
执行语言搜索.	85	启动浏览对话 (DesStartBrowseSession)	115
用布尔运算符 NOT 进行搜索	86	打开文档 (DesOpenDocument)	115
模糊搜索	87	获取匹配 (DesGetMatches)	116
考虑字与短语的边界	87	关闭文档 (DesCloseDocument)	116
搜索发音类似的字	87	结束浏览对话 (DesEndBrowseSession)	117
同义词典搜索.	87	释放浏览信息 (DesFreeBrowseInfo)	117
自由文本和混合搜索	88	<hr/>	
细化先前搜索.	89	第2部分 参考	119
在句柄中设置并抽取信息.	90	第9章 客户机的文本准备和管理命令	121
插入新文本时设置文本信息	91	CHANGE INDEX SETTINGS	123
从句柄中抽取信息	91	CHANGE TEXT CONFIGURATION	125
从句柄中更改信息	92	CONNECT	128
提高搜索性能.	92	DELETE INDEX EVENTS	129
第7章 管理	95	DISABLE DATABASE	130

DISABLE TEXT COLUMN.	131	LANGUAGE	194
DISABLE TEXT FILES	132	NO_OF_DOCUMENTS	195
DISABLE TEXT TABLE	133	NO_OF_MATCHES	196
ENABLE DATABASE	134	RANK.	197
ENABLE TEXT COLUMN	135	REFINE	198
ENABLE TEXT FILES	142	SEARCH_RESULT	199
ENABLE TEXT TABLE.	145		
GET ENVIRONMENT	149	第12章 搜索变元的语法	201
GET INDEX SETTINGS.	150	搜索变元.	202
GET INDEX STATUS	152		
GET STATUS	153	第13章 用于搜索和浏览的 API 函数	213
GET TEXT CONFIGURATION	154	DesCloseDocument.	214
GET TEXT INFO	155	DesEndBrowseSession.	215
QUIT	156	DesFreeBrowseInfo.	216
REORGANIZE INDEX	157	DesGetBrowseInfo.	217
RESET INDEX STATUS	158	DesGetMatches	220
UPDATE INDEX	159	DesGetSearchResultTable.	225
		DesOpenDocument.	229
		DesStartBrowseSession	231
第10章 服务器的管理命令	161		
TXICRT	162	第14章 样本 API 程序	233
TXIDROP	164		
TXILIST (仅限于 UNIX)	165	第15章 对语言索引和精确索引的语言处理	235
TXIPCLEAN (仅限于 UNIX)	166	创建索引时的语言处理	235
TXNADD	167	基本文本分析	236
TXNCHECK.	168	将词条缩减至其基本格式 (主题化)	240
TXNDROP	169	停止字过滤	241
TXSAMPLE.	170	字符分解 (划分复合词)	241
TXSTART	171	用于检索的语言处理	242
TXSTATUS	172	同义词	242
TXSTOP	173	同义词扩展	243
TXTHESC	174	声音扩展	243
TXTHESN	176	字符和字屏蔽	243
TXTRACE	178	用于浏览的语言处理	244
TXVERIFY	183	步骤 1: 规范化和词条扩展	244
TXWIZARD.	184	步骤 2: 扩展匹配	244
		同义词概念	246
第11章 搜索函数	185	词条	247
DB2 Text Extender 单值类型	185	关系	247
DB2 Text Extender 函数概述	186	Ngram 同义词典关系	249
CCSID	187	创建同义词典	250
CONTAINS	188	创建 Ngram 同义词典	254
FILE	189		
FORMAT.	190	第16章 配置文件	257
HANDLE.	191	客户机配置文件	257
HANDLE_LIST.	192	服务器配置文件	258
INIT_TEXT_HANDLE	193	文本分析配置文件	260

公共选项	261	注意事项	305
第17章 返回码	263	注册商标	307
第18章 信息	271	词汇表	309
由 DB2 Text Extender 函数返回的 SQL 状态	271	索引	315
来自 DB2 Text Extender 的信息	273	与 IBM 联系	325
第19章 搜索引擎原因码	285	产品信息	325
第20章 错误事件原因码	289	读者意见表	327
<hr/>			
第3部分 附录及附属资料	303		

关于本书

本书描述如何使用 DB2 Text Extender 来准备和维护 DB2(R) 数据库，以检索文本数据。它还描述了如何使用 DB2 Text Extender 提供的 SQL 函数和应用程序设计接口 (API) 来存取和处理这些类型的数据。通过将 DB2 Text Extender 的函数合并到程序的 SQL 语句中，并且合并 API，就可以创建功能强大的通用的用于检索文本的程序。

本书中提到的 “DB2” 指的就是 DB2 UDB。

谁应使用本书

本书面向熟悉 DB2 管理概念、工具和技术的 DB2 数据库管理员。

本书还面向熟悉 SQL，并熟悉可用于 DB2 应用程序的一种或多种程序设计语言的 DB2 应用程序员。

本书是供将使用 DB2 Text Extender 的人员使用的。使用 DB2 Image Extender、Audio Extender 和 Video Extender 的人员应阅读 *DB2 Image Extender*、*Audio Extender* 和 *Video Extender* 管理和程序设计。

使用本书的方法

本书的结构如下：

『第 1 部分 指南』

此部分对 DB2 Text Extender 进行了概述，描述如何安装和设置它，并讨论计划方面的考虑事项。它还描述如何准备和维护 DB2 数据库，以便使您可以搜索文本。

若您不熟悉 DB2 Text Extender，并且想了解如何使用 DB2 Text Extender 功能和 API 来搜索文本，请阅读此部分。

『第 2 部分 参考信息』

此部分提供了 DB2 Text Extender 功能、API、命令的参考信息，以及诊断信息（例如信息和代码）。

若您熟悉 DB2 Text Extender 概念和任务，但需要关于特定的 DB2 Text Extender 功能、API、命令、信息或代码的信息，则阅读此部分。

如何阅读语法图

整本书内，都使用如下定义的结构描述语法：

- 顺着线路从左到右、从上到下阅读语法图。

▶— 符号指示语句开始。

—▶ 符号指示语句语法在下一行上继续。

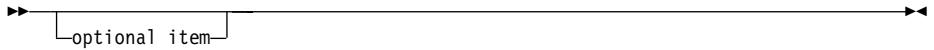
▶— 符号指示语句继续前一行。

—▶ 符号指示语句结束。

- 必需的项目出现在水平线（主路径）上。

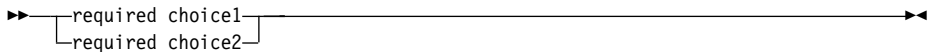


- 可选的项目出现在主路径下面。



- 若可从两个或多个项目中进行选择，则它们出现在堆栈中。

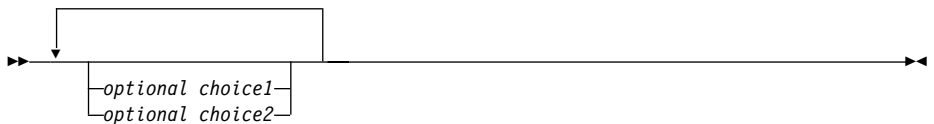
若必须选择其中一个项目，则堆栈的其中一个项目出现在主路径上。



若“不选择任何项目”是一个选项，则整个堆栈出现在主路径下面。



堆栈上方的双箭头指示可选择堆栈中的多个项目。



- 关键字以大写字体出现；必须如同显示的一样准确拼写它们。变量以小写出现（例如，srcpath）。它们代表语法中用户提供的名称或值。

- 若显示标点符号、圆括号、算术运算符或其他这样的符号，则必须作为语法的一部分输入它们。

如何发送意见

您的反馈能帮助 IBM 提供高质量的信息。请将您对本书或其他 DB2 Extender 文档的意见发送给我们。您可以用下列任何方法来提供意见:

- 从 Web 发送您的意见。访问以下地址的 Web 站点:

<http://www.ibm.com/software/data/db2/extenders>

此 Web 站点有一个反馈页, 您可使用该页输入并发送意见。

- 通过电子邮件将您的意见发送至 swsdid@de.ibm.com。务必提供书名、书的部件号、产品版本以及您有意见的文本的特定位置 (如页码和表号, 如果有的话)。
- 填写本书背后的“读者意见表”, 并通过邮寄、传真, 或通过交给 IBM 代表返回它。邮寄地址在该表的背后。传真号码是 +49-(0)7031-16-4892。

当您发送信息给 IBM 时, 即授予了 IBM 非专有权, 以她认为适当的任何方式使用或分发该信息, 而无须负任何责任。

更改摘要

在“发行版 7.1”中，对此文件所作的更改包括：

- **搜索具有特殊类型的部分**

支持的类型是日期 / 时间和整数。例如，这允许您在发行日期中搜索日期在 12/13/99 之后的文档。

- **Datalink 支持**

可以对通过具有 Datalink 数据类型的一类来引用的文档进行索引。

- **更多 CCSID 支持**

已经除去了 Ngram 索引类型的 CCSID 限制。

- **不再支持双重索引类型**

而是使用语言索引类型或精确的规格化索引类型。

- **更改了 search_result 语法**

已经更改了 search_result 函数的语法。仍然支持旧语法，以便实现向后兼容。

- **支持更多语言**

已经扩充了受支持的语言和 CCSID 的列表

- **已经扩充了配置能力**

可以配置 tokenizer 和定制服务器配置，例如，重新组织阈值。

- **用于 UNIX 的新系统清理命令**

若不能成功地启动 DB2 Text Extender 服务器，则对于 UNIX 平台，有新的服务器命令 TXIPCLEAN 来使系统清理更容易。

第1部分 指南

第1章 DB2 Text Extender 概述

DB2 Text Extender 是 DB2 Extender 系列中的一员。它允许程序员在他们的应用程序中对文本文档使用 SQL 查询。

其他 Extender 可以搜索图像数据、视频数据和语音数据，并且可以处理 XML 文档和空间数据。

通过利用在 DB2 UDB 中可得到的功能部件，DB2 Text Extender 将全文本检索功能添加给 SQL 查询，从而使您能够在数据库中存储非结构化文本文档。

DB2 Text Extender 为 DB2 UDB 用户及应用程序员提供了一种在此类文档中进行搜索的快速、通用和智能化的方法。DB2 Text Extender 的强大功能在于它能够高速搜索成千上万的大型文本文档，不仅可以查找您直接查询的内容，而且还能够查找单词变体及同义词。

您并不仅限于在存储在 DB2 UDB 数据库中的文本文档中进行搜索，也可以在存储在文件中的文本文档中进行搜索。

DB2 Text Extender 的核心部分就是在第235页的『第15章 对语言索引和精确索引的语言处理』中描述的 IBM 的高性能语言搜索技术。该技术允许应用程序以多种方法存取和检索文本文档。应用程序可以：

- 对包含特定文本、单词或短语的同义词的文档进行搜索，或者在同一句子或段落中寻找近似的单词。
- 使用前端屏蔽、中间屏蔽和末端屏蔽来对单词和字符进行屏蔽，以执行通配符搜索。
- 搜索各种文档格式的用各种语言编写的文档。
- 对与搜索词条具有相似拼写的单词执行“模糊”搜索。这对于查找即使是拼写错误的单词也是很有用的。
- 执行自由文本搜索，在该搜索中，搜索变元是用自然语言表示的。
- 搜索与搜索词条发音相似的单词。

可以将文本搜索与商务数据查询结合起来进行。例如，可以在应用程序中编写 SQL 查询，以对由特定作者创建的、在一定日期范围内并且包含特定单词或短语的文本文档进行搜索。通过使用 DB2 Text Extender 程序设计界面，还可以允许应用程序用户浏览文档。

概述

通过将全文本搜索集成到 DB2 UDB 的 SELECT 查询中，就具有了强大的检索功能。下列 SQL 语句显示一个示例：

```
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 是多种 DB2 Text Extender 搜索功能中的一个。DB2BOOKS_HANDLE 是一个句柄列的名称，该句柄列引用包含要搜索的文本文档的列 DB2BOOKS。该语句的其余部分是一个搜索变元的例子，用于寻找与 table 处在同一段落中的 authorization，并寻找 delete 或其同义词。

在 DB2 客户机 / 服务器环境中的 DB2 Text Extender

第5页的图1显示如何将 DB2 Text Extender 集成到 DB2 客户机 / 服务器环境中。

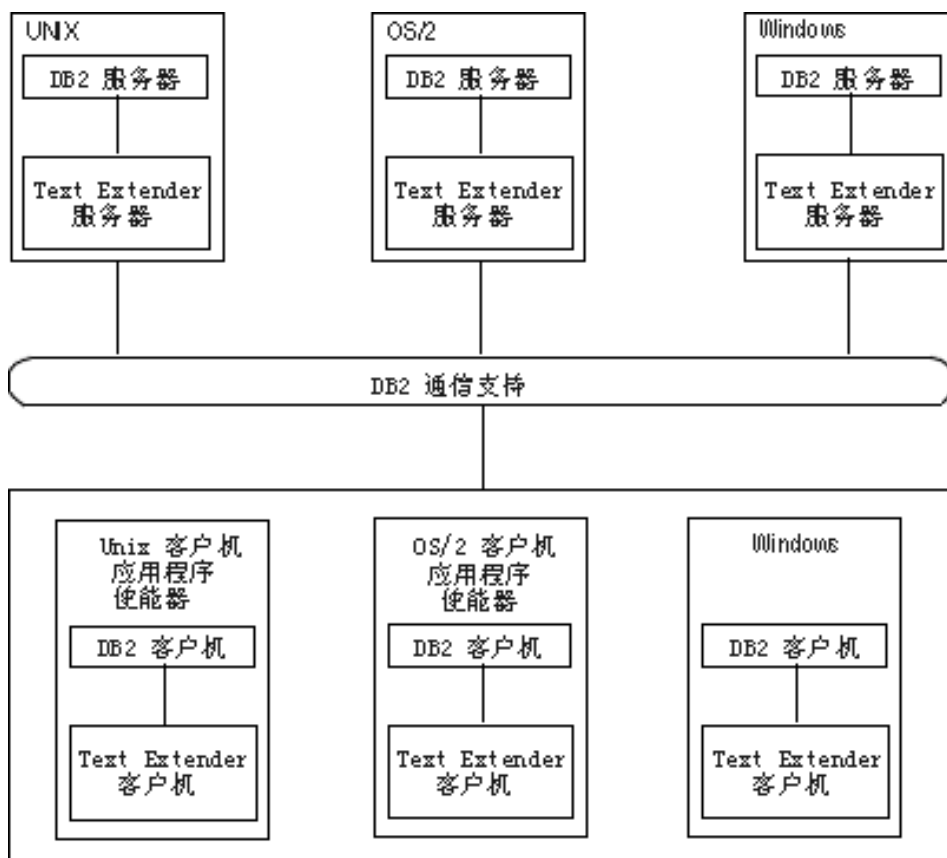


图1. 将 DB2 Text Extender 集成到 DB2 客户机/服务器环境中

有关客户机的“DB2 通信支持”协议（如 TCP/IP 或 NETBIOS）的列表，参见相应平台的 *DB2 快速入门*。

DB2 Text Extender 的主要部件与 DB2 服务器安装在同一机器上。一个 DB2 服务器实例只能与一个 DB2 Text Extender 服务器实例安装在一起。

DB2 Text Extender 的安装十分灵活，可包括：

- 位于图1所示的某种操作系统上的一个或多个 DB2 Text Extender 服务器，其中，UNIX 包括 AIX(R)、SUN-Solaris 和 HP-UX 工作站。
- 可存取一个或多个远程 DB2 Text Extender 服务器的 AIX、SUN-Solaris、Windows NT 和 Windows 2000 客户机。
- 包含本地服务器而且可存取远程服务器的 AIX 客户机。

概述

图2 显示了一个典型的 DB2 Text Extender 配置。要从客户机中运行 DB2 Text Extender, 必须首先安装 DB2 客户机和一些 DB2 Text Extender 实用程序。这些实用程序构成了 DB2 Text Extender “client”, 尽管严格说来它并不是真正的客户机。该客户机通过 DB2 客户机连接与服务器进行通信。

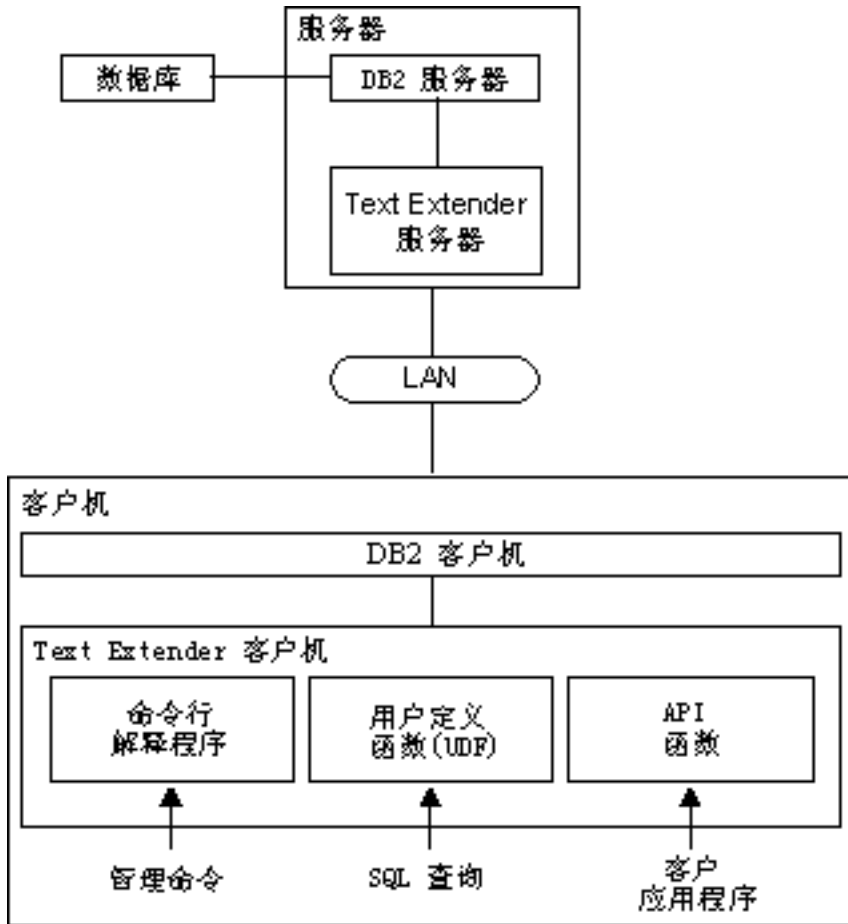


图2. DB2 Text Extender 配置

DB2 Text Extender 具有下列主要部件:

- **命令行解释程序。** 可使用一些命令来准备列中的文本以供要搜索, 并维护文本索引。
- **SQL 函数。** 可以将函数包括在 SQL 查询中, 用于在文本中进行搜索, 并了解一些内容, 如搜索词条在文本中出现的次数。确切地说, 该数字显示客户机上的 SQL 函数, 原因是它们被用作 SQL 查询的一部分。事实上, 它们是服务器安

装的一部分，并在安装过程中执行。然而，在任何 DB2 客户机中都可以使用这些函数，而不需要安装 DB2 Text Extender 客户机。

- **应用程序设计接口 (API)**。它由一些可以用 C 语言程序调用的函数组成，用于在文本中进行搜索并显示搜索结果。

提示

DB2 Text Extender 客户机实用程序提供了文本准备功能、管理功能和 API。（在服务器上，这些功能可用。）要使用这些功能，必须安装 DB2 Text Extender 客户机。若仅在 DB2 Text Extender 服务器中执行文本准备和管理，则只需要在使用 API 函数的客户机上安装 DB2 Text Extender 客户机实用程序。

若在使用 DB2 UDB SQL 语句的客户机中只需要搜索功能，则不需要安装 DB2 Text Extender 客户机。所有通信都是由 DB2 UDB 处理的，而 DB2 Text Extender 搜索引擎仅在服务器上运行。

在分区数据库环境中的 DB2 Text Extender

图3显示如何将 DB2 Text Extender 集成到 DB2 分区数据库环境中。

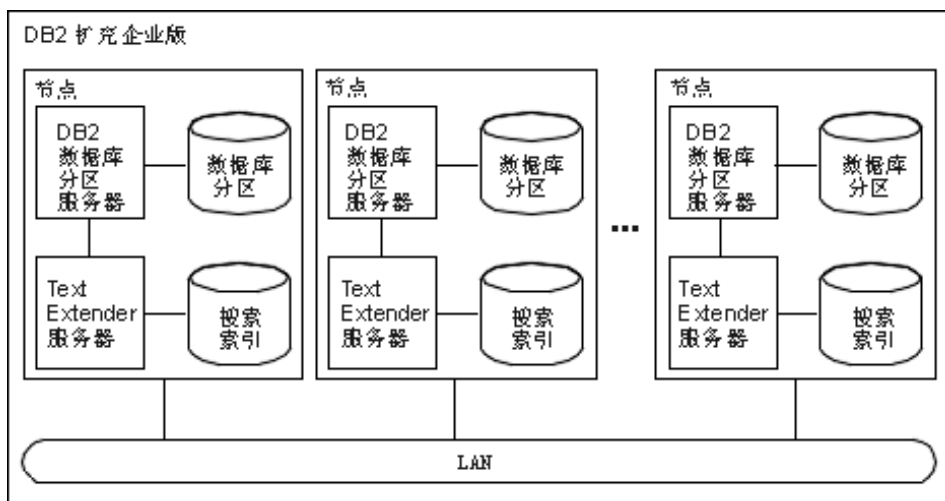


图3. 将 DB2 Text Extender 集成到 DB2 分区数据库环境

概述

每个数据库分区服务器（称为节点）都可被指定给单独的机器，它们也可以位于单个机器中。位于单独机器中的数据库分区服务器称为物理节点。那些位于单个机器中的数据库分区服务器称为逻辑节点。DB2 Text Extender 支持这两种配置。

节点组是一个或多个数据库分区服务器的已命名的子集。若将启用 DB2 Text Extender 的表指定给多分区节点组，则由 DB2 Text Extender 创建的文本索引与每个对应的表分区位于同一节点上。这可确保在索引期间不必在节点间移动数据。

DB2 Text Extender 服务器在节点组的每个分区上运行，并管理相应分区中的已索引行。可通过一个数据库分区来与 DB2 Text Extender 交互作用。每个数据库分区都可被用来发出 DB2 Text Extender 请求。将这些请求按路径发送至所有相关的数据库分区。

第2章 安装、设置和维护

在 AIX、Sun-Solaris 和 HP-UX 系统中安装

DB2 Text Extender 版本 7 AIX 版、Sun-Solaris 版和 HP-UX 版可以在非分区数据库环境中运行。对于 AIX 和 Sun-Solaris，它还可以利用“DB2 扩充企业版”提供的分区数据库支持。

以下是安装步骤：

1. 在目标机器中安装产品部件。
2. 建立 DB2 Text Extender 实例。
3. 启动 DB2 Text Extender 服务器。

安装需求

在安装 DB2 Text Extender 之前，必须确保您满足它们的服务器和客户机部件的下列安装需求。

服务器需求

服务器部件要求安装 DB2 版本 7.1（或更高版本）。

客户机需求

客户机部件要求安装“DB2 版本 7.1（或更高版本）个人版”、“DB2 版本 7.1（或更高版本）客户机应用程序使能器”、或“DB2 版本 7.1（或更高版本）软件开发工具箱”。

对于 AIX, 第 1 步: 安装产品部件

在安装期间，将自动创建一个称为 smadmin 的组。

1. 在目标客户机或服务器上注册为超级用户。
2. 在提示的界面中，使用“系统管理接口工具”（SMIT）来运行 `installp`，或者输入 `installp` 命令。
3. 输入 `smit install_latest` 命令。“软件安装”和“维护工具”菜单打开。
4. 在软件的输入设备 / 目录字段中输入程序包的位置。
5. 单击执行按钮或者按 ENTER 键。这将确认安装目录。
6. 在要安装的软件字段中标识是要安装客户机还是服务器部件。

在 UNIX 系统中安装

- 单击执行按钮或者按 ENTER 键。将提示您确认安装参数。要确认的话，按 ENTER 键。
- 在要安装的软件字段中标识您想在客户机中安装的词典（如果有的话）。若未指定词典，则安装“美国英语”词典。

表1列示了 DB2 Text Extender 词典。要在客户机中安装所有词典，在要安装的软件字段中输入 db2tx_07_01.dic。

表 1. DB2 Text Extender 词典

词典	语言
db2tx_07_01.dic.Ar_AA	阿拉伯语
db2tx_07_01.dic.Da_DK	丹麦语
db2tx_07_01.dic.Nl_NL	荷兰语
db2tx_07_01.dic.En_GB	英语（英国英语）
db2tx_07_01.dic.En_US	英语（美国英语）
db2tx_07_01.dic.Fi_FI	芬兰语
db2tx_07_01.dic.Fr_FR	法语
db2tx_07_01.dic.Fr_CA	法语（加拿大）
db2tx_07_01.dic.De_DE	德语
db2tx_07_01.dic.De_CH	德语（瑞士）
db2tx_07_01.dic.Iw_IL	希伯来语
db2tx_07_01.dic.Is_IS	冰岛语
db2tx_07_01.dic.It_IT	意大利语
db2tx_07_01.dic.No_NO	挪威语
db2tx_07_01.dic.Pt_PT	葡萄牙语
db2tx_07_01.dic.Ru_RU	俄语
db2tx_07_01.dic.Es_ES	西班牙语和加泰隆语
db2tx_07_01.dic.Sv_SE	瑞典语
db2tx_07_01.dic.Th_TH	泰国语

- 注销。

对于 SunXSolaris, 第 1 步: 安装产品部件

在安装期间，将自动创建一个称为 smadmin 的组。

- 在目标客户机或服务服务器上注册为超级用户。
- 使用 tar 来从修订包文件中获取可安装的程序包:

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


3. 使用 `pkgadd` 来将文件从目录中传送到目标机器中。可以选择下列两种方法中的其中一种来传送文件:
 - 为 DB2 Text Extender 客户机和服务器安装“美国英语”词典:


```
pkgadd -d . db2txclnt db2txsrv db2txcom db2txmsgs db2txmsc db2txdius
```
 - 指示您是否想安装 DB2 Text Extender 客户机或服务器, 以及您想安装哪些词典:


```
./db2tx_install
```
4. 注销。

对于 HP-UX, 第 1 步: 安装产品部件

在安装期间, 将自动创建一个称为 `smadmin` 的组。

1. 在目标客户机或服务器上注册为超级用户。
2. 使用 `tar` 来从修订包文件中获取可安装的程序包:

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

3. 运行 `swinstall` 程序:

```
swinstall
```

这将打开“软件选择”窗口和“指定源”窗口。

4. 如果需要的话, 更改“源主机名”。
5. 使用 `IBMdb2tx` 输入 `tar` 目录, 作为“源仓库路径”字段的值。
6. 在“更改软件视图”字段中, 选择产品。
7. 选择确认以返回到“软件选择”窗口, 该窗口中包含了要安装的可用软件的列表。
8. 现在, 执行下列步骤之一:
 - 要对 DB2 Text Extender 安装所有词典以及用提供的所有语言所编写的文档, 突出显示 DB2TX 产品。
 - 要安装 DB2 Text Extender 的部件:
 - a. 从操作菜单中选择打开项目。
 - b. 对于想要安装的每个文件集, 突出显示它, 并从操作菜单中选择选择安装的标记。可以选择安装 DB2 Text Extender 服务器或客户机:


```
Text Extender Server - DB2TX.db2txsrv
Text Extender Client - DB2TX.db2txclnt
```
 - c. 选择您想安装的词典(如果有的话)。第12页的表2列示了 DB2 Text Extender 词典。

表 2. DB2 Text Extender 词典

词典	语言
db2txdiaa	阿拉伯语
db2txdidk	丹麦语
db2txdinl	荷兰语
db2txdigb	英语（英国英语）
db2txdius	英语（美国英语）
db2txdifi	芬兰语
db2txdifr	法语
db2txdica	法语（加拿大）
db2txdide	德语
db2txdich	德语（瑞士）
db2txdiil	希伯来语
db2txdiis	冰岛语
db2txdiit	意大利语
db2txdino	挪威语
db2txdipt	葡萄牙语
db2txdies	俄语
db2txdiru	西班牙语和加泰隆语
db2txdise	瑞典语
db2txdith	泰国语

9. 从操作菜单中选择安装，以便开始产品安装，并打开“安装分析”窗口。
10. 当状态字段中显示就绪信息时，在“安装分析”窗口中选择确认按钮。
11. 在“确认”窗口中选择是按钮，以便确认您想安装“DB2 HP-UX 版”软件产品。
12. 在安装该软件时查看“安装”窗口，以读取处理数据。当状态字段指示就绪，并且“注意事项”窗口打开时，安装就完成了。

swinstall 程序装入文件集，并运行文件集的控制脚本。

HP-UX 操作系统为 swinstall 提供了详细帮助。通过输入以下命令，就可调出帮助：

```
man swinstall
```

13. 注销。

第 2 步: 建立 DB2 Text Extender 实例

1. 确保您是超级用户。
2. 对于 AIX, 输入:

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

对于 Solaris 和 HP-UX, 输入:

```
cd /opt/IBMd2tx/V7.1/instance
```

3. 若您具有现存的实例, 则输入:

```
./txiupdt instanceid db2instanceid
```

以使用新文件来更新您的实例。

否则, 要创建新的 DB2 Text Extender 实例, 输入:

```
./txicrt instanceid db2instanceid
```

其中, *instanceid* 是现存的 DB2 实例用户 ID 或者是要与 DB2 实例 ID 相关联的用户 ID。

运行 `txicrt` 来创建 `/u/instanceid/db2tx` 目录。不要在 `/u/instanceid/db2tx` 下面创建其他文件或目录; 若删除实例的话, 这些文件将丢失。

若您正在使用分区数据库环境, 则需要附加参数。有关详情, 参见第162页的『TXICRT』。

4. 注销。

第 3 步: 启动 DB2 Text Extender 服务器

1. 注册为 *instanceid*。
2. 停止并重新启动 DB2。
3. 要启动 DB2 Text Extender 服务器, 输入:

```
txstart
```

4. 注销。

在 Windows 95、Windows 98、Windows NT 和 Windows 2000 中安装

DB2 Text Extender 是作为 DB2 Extender 程序包的一部分来安装的。

服务器需求

可以在 Windows NT 或 Windows 2000 服务器中安装 DB2 Extender 的服务器部件。

在 Windows 系统中安装

要在非分区数据库环境中运行的服务器部件需要安装 DB2 版本 7.1（或更高版本）。

要在分区数据库环境中运行的服务器部件需要安装“DB2 扩充企业版”的版本 7.1（或更高版本）。

客户机需求

可以在 Windows 95、Windows 98、Windows NT 或 Windows 2000 中安装 DB2 Extender 的客户机部件。

客户机部件要求安装“DB2 版本 7.1（或更高版本）个人版”、“DB2 版本 7.1（或更高版本）客户机应用程序使能器”、或“DB2 版本 7.1（或更高版本）软件开发者工具箱 Windows 版”。

安装 DB2 Extender

要在 Windows 中安装 DB2 Extender:

1. 将 CD-ROM 插入客户机工作站的 CD-ROM 驱动器中。
2. 若启用了 CD 自动启动功能，则 DB2 Extender LaunchPad 将自动启动。否则，单击任务栏上的开始按钮。选择运行。 输入:

```
d:\db2ext\winnt95\setup
```

其中，“d”是 CD-ROM 驱动器盘符。

3. 从 LaunchPad，可以查看 DB2 Extender 自述文件。了解安装前提条件，或安装产品。
4. 选择左边的安装按钮以开始安装。
5. 在第一个窗口中，选择要安装的 DB2 Extender 的版本：客户机 / 服务器的 EEE 版、客户机 / 服务器的非 EEE 版、或者仅安装客户机。
6. 在下一个窗口中，选择将 Extender 安装在哪个文件夹中。
7. 然后选择您想要的安装类型：典型安装、定制安装或压缩安装。

典型安装包括 DB2 Extender 客户机 / 服务器、样本文件、SDK、“美国英语”联机文档和“美国英语”DB2 Text Extender 词典。

压缩安装将仅安装客户机 / 服务器和“美国英语”DB2 Text Extender 词典。

定制安装允许您选择下列部件:

- 若您正在分区数据库环境（DB2 扩充企业版）中安装 DB2 Extender，则选择 **DB2 Extender 服务器 (EEE)**。这也会安装 **DB2 Extender 客户机** 部件。
- 若您正在非分区数据库环境中安装 DB2 Extender，则选择 **DB2 Extender 服务器（非 EEE）**。这也会安装 **DB2 Extender 客户机** 部件。

- 若您正在安装 DB2 Extender 客户机环境，则选择 **DB2 Extender 客户机**。
- 若您想安装用于开发 DB2 Extender 应用程序的头文件和库，则选择 **DB2 Extender SDK**。
- 若您想安装样本程序和样本媒体文件，则选择**样本文件**。
- 若您想安装软拷贝格式的 Extender 出版物，则选择**联机文档**。一旦作出了此选择，您就可以在“说明”部分单击 **更改按钮**来查看可用的特定于语言的出版物。从列表中选择您想安装的出版物。
- 若您想安装对所选语言的 DB2 Text Extender 的语言支持，则选择 **Text Extender 词典**。一旦作出了该选择，您就可以在“说明”部分单击 **更改按钮**来查看可用的特定于语言的词典。从列表中选择您想安装的词典。始终都需要“美国英语”词典。

若没有足够的磁盘空间可供所选择的部件使用，则按上一步以重新选择安装目录。否则，按下一步。

8. 若您正在分区数据库环境中、并且是在拥有实例的机器上安装 Extender 服务器 (EEE)，则会提示您指定 TCP/IP 端口的范围。在与当前的 DB2 实例相关的所有节点上必须存在相同的共享名称。

若您正在一台未拥有实例的机器上安装 DB2 Extender，则会提示您输入拥有实例的机器的名称。将从拥有实例的机器中获得 TCP/IP 端口范围和数据路径信息。

9. 单击下一步按钮以便安装您所选择的部件。

当开始安装时，安装程序将检查您的工作站是否满足 DB2 Extender 的软件需求。

10. 关机，然后重新启动工作站。

创建和管理实例

可以创建和使用 DB2 Extender 服务器的多个实例。若已经为 DB2 服务器 Windows 版创建了多个实例，则应创建多个实例。DB2 Extender 服务器的每个实例都是与 DB2 服务器 Windows 版的实例相关联的。

在非分区数据库环境中，当您在 Windows 中安装 DB2 Extender 时，将创建一个名称为 DB2 的缺省 DB2 Extender 实例。

在分区数据库环境中，当您在 Windows 中安装 DB2 Extendr 时，将创建一个名称为 DB2MPP 的缺省 DB2 Extender 实例。在分区数据库环境中，不会为 DB2 Text Extender 创建缺省实例。要创建实例，使用 txicrt 命令。有关更多信息，参见第20页的『创建 DB2 Text Extender 实例』。

在 Windows 系统中安装

准备样本数据库以进行安装验证

DB2 Text Extender 提供实用程序以准备样本数据库。对于快速准备文本以在安装之后立即测试 DB2 Text Extender 的搜索能力，以及对于在这之后任何时候的一般测试目的来说，这些实用程序都是很有用的。

这些实用程序位于下列目录中：

- 对于 OS/2(R)、Windows NT 和 Windows 2000，它们位于：
`DMBMMPATH\samples`
其中 `DMBMMPATH` 是安装路径。
- 对于 AIX、HP-UX 和 SUN-Solaris，它们位于：`$HOME/db2tx/samples`

要在服务器或在客户机工作站上准备样本数据库：

1. 在操作系统提示处，运行：

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

此命令创建并启用样本数据库。

2. 在操作系统提示处，运行：

```
TXSAMPLE 数据库-name [user-id] [password]
```

仅当在客户机工作站上工作时才需要用户 ID 和口令。此命令执行下列操作：

连接至指定的数据库

创建表 `db2tx.sample`

调入样本英语文档来填充该表

启用表中具有下列索引类型的文本列：

精确

精确规范化

语言

语言且启用节

Ngram

Ngram 且区分大小写

等待构建文本索引。

配置

本节描述 DB2 Text Extender 环境变量和配置信息。它们均让您为 DB2 Text Extender 所需要的许多参数指定缺省值。

环境变量

环境变量设置环境参数的缺省值。要显示环境变量的当前设置，使用第101页的『显示环境变量的设置』中描述的 `GET ENVIRONMENT` 命令。

DB2INSTANCE	DB2 UDB 实例名。
DB2DBDFT	数据库的缺省名称。在未指定数据库名时假设的 DB2 UDB 数据库名。
DB2TX_INSTOWNER	DB2 Text Extender 实例名。这是拥有实例的用户的注册名。若 DB2 Text Extender 客户机与 UNIX 服务器相连，则它是必需的。
DB2TX_INSTOWNERHOMEDIR	实例拥有者的主目录（仅限于 UNIX）。

在 **Windows** 环境中，DB2 Text Extender 环境变量被指定为系统特性页中的系统变量。安装期间，它们被添加到您的系统设置中。

在 **OS/2** 环境中，安装期间，DB2 Text Extender 环境变量被添加至 `CONFIG.SYS` 文件中。

在 **UNIX** 环境中，DB2 Text Extender 环境变量（DB2DBDFT 除外）是在下列简要表 shell 脚本中指定的：

```
db2txcshrc (对于 C shell)
db2txprofile (对于 Korn shell 和 Bourne shell)。
```

创建实例时，文件 `db2txprofile` 或 `db2txcshrc` 被复制至 `.profile`。

创建实例时，`db2txprofile` 被添加至 `.profile` 和实例拥有者的 `db2profile`。shell 脚本存储在 `$DB2TX_INSTOWNERHOMEDIR/db2tx` 中。

还需要下列环境变量：

LANG	指定语言设置
NLSPATH	指定 DB2 Text Extender 信息文件的路径
DB2ENVLIST	由 DB2 传送至 DB2 Text Extender 函数的环境变量列表
RAHUSER	指定用于 DB2 命令 <code>db2all</code> 的实例拥有者的名称。仅在 DB2 EEE 环境中使用。

配置

若您实例所有者，则创建实例时将自动设置这些环境变量。若您不是实例所有者，则将实例所有者的 `db2tx` 简要表包括在内，而不是尝试人工设置这些环境变量。

提示

无论何时更改任何环境变量，要使这些更改生效，必须停止并重新启动 DB2 和 DB2 Text Extender。

文本配置设置

每个数据库都具有文本配置设置，它们由下列项组成：

- 文本特性
- 索引特性
- 处理特性

这些项是在启用数据库以供 DB2 Text Extender 使用时设置的。ENABLE DATABASE 命令采用在命令中指定的设置，或采用此处所述的初始设置。可以显示和更改这些缺省设置；参见第101页的『显示文本配置设置』和第19页的『更改文本配置』。

文本特性

第29页的『第4章 计划搜索需要』描述受 DB2 Text Extender 支持的文档格式、语言和 CCSID。各种命令都需要上述各项的缺省值。

FORMAT 初始设置: TDS

LANGUAGE 初始设置: 为数据库设置的 LANGUAGE

CCSID 初始设置: 为数据库设置的 CCSID

索引特性

DIRECTORY 用来存储索引的目录。

UNIX 系统的初始设置：
`DB2TX_INSTOWNERHOMEDIR/db2tx/indexes`

OS/2、Windows NT 和 Windows 2000 的初始设置：
`DMBMMPATH\instance\instance-name \db2tx\indexes`（其中

DMBMPATH 是安装有 DB2 Text Extender 的路径)。对于 Windows 客户机，必须将 *DIRECTORY* 设置为服务器上的一个目录名。

INDEXTYPE 要使用的索引类型。参见第40页的『搜索类型』以了解说明。

初始设置: NGRAM

UPDATEFREQ

定期索引更新的频率。参见第45页的『更新索引』以了解说明。

初始设置: NONE

处理特性

UPDATEINDEX

一个设置，用于确定何时进行首次索引更新；是在启用步骤期间立即更新，还是根据更新频率设置稍后更新 (*NOUPDATE*)，或是生成一个显式的 *UPDATE INDEX* 命令。

初始设置: UPDATE

COMMITCOUNT

一个设置，确定在多少句插入或更新语句后 DB2 Text Extender 发出 DB2 UDB 落实语句。参见第67页的『启用大型表中的文本列』。

初始设置: 0

更改文本配置

首次安装 DB2 Text Extender 时，缺省值是为文本配置设置的。要显示当前的文本配置值，参见第101页的『显示文本配置设置』。

当创建索引时，要更改用作缺省值的文本配置，输入：

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

示例:

要为将来的索引更改缺省索引类型和缺省索引目录:

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

要更改缺省的索引更新频率，使之在星期一至星期五的 12:00 或 15:00 进行更新（若至少有 100 个文本文档排队的话）：

配置

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

要停止索引的定期更新:

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

设置和维护 DB2 Text Extender 服务器

新安装 DB2 Text Extender 的下一步是设置 DB2 Text Extender 服务器以供 DB2 Text Extender 客户机使用。这通常由 DB2 Text Extender 服务器管理员来完成,且涉及以下几个方面:

1. 创建 DB2 Text Extender 实例
2. 启动 DB2 Text Extender 服务器

维护 DB2 Text Extender 服务器的其他任务有:

1. 备份和复原 DB2 Text Extender 索引及启用的数据库
2. 跟踪故障

参见第161页的『第10章 服务器的管理命令』以了解命令语法。

创建 DB2 Text Extender 实例

在开始使用 DB2 Text Extender 之前,必须创建 DB2 Text Extender 实例。所创建的每个实例都提供一个隔离的环境,在该环境中可以维护索引,并将它们存储在单独的目录中。

要在 **UNIX** 单节点环境中创建实例,输入:

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

其中 *instance-name* 是实例拥有者的用户 ID,而 *db2-instance-name* 是要与之相关联的 DB2 UDB 实例的名称。

在 **Windows NT** 和 **Windows 2000** 中,缺省实例是在安装期间创建的。要创建其他实例,输入:

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

要在多节点环境中创建实例,参见第162页的『TXICRT』。

要在 **UNIX** 系统中卸下实例，输入：

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

要在 **Windows NT** 和 **Windows 2000** 中卸下实例，输入：

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

要确定想要使用哪一个实例，设置适当的环境变量：

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

然后停止并重新启动 DB2 和 DB2 Text Extender。

启动和停止 DB2 Text Extender 服务器

要启动 **DB2 Text Extender** 服务器，注册至实例所有者用户 ID（仅 UNIX 系统），然后输入：

```
txstart
```

要显示 **DB2 Text Extender** 服务器的状态，输入：

```
txstatus
```

要停止 **DB2 Text Extender** 服务器，输入：

```
txstop
```

UNIX 用户的提示

如果在启动 DB2 Text Extender 服务器时有任何问题，尝试清除共享资源。参见第166页的『TXIPCLEAN（仅限于 UNIX）』。

备份和复原索引及启用的数据库

可备份和复原启用的数据库以及 DB2 Text Extender 已创建的文本索引。

要备份：

1. 找出 DB2 Text Extender 已启用的表。为此，输入：

```
db2tx GET STATUS
```

2. 找出数据库所使用的索引目录的名称。为此，输入：

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

3. 停止 DB2 Text Extender 服务器。为此，输入：

```
TXSTOP
```

设置和维护服务器

4. 备份索引目录及其子目录 `index` 和 `work`。
5. 备份文件 `desmastr.dat`，该文件位于：
 `instance_owner_home_directory/db2tx/txins000` (UNIX)
 或
 `DMBMPATH\db2tx\txins000` (OS/2、Windows NT 和 Windows 2000)

注意，对于分区数据库，有几个名为 `txins000`、`txins001`、... 的实例目录，因此，备份整个 `.../db2tx` 目录比备份单个目录中的文件要容易得多。

6. 重新启动 DB2 Text Extender 服务器：
 TXSTART

要复原：

1. 停止 DB2 Text Extender 服务器：
 TXSTOP
2. 保存现存的 `desmastr.dat` 文件。
3. 复原 `desmastr.dat` 文件的备份副本。
4. 将索引目录的备份副本复原至与以前相同的路径中。
5. 重新启动 DB2 Text Extender 服务器：
 TXSTART

跟踪故障

若需要向 IBM 代表报告错误，则可能会让您打开跟踪，以便可以将信息写入文件中，用来找出错误。仅在“IBM 支持中心”代表或技术支持代表的指导下，才可使用跟踪设施。

打开跟踪时会影响系统性能，因此，仅当发生错误情况时才使用跟踪。

要打开跟踪，输入：

```
txtrace on [options]
```

语法、事件和部件的列表都在第178页的『TXTRACE』中给出。其中还描述了其他选项。

可通过指定“mask”过滤跟踪，“mask”可导致跟踪根据每个跟踪记录的 ID 来接受或拒绝每个跟踪记录。缺省值是跟踪每一件事。

掩码由用句点分隔的四个部分组成，例如：2.2-6.1,3.*，其中：

- 2 指示 DB2 UDB DB2 Text Extender。

- 2-6 只包括事件 ID 为 2 至 6 的项目。
- 1,3 只包括那些由部件 1 和 3 报告的事件。
- * 包括部件的所有功能。

可排除严重性级别在某一严重性之下的系统错误，并且若跟踪缓冲区已满，则可指定是否保持第一个记录或最后一个记录。

要再生成错误并将二进制的跟踪信息写入转储文件，输入：

```
txtrace dump dump-filename
```

要生成转储文件的格式化版本，输入：

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

还可在跟踪打开的情况下将跟踪信息直接从共享内存写入格式化文件：

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

在将跟踪信息写入文件中之后，关闭跟踪：

```
txtrace off
```

第3章 入门

使用本章来熟悉使文本成为可搜索的基本知识。假定您使用的是运行中的 DB2 Text Extender 系统，已经安装并配置了该系统，且 DB2 Text Extender 实例就是在其中创建并启动的。

提示

本章仅描述使文本可搜索的基本知识。在准备您自己的文本以供搜索之前，请阅读第55页的『在使文本可搜索之前作准备』。

有两个方法可让您熟悉使文本可搜索的内容。

- 您可浏览一个简单的样本示例，人工输入命令，或
- 可运行样本索引和搜索向导。

使文本可搜索的样本示例

1. 在 **Windows** 系统中，启动 **DB2** 命令行处理器

在操作系统提示符处，输入：

```
=>db2 cmd
```

2. 启动 **DB2 Text Extender** 命令行处理器

在 UNIX 系统中，在操作系统提示符处，输入：

```
=>db2tx
```

在 Windows 系统中，在 DB2 提示符处，输入：

```
db2=>db2tx
```

3. 连接至数据库

选择包含您想要其成为可搜索的文本的数据库。要连接至该数据库，输入：

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

4. 启用要进行文本搜索的数据库

要启用连接的数据库，输入：

```
db2tx=>ENABLE DATABASE
```

5. 启用文本表以进行文本搜索（可选）

只有在您想要为整个表创建单个索引时，才需要启用一个文本表。

一个索引还是几个索引？

第44页的『为表创建一个或若干个文本索引』说明您通过创建为整个表创建一个文本索引或创建几个索引（每个文本列一个）来使文本成为可搜索的。

- 要为整个表创建一个索引，您应在此刻运行 `ENABLE TEXT TABLE` 以创建一个空索引，然后运行 `ENABLE TEXT COLUMN`（参见下一步）几次（每次针对一个文本列）以填充这个索引。
- 或者，要为每个文本列创建一个单独的索引，可跳过为整个表创建一个索引的步骤，即跳过 `ENABLE TEXT TABLE`，运行 `ENABLE TEXT COLUMN` 几次来创建并填充各索引，每个文本列一个。

对于本示例，将不会为整个表创建一个索引。而是，继续进行下一步以为一个文本列创建一个索引。

6. 启用一个文本列以进行文本搜索

输入以下命令来启用 DB2 Text Extender，以在表 `db2tx.sample` 中的文本列 `mycolumn` 中进行搜索，并将名称 `myhandle` 指定给此命令创建的句柄列。

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

此命令创建一个文本索引。缺省值用于被索引的文档类型和索引特性。

7. 检查您正在创建的索引的状态

输入：

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

8. 脱离 DB2 Text Extender 命令行处理器

输入：

```
db2tx=>QUIT
```

9. 启动 DB2 命令行处理器

在操作系统提示符处，输入：

```
=>db2
```

10. 搜索文本

现在可搜索文档。尝试使用此 `SELECT` 命令，它会在您刚刚进行索引的文本中查找 `searchterm` 的所有出现情况：

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

创建索引和搜索的样本向导

提供了创建索引和搜索向导，以举例说明如何在选择的表或列上创建索引，以及如何对这些索引进行搜索。对于能够使用 DB2 Text Extender 搜索功能，对所选表列进行搜索所需的步骤，本样本向导将指导您如何完成

要启动样本向导，输入：

```
txwizard
```

可使用随样本向导一起提供的 Java 源代码来构建您自己的文本准备和搜索应用程序，或了解如何在 Java 环境内使用和调用 DB2 Text Extender 功能。

在 AIX 中，向导的 Java 源代码存储在以下目录中：

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

在 Windows NT 和 Windows 2000 中，它存储在以下目录中：

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

第4章 计划搜索需要

在开始第55页的『第5章 使文本可搜索』中所述的步骤之前，必须了解：

- 文档所具有的格式和代码页，以及它们所使用的语言
- 如何避免代码页问题
- 您将需要哪种搜索功能
- 您将需要的磁盘空间
- 文本索引的内容，以及您是想要 DB2 表的公用索引，还是每个表列一个单独索引。

本章描述收集此信息的原因及方式。

有几种索引类型可供选择：语言、精确和 Ngram。选择索引类型是很重要的。例如，若选择语言作为索引类型，则可以搜索单词变体以及搜索词条的同义词。索引类型还会影响创建索引性能和索引大小。您还可以通过对每个文本列创建数个索引，这些索引各自具有不同的索引类型，从而使用多个索引类型的搜索能力。

为何要对文本文档创建索引

快速信息检索系统并不依次扫描整个文本文档；那样将花费太长时间。它只是处理先前构建的文本索引。可以将文本索引看作是由从文本文档中抽取的重要词条组成的，每一词条都与关于包含该词条的文档的信息存储在一起。

文本索引仅包含有关的信息；不对一些不重要的单词（例如 "and"、"of" 和 "which"）创建索引。（未对 Ngram 索引进行任何停止字过滤。）DB2 Text Extender 使用这些单词的列表来防止对它们创建索引，这些单词被称为停止字。检索系统在整个索引中搜索所请求的词条，以找出哪些文本文档包含那些词条。

提示

若需要修改停止字列表，则仅在安装时修改一次。

每种语言的停止字列表都存储在可修改的文件中，（参见第53页的『修改停止字和缩写文件』），但是，因为整个系统只有一个可修改的文件，所以仅当首次设置 DB2 Text Extender 时更改它，且仅更改一次。若以后更改该文件，则现存的索引将不影响更改。

为何要对文本文档创建索引

作为一个示例，假设一些文档包含名称为“Now”的周刊名称。若从停止字中除去此单词，则将对它创建索引并可被将来的搜索找到。但是，在除去该停止字之前创建的任何索引将都不包含单词“now”，并且对它的搜索将不成功。

若您确实决定要更改停止字，并且想让此更改一直起作用，则必须重新创建所有索引。

索引过程包括两个步骤。第一步是在日志表中记录需要索引的文本文档。每当您在某列中插入、更新或删除文本文档时，通过 DB2 触发器，自动进行记录。

第二步是对列示在日志表中的文本文档创建索引。此步骤可能会定期执行。在该列中插入或更改的那些文档的词条被添加到索引中。从该列中删除的那些文档的词条被从索引中除去。

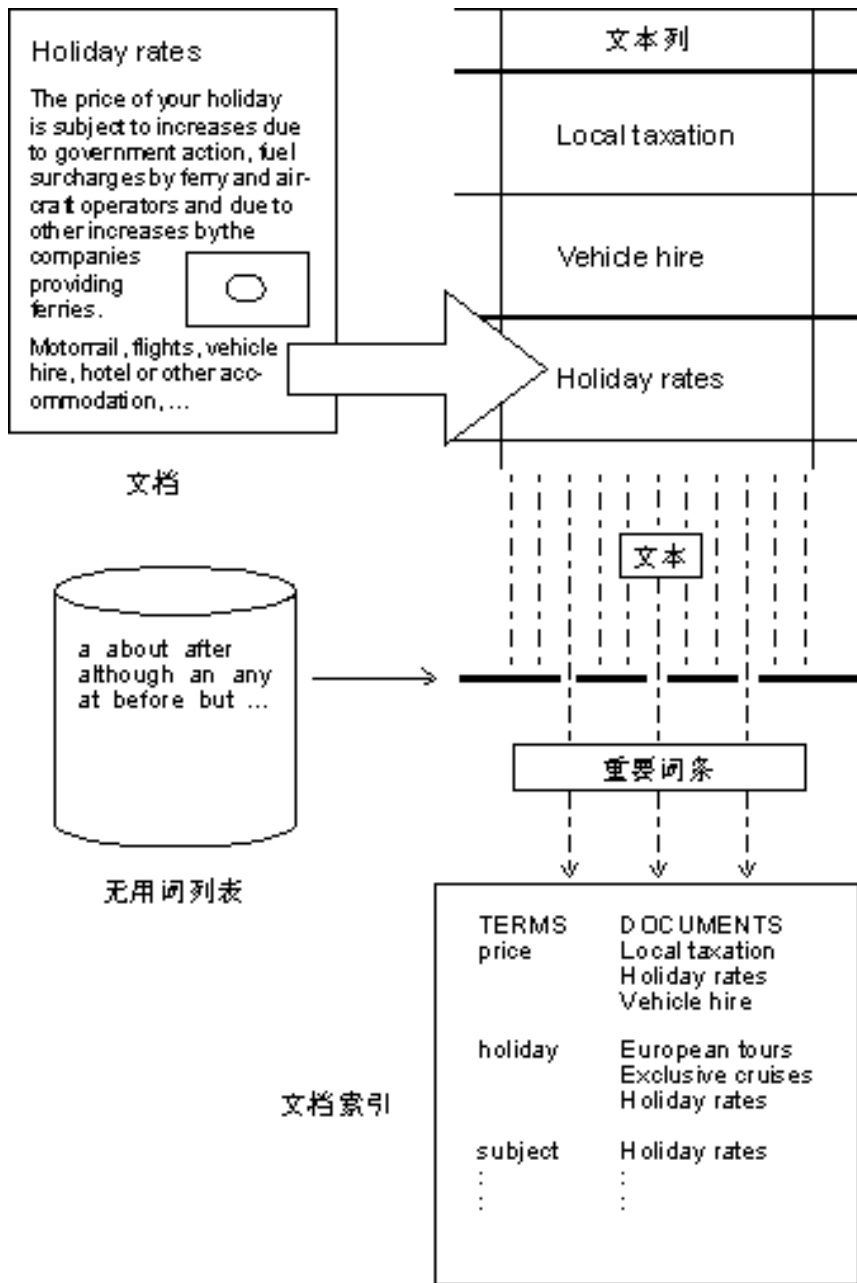


图 4. 仅对重要词条创建索引

受支持的文档格式

DB2 Text Extender 需要知道您要搜索的文本文档的格式（或类型），例如 HTML 或 ASCII。在创建文本文档的索引时需要此信息。

受支持的文本文档格式有：

HTML	超文本标记语言
XML	扩展标记语言
ASCII_SECTIONS	包含节的结构化 ASCII
TDS	平面 ASCII
AMI	AmiPro 体系结构版本 4
FFT	IBM 最终格式文本：文档内容体系结构
MSWORD	Microsoft Word，版本 5.0 和 5.5
RFT	IBM 可修订格式文本：文档内容体系结构
RTF	Microsoft RTF 文本格式，版本 1
WP5	WordPerfect（OS/2 和 Windows），版本 5.0、5.1 和 5.2

HTML 文档，特殊考虑事项

HTML 文档中的元音变音和区别字符的处理要视文档的代码页而定：

- 对于代码页 37、273、277、278、280、284、297、437、500、819、850、858、860、863、865、871、923、924 和 1252，下列情况适用：
 - 实体表示法用于元音变音和特殊字符，例如，
ä（对于 ä）。
 - 仅在代码页 819 (ASCII) 或 500 (EBCDIC) 上具有代码点的字符有效。
 - 在创建索引期间，如果文档的代码页既不是 819 也不是 500 的话，包含语言特定字符的文档（如 ä）会在单词识别方面出错。
 - 如果将文档添加至 Ngram 索引，则必须是使用代码页 819、500 或 UTF8 创建的索引。
- 对于所有其他单字节字符集代码页，下列情况适用：
 - 实体未被分解。
 - 特殊字符必须用语言特定的代码点编写。

XML 文档，特殊考虑事项

XML 文档必须包含一个编码字符串，而这个编码字符串是在语法分析期间使用的。下列编码是受支持的：

```

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-roeece,
    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

```

对于 XML 文档，缺省的代码页会被忽略。

如果将 XML 文档添加至 Ngram 索引，则该索引必须是使用代码页 UTF8 来创建的。如果该索引具有部分支持，且仅当创建索引时才使用叶子级部分标识符，则对于嵌套部分中的解析文件无任何支持。

使用未受支持的文档格式

对于不受支持的文档格式，指定一个数字 ID。有效值是 1 至 100。将此值作为源格式传送给将原始格式转换为 TDS 的用户出口。

如果在索引期间，有一个不是受支持的类型的文档，则 DB2 Text Extender 提供一个出口，将文档写入磁盘，并调用所提供的程序来将文本抽取到受支持的格式之一。用户出口必须同时在服务器配置文件 (DESSRV.INI) 和客户机配置文件 (DESCL.INI) 中注册。使用用户出口的名称更新 [DOCUMENTFORMAT] 段中的 USEREXIT 选项。

要启用用户出口，编辑下列 ASCII 文件：

```

Windows NT 和 Windows 2000:
%DMBMPATH%\instance%\DB2INSTANCE%\db2tx\descl.ini
%DMBMPATH%\instance%\DB2INSTANCE%\db2tx\txinsnn\dessrv.ini

```

文档格式

```
UNIX:  
$DB2TX_INSTOWNERHOMEDIR/db2tx/desc1.ini  
$DB2TX_INSTOWNERHOMEDIR/db2tx/txinsnnn/dessrv.ini
```

通过添加下列语句:

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

其中, <name_of_executable> 是用户出口的名称。可以指定全限定文件名, 或者, 如果将用户出口存储在 PATH 语句中的目录中, 则可以只指定文件名。

要调用用户出口, 使用以下语法:

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

sourcefilename

要由用户出口程序转换的文件。该文件名是全限定文件名, 且位于客户机简要表或服务器实例中的指定工作目录中。

targetfilename

包含用户出口的输出的文件。然后, 此文件会由 DB2 Text Extender 进行处理。该文件名是全限定文件名, 且指向客户机简要表或服务器实例中的指定工作目录。客户机简要表中的各项用于 API 调用 EhwGetMatches, 而服务器实例中的那些项将用于 API 调用 EhwUpdateIndex。

sourceccsid

源文件的代码页。这是缺省代码页。

targetccsid

DB2 Text Extender 所期望的代码页。该代码页为 850。

sourceformat

源文件的格式。这是缺省格式。

targetformat

DB2 Text Extender 所期望的文件格式。目前, 仅平面文件格式 (TDS) 或 ASCIISECTION (对于启用段的索引) 受支持。

用户出口必须能够返回下列值。

0 格式转换已成功。

>0 格式转换未成功。在创建索引期间, 错误信息被写至文档错误表。使用 desmsgix 命令来显示错误信息。

 语言

DB2 Text Extender 还需要知道文档是用哪种语言编写的，以便对发生的语言处理使用正确的词典。第53页的表5显示在启用文本列或外部文档时可指定的语言参数的列表。

 CCSID

提示

在指定 CCSID 何时启用文本列时，阅读第38页的『在存储和启用文本时避免代码页问题』

如果文档为下表其中一种 CCSID，则可对它们创建索引。

注：CCSID 861、865 和 4946 不受 DB2 UDB 支持。要对带有这些 CCSID 的文档创建索引，应将这些文档存储在具有二进制数据类型（BLOB 或 FOR BIT DATA）的列中。

EBCDIC

37	美国英语，加拿大英语
273	奥地利语，德语
277	丹麦语，挪威语
278	芬兰语，瑞典语
280	意大利语
284	西班牙语，拉丁美洲语
285	英国英语
297	法语
420	阿拉伯语
424	希伯来语
437	美国英语
500	国际拉丁语-1
871	冰岛语

CCSID

875	希腊语
1025	俄语
1112	拉脱维亚语
1122	爱沙尼亚语
1123	乌克兰语

ASCII

813 AIX、HP、SUN	希腊语
819 AIX、HP、SUN	拉丁语-1
850 AIX, OS/2	拉丁语-1
855 OS/2	保加利亚语
860 OS/2	葡萄牙语
861 参见注释	冰岛语
862 OS/2	希伯莱语
864 OS/2	阿拉伯语
863 OS/2	加拿大语
865, 参见注释	丹麦语, 挪威语
866 OS/2	俄语
869 OS/2	希腊语
915 AIX, OS/2, HP	俄语
916 AIX	希伯莱语
921 AIX、OS/2、WIN	拉脱维亚语
922 AIX、OS/2、WIN	爱沙尼亚语
1064 AIX	阿拉伯语
1089 AIX、HP	阿拉伯语
1124 AIX	乌克兰语
1125 OS/2	乌克兰语
1129	越南语
1130	越南语
1131	越南语

1250 WIN	克罗地亚语, 白俄罗斯语
1251 WIN	俄语
1252 WIN	拉丁语-1
1253 WIN	捷克语
1255 WIN	希伯莱语
1256 WIN	阿拉伯语
1257 WIN	希腊语
1258 WIN	越南语

DBCS

932 AIX, OS/2	日语, 组合 SBCS/DBCS
942 OS/2	日语, 组合 SBCS/DBCS
943 OS/2, WIN	日语, 组合 SBCS/DBCS
5039 HP	日语, 组合 SBCS/DBCS
954 AIX, HP, SUN	日语
949 OS/2	韩语
970 AIX, HP, SUN	韩语
1363 WIN	韩语
948 OS/2	中文(繁体), 组合 SBCS/DBCS
950 AIX, HP, OS/2, SUN, WIN	中文(繁体), 组合 SBCS/DBCS
964 AIX, HP, SUN	中文(繁体), 组合 SBCS/DBCS
1381 OS/2, WIN	中文(简体), 组合 SBCS/DBCS
1383 AIX, HP, SUN	中文(简体), 组合 SBCS/DBCS
1386 AIX, OS/2, WIN	中文(简体), 组合 SBCS/DBCS
4946, 参见注释	拉丁语- 1 (CP850)
5039 HP	日语

UNICODE

1208	UTF8
13488	UCS2

在存储和启用文本时避免代码页问题

下列区域具有代码页设置:

- 活动应用程序环境
- 每个文档
- 每个 DB2 数据库
- 每个 DB2 Text Extender 索引

在带有字符数据类型（如 VARCHAR 和 CLOB）的 DB2 数据库列中存储文档时，DB2 假定每个文档与活动的应用程序环境具有相同的代码页，且将文档从该代码页转换为数据库的代码页。数据库的代码页或者已经与活动应用程序环境代码页相同（不会进行转换），或者它是在创建数据库时指定的代码页，与应用程序代码页有所不同。（会发生转换）。

在带有二进制数据类型（如 BLOB 或 FOR BIT DATA）列的 DB2 数据库中存储数据时，DB2 不转换数据，且这些文档仍为它们原来的 CCSID。

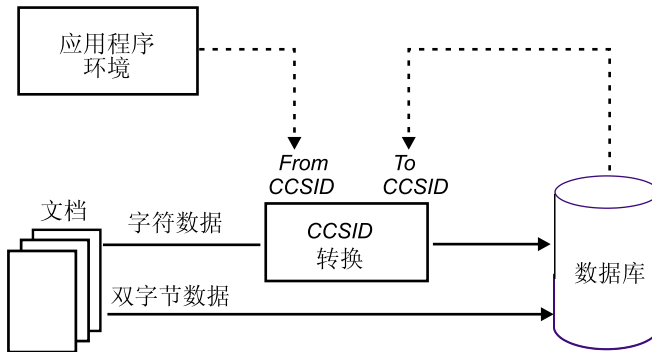


图 5. DB2 设置数据库文档的 CCSID 的方式

在启用文本列以供 DB2 Text Extender 使用时（即，在使用 ENABLE TEXT COLUMN 命令来创建索引以供搜索时），索引的代码页被设置为数据库的代码页（缺省情况），或可使用 CHANGE TEXT CONFIGURATION 命令设置为当前缺省值，又或者设置为在 ENABLE TEXT COLUMN 命令中指定的代码页。

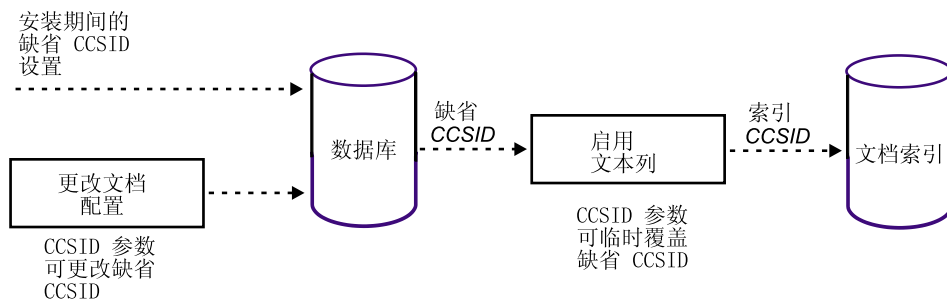


图 6. DB2 设置文本索引的 CCSID 的方式

搜索期间，将使用数据库的 CCSID 来解释搜索字符串的 CCSID。

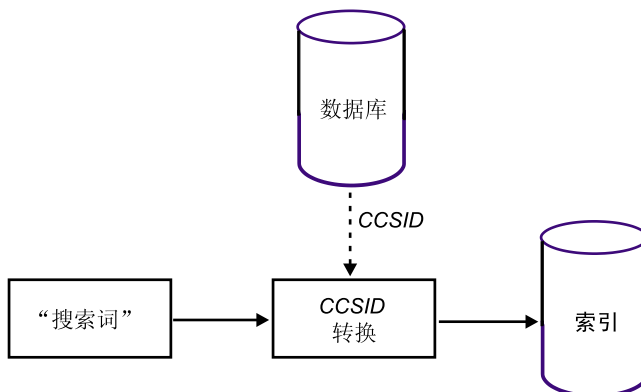


图 7. 使用数据库 CCSID 进行搜索

此处介绍了如何检查代码页设置:

- 要检查数据库代码页，使用 DB2 命令：
db2 get db cfg
- 要检查缺省索引代码页，使用 DB2 Text Extender 命令：
db2tx get text cfg

要避免代码页问题，最重要的是正确地协调这些代码页设置:

- **示例 1:** 环境 850、文档 850、数据库 850、索引 850

正确。 DB2 正确地假定文档与环境具有相同的代码页，而由于数据库与环境具有相同代码页，所以 DB2 不进行任何转换，且会将代码页为 850 的文档存储在数据库中。然后，在启用文本列时，采用缺省的代码页设置（850，数据库代码页），文档会被正确地索引到 850 代码页索引中。

- **示例 2:** 环境 1252、文档 1252、数据库 850、索引 850

正确。 DB2 正确地假定文档与环境具有相同代码页，而在数据库中存储该文档时会将代码页从 1250 更改为 850。然后，在启用文本列时，采用缺省的代码页设置 850（数据库代码页），文档会被正确地索引到 850 代码页索引中。

- **示例 3:** 环境 1252、文档 850、数据库 850、（索引为任意值）

错误。 DB2 不正确地假定该文档（代码页 850）与环境（代码页 1252）具有相同的代码页，而在数据库中存储该文档时进行了不正确的代码页转换（从 1252 转换为 850）。

- **示例 4:** 环境 1252、文档 850、数据库 1252、索引 850

正确。 DB2 不正确地假定文档与环境具有相同的代码页，但由于环境代码页与数据库代码页相同，所以 DB2 不进行任何转换，且会文档存储在代码页为 850 的数据库中。但是，稍后在启用文本列时，必须将文档的代码页指定为 850，以使文档被正确地索引到 850 代码页索引中。

- **示例 5:** 环境 1252、文档 1252、数据库 850、索引 1252

潜在错误。 DB2 正确地假定文档与环境具有相同的代码页（即 1252代码页），且在数据库中存储该文档时将文档转换为代码页 850。然后，如果对 850 代码页索引启用文本列时将文档的原始代码页指定为 1252，则会发生潜在错误。正确的操作应是采用缺省代码页设置 850（数据库代码页）。

搜索类型

可对包含要搜索的文本的列指定下列其中一个索引类型和各种选项：*语言*、*精确*和 *Ngram*。在您准备任何这样的列以供 DB2 Text Extender 使用之前，必须确定创建哪种索引类型 有关每种索引类型如何影响语言处理的更详细说明，参阅第235页的『第15章 对语言索引和精确索引的语言处理』。

摘要

- 对于**搜索语言文字变化**，使用**语言索引**。
根据标准和原始内容及词典的使用来查找单词变化；使用最少的磁盘空间。
- 对于**进行精确搜索**，使用**精确索引**。
查找与输入项完全相同的项；进行索引和搜索会更加迅速；使用更多的磁盘空间。如果是标准化的话，搜索是区分大小写的。
- 对于**搜索字符变化**，使用 **Ngram 索引**。
查找单词即使是拼写不正确的话；如果使用 CASE_ENABLED 选项以允许区分大小写搜索，则索引会使用更多空间，而搜索会花更长的时间。
- 对于在 **DBCS 文档中进行搜索**，使用 **Ngram 索引**。
对于 DBCS 文档，这是唯一选择，但该索引还可用于类型为 TDS 的 SBCS 文档。

DB2 Text Extender 提供了大量的搜索选项，尽管并非所有选项都适用于所有索引类型。在决定使用哪种索引类型之前，参见第207页的表8和第207页的表9。

语言搜索

对于语言索引，为创建索引分析每个文档的文本时，将应用语言处理。这意味着在将单词存储到索引中之前，这些单词将缩减为基本格式；例如，词条“mice”在索引中存储为 mouse。

对于针对语言索引的查询，在搜索文本索引之前，相同的语言处理适用于该搜索词条。因此，若要搜索“mice”，则在开始搜索之前，将它缩减为其基本格式 mouse。第235页的表18概述了当使用语言索引时如何抽取词条来创建索引。

这种索引类型的优点是搜索词条的任何变体与在某个已索引的文本文档中发生的其他任何变体相匹配。搜索词条 mouse 与文档词条“mouse”、“mice”和“MICE”（大写字母）等相匹配。类似地，搜索词条 Mice 也匹配同一文档词条。

此索引类型需要的磁盘空间最少。但是，创建索引和搜索会比精确索引花的时间长。

可用的语言处理类型取决于文档的语言。以下是这些类型的列表：

搜索类型

- 单词和句子分开。
- 句首处理。
- 取消连字符。
- 将词条规范化为标准格式，在标准格式中没有大写字母，并且将重读字母（如“ü”）更改为不重读的格式。例如，德语单词“Tür”（门）在索引后成为 tuer。
- 将词条缩减为其基本格式。例如，“bought”被索引成 buy，“mice”被索引成 mouse。

提示

单词分段（用通配符屏蔽了的单词）不能缩减至基本格式。因此，若搜索 swu%，则将找不到单词“swum”，因为在索引中它已缩减至其基本格式。要找到它，必须搜索 swi%。

未正确拼写的单词的变化不能缩减至基本格式。例如，将德语单词 röstete（它被正确索引至其基本格式）用作动词 rösten。搜索项 röstete 或 rösteten 被正确地规范化为基本格式 rösten，该项可在索引中找到。搜索项 rostete 不能被规范化为 rösten，所以在索引中找不到该项。

- 单词字符分解，这里，德语单词“Wetterbericht”（天气预报）不仅被索引成 wetterbericht，还被索引成 wetter 和 bericht。
- 停止字过滤，它将不对无关的词条创建索引。将对“A report about all animals”创建索引 report 和 animal。
- 部分语句过滤，它类似于停止字过滤；它只索引名词、动词和形容词。“I drive my car quickly”被索引成 drive 和 car。单词“I”和“my”被作为停止字除去，另外，副词“quickly”也被部分语句过滤所除去。

精确搜索

在精确索引中，文本文档中的词条在索引后与它在文档中时完全一样。例如，搜索词条 mouse 可以找到“mouse”，但是，找不到“mice”，也找不到“Mouse”；精确索引中的搜索是区分大小写的。

在查询中，对所有查询词条应用同一处理，然后再将查询词条与索引中所找到的词条进行比较。这意味着所找到的词条与搜索词条完全相同。可以使用屏蔽字符来扩展搜索范围；例如，搜索词条 experiment% 可以找到“experimental”、“experimented”等等。

第236页的表19 给出一些示例来说明当使用精确索引时，如何从文档文本中抽取词条来创建索引。

这种索引类型的优点是搜索更精确，并且索引和检索更快。因为对每个词条的各种不同的格式和拼法都创建了索引，因此，它所需要的磁盘空间要比语言索引多。

用来对精确索引的文本文档创建索引的语言处理有：

单词和句子分开

停止字过滤。

在 DBCS 文档进行模糊搜索或搜索

Ngram 索引通过对字符集进行语法分析来分析文本。这种分析并不根据词典来进行。

若文本中包含 DBCS 字符，则必须使用 Ngram 索引。没有其他索引类型支持 DBCS 字符。

此索引类型支持“模糊”搜索，这意味着可以找到与所指定的搜索词条相似的字符串。例如，对 Extender 进行搜索可以找到错误输入的单词 Extendrrs。还可以指定所需的相似程度。

注：即使使用模糊搜索，前三个字符也必须匹配。

要在 Ngram 索引中执行区分大小写的搜索，仅在查询中指定 PRECISE FORM OF 关键字并不足够。这是因为 Ngram 索引通常对索引的字符不区分大小写。然而，当创建了索引后，可通过指定 CASE_ENABLED 选项来使 Ngram 索引区分大小写。然后，在查询中指定 PRECISE FORM OF 关键字。

当使用 CASE_ENABLED 选项时，索引将需要更多空间，并且搜索可能要花更长时间。

参见第35页的『CCSID』以获取 Ngram 索引支持的 CCSID 列表。Ngram 索引支持本机 CCSID列表。对于所有其他 CCSID，数据库从此 CCSID 映射至 UTF8。

尽管 Ngram 索引类型是设计用来对 DBCS 文档创建索引的，但它也可用于 SBCS 文档。然而，它仅支持 TDS 文档。

还要注意，并不支持所有的搜索语法选项。有关详情，参见第201页的『第12章 搜索变元的语法』中对规则和限制条件的概述。

更改索引类型

如果决定使用的索引类型不太合适，首先通过禁用文本列或文本表来删除该索引，然后重新启用该文本列或文本表来重新创建索引。

为表创建一个或若干个文本索引

第55页的『第5章 使文本可搜索』描述了如何准备表，以便在其中搜索文本。然而，在做这个准备之前，必须确定是为表中的所有索引文本列创建一个公共的文本索引，还是为每个索引文本列都创建一个文本索引。为每个文本列创建一个单独的索引的表被称为多索引表。

提示

若要对外部文件创建索引（参见第69页的『启用外部文本文件』），则相关联的表必须是多索引表。

使用多个索引有下列优点：

- 为每个文本列创建不同的索引类型

对于与文本列相关联的特性，这具有很大的灵活性，例如，何时对文本列的索引进行定期更新，以及该索引存储在哪一个目录中。有关这些特性的说明，参见第135页的『ENABLE TEXT COLUMN』。

- 在不同时间创建索引列

创建索引是一种消耗时间和资源的活动。在具有多索引表的情况下，可通过在不同时间对各列创建索引来在一段时间内安排此活动。

如果不需要由多索引表提供的灵活性，公共索引使得 DB2 Text Extender 更易于维护；启用文本表时，设置创建索引参数，它们将用作该表所有文本列的缺省值。此外，如果需要禁用这些列，可通过使用一个命令禁用文本表来做到这一点。

计算索引的大小

索引所需要的磁盘空间取决于要索引的数据的大小和类型以及索引类型。用处理器编写的文本文档需要的空间较小，因为它们的内容很大一部分是控制字符。作为标准，对于语言索引，保留的磁盘空间大约为正在创建索引的文档大小的 0.7 倍，然后将此大小乘以 2，依据它为重组索引保留临时空间。对于 Ngram 索引，您将需要大约两倍于磁盘空间的空间。

若有数个大型索引，则应将它们存储在不同的磁盘设备上，特别是索引更新或搜索期间要对索引进行并行存取时更应如此。

更新索引

当将文本文档添加至数据库时，或更改数据库中的现存文档时，必须创建文档的索引，才能保持索引内容与数据库内容同步。从数据库中删除文本文档时，必须从索引中除去它的项。

有关哪些文档是新的、更改过的以及已删除的文档的信息，由触发器自动存储在日志表中。在下次执行索引更新时，将创建在日志表中列示的文档的索引。

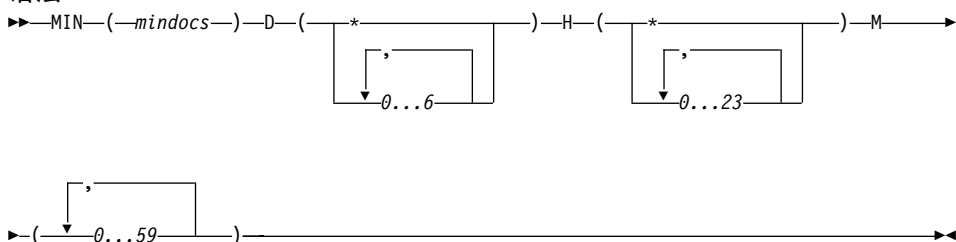
UPDATE INDEX 命令允许您一经请示就立即更新索引。

但是在一般情况下，将按照在环境变量 DB2TXUPDATEFREQ 中指定的时间间隔自动更新索引。该环境变量确定缺省设置。当使用 ENABLE TEXT COLUMN 或 ENABLE TEXT TABLE 命令来创建索引时，会覆盖缺省设置。可以使用 CHANGE INDEX SETTINGS 命令来更改现存索引的更新频率。

根据要进行更新的时间以及必须排队的最少文本文档数，来指定索引更新的频率。如果在指定的日期和时间，在日志表中没有足够的文档，则不更新索引。

应当仔细地计划定期创建索引；创建文本文档的索引是一项消耗时间和资源的任务。所花费的时间取决于多种因素，例如，从前一次索引更新以来已添加或更改了多少文本文档，文档的大小以及处理器的处理能力有多强。

语法



MIN mindocs

在可以更新索引之前，必须排队的文本文档的最小数目。

D 在星期几更新索引:

* 每天

0 星期天

更新索引

- 1 星期一
- 2 星期二
- 3 星期三
- 4 星期四
- 5 星期五
- 6 星期六

H 在指定日期的什么时间更新索引:

* 每小时

0...23 在指定的时间

M 在指定小时的第多少分钟更新索引:

0...59 在指定的分钟

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

如果在星期一至星期五的 12:00:00 或 15:00:00 至少有 100 个文本文档在排队, 则更新索引。

可以组合几种频率规范:

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

同前面一样, 索引更新安排在星期一至星期五的 12:00:00 和 15:00:00, 但是, 另外还在每天的 22:00:00 更新索引, 即使在日志表中只有一个文本文档。

维护多节点环境中的索引

若启用的文本列属于某个表 (该表是多节点节点组的一部分), 则指定的索引目录必须可在所有物理节点上获得。若使用在文本配置中指定的缺省目录, 则确保该路径可在该节点组的所有节点上获得。若这样不方便, 则可以在 `ENABLE TEXT COLUMN` 命令中为每个节点指定一个特定路径。

若更改节点组的节点配置, 该节点组包含为 DB2 Text Extender 启用的表, 则必须重新索引该表可通过使用命令 `TXNCHECK` 检查是否已更改节点配置。

提示

若正在使用 DB2 UDB 扩充企业版，并且已将表分布在几个节点当中，则这些节点的系统时间可能有些差异（例如，其中一个节点处于夏令时间）。要防止出现问题，应确保节点的系统时间设置之间的差异在 DB2 的 MAX_TIME_DIFF 设置范围以内。在 Windows NT 和 Windows 2000 中，使用 NET TIME 命令来使系统时间设置同步。

使用结构文档（部分支持）

部分支持允许您对结构文档中的特定部分（例如，在 title、author 或 description 中）进行索引和搜索。这些文档可为 XML 或 HTML 格式，或带有类似 HTML 标记的平面文件文档。可在文档模型中定义 markup 标记及其相应的部分名。该文档模型定义将对文档中的哪些段进行索引，以便可对其进行搜索。这些部分名都是描述性的名称，用于对该部分进行查询。

要使部分支持可用，必须在启用包含这些文档的文本列时指定 INDEXPROPERTY SECTIONS_ENABLED。

文档模型文件列示了所有定义的服务器实例文档模型。创建服务器实例时，样本文档模型文件 DESMODEL.INI 是在服务器实例子目录中自动创建的。该文件在 ASCII 代码页中。

文档模型信息被复制至索引目录中。如果在创建索引之后更改了服务器实例的文档模型文件，它将不会影响到已创建索引的部分支持。

对带有部分支持的索引进行的搜索，例如，要在部分 Author 中搜索 McDaniel，可能会类似于如下内容。该部分（在本示例中为 Author）总是以模型名为前缀。

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

属性部分

部分可为不同类型。普通文本部分无任何类型。带有说明的类型的部分被称为属性部分。受支持的属性类型为：

- DATE
- TIME

结构文档（部分支持）

- FLOAT
- INTEGER

可使用数值范围来搜索包含属性部分的文档。该部分必须有适当的部分类型。在以下示例中，部分 ABC 必须具有受支持的查询类型 TIME:

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

一个部分可包含多个属性值。用空格或换行将各个值隔开。不能嵌套属性部分。

受支持的日期格式

可包括日期，例如，文档模型中受支持部分列表中的创建日期或发布日期。下列日期格式是受支持的:

表 3. 受支持的日期格式

格式名	格式	示例
ISO	YYYY-MM-DD	1986-12-25 100-12-25 10123-2-17
	DD-MM-YYYY	01-10-1999 5-8-1990
IBM 美国标准 (USA)	MM/DD/YYYY	08/18/1999 5/13/1977
	YYYY/MM/DD	1999/08/18
IBM 欧洲标准 (EUR)	DD.MM.YYYY	20.10.1999 1.5.1980

对带有对在特定日期发布的所有文档的部分支持的索引可能类似如下所示:

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

受支持的时间格式

下列日期格式是受支持的:

表 4. 受支持的时间格式

格式名	格式	示例
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

可省略小时的前导零。下列范围之外的值无效:

小时	分钟	秒
0-23	0-59	0-59

受支持的浮点格式

下列格式是受支持的:

- 范围 1.175494351e-38F 至 3.402823466e+38F 内的数字序列
- 使用 e 或 E 的科学表示法。

使用空格或换行来隔开浮点值。

以下是浮点值的正确和不正确格式的示例:

正确的格式	不正确的格式
1000	1 000
1	1,000
1.0e3	e3
1.0E3	
1E3	
.1E4	
0.1E4	

受支持的整数格式

范围 -2147483648 至 2147483647 内的整数是受支持的。使用空格来隔开整数。这意味着类型为 INTEGER 的部分中的字符“42 234”解析为整数值“42”和“234”，而不是“42234”。

结构文档（部分支持）

以下是整数值的正确和不正确格式的示例:

正确的格式	不正确的格式
1000	1.000 1,000 1 000
1000000	100.000 100,00 100 000

注释

语言特定的分隔符和货币格式不受支持。

平面文件和 HTML 文档

对于平面文件，这些部分是使用类似于 HTML 的标记来标记的，如 <title> 和 <subject>。带有标记的部分的文档可能类似如下所示:

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

平面文件的文档模型文件或 HTML 文档可能类似如下内容。这些模型名和部分名仅可包含 A-Z、a-z 和 0-9。模型名始终都是区分大小写的。部分名可是区分大小写的或不区分大小写的；该设置是在创建模型时设置的。

```
;list of document models
;model always starts with 'modelname' and the name of the model
[MODELS]
modelname=sample
modelname=sample2
modelname=sample3

; a 'sample' document model definition
; left - section name identifier
; right - section name tag
[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

```

如果文档包含文档模型中未定义的已标记部分，则该部分的内容被包括在先前定义的部分中，以供进行索引和搜索。例如，某个文档包含下列标记的部分：

```

<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.

```

文档模型 `book` 被定义为：

```

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

```

`<subject>` 部分未包括在 `book` 文档模型中。在对文档进行索引时，`subject` 部分的内容是随 `title` 部分的内容一起索引的。它们还可供在 `title` 部分中搜索使用。

如果在创建索引时指定了模型列表，则缺省模型是列表中的首个模型。可使用 `desmodix` 命令来更改缺省模型。

XML 文档

对于启用部分的索引，必须正确地构建 XML 文档，该文档应包含一个根元素。根元素的名称必须与定义的模型名之一相同，且大小写必须匹配。文档模型文件中的模型说明必须是某个文档模型的子集，该文档模型是在文档的 DTD（文档类型定义）文件中定义的。

模型说明必须以根元素开头。对于想要用作部分的每个 XML 元素，必须将该元素的完整分层结构包括在模型说明中。如果部分的类型为日期，则此部分必须是文档模型树中的叶子。属性部分的嵌套不受支持。

XML 文档的模型说明可能类似如下所示：

```

; list of document models
[MODELS]
modelname = LETTER
; sample for XML model definition

```

结构文档（部分支持）

```
; left-hand side = section name identifier encoding whole path
; right-hand side = section name tags specifying tag for each
;                 element of the path through the tree down to
;                 the specified node. Tag delimiter is /.
[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
```

XML 文档可能类似如下内容。它还显示了模型中未定义的部分将如何进行索引。

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

<LETTER>
  <HEADER>This tag has been skipped in the definition, to this text will
           be added to the section named LETTER
  </HEADER>
  <DATE>
    01.01.2000  03.02.2000
  </DATE>
  <ADDRESS>
    Text will be added to the section named LETTER/address.
    <CITY>
      Text will be added to section named LETTER/address/City.
    </CITY>
  </ADDRESS>
  <CONTENT>
    Text will be added to the section named LETTER/Content.

    <NOSECTION>Text will be added to the section named LETTER/Content
                because NOSECTION is not defined.
    </NOSECTION>
    <GREETINGS>
      Text will be added to section named LETTER/Content/Greetings.
    </GREETINGS>
  </CONTENT>
</LETTER>
```

词典、停止字列表、缩写列表和语言参数

第53页的表5显示受支持的语言，并显示词典文件、停止字列表文件和缩写列表文件的名称。词典文件使用二进制格式，不能更改。停止字文件和缩写文件（如果有的话）使用平面文件格式，可以更改。如果更改其中的任何文件，确保使用语言的代码页。

此表还显示了在启用文本列或外部文档时必须指定哪个语言参数。这告知了 DB2 Text Extender 文档是用哪种语言编写的，以对出现的语言处理使用正确的词典。

表 5. 用于各种语言的语言功能

语言	文件名	LANGUAGE 参数	代码页
阿拉伯语	arabic	ARABIC	864
巴西葡萄牙语	brazil	BRAZILIAN	850
加拿大法语	canadien	CAN_FRENCH	850
加泰隆语	catala	CATALAN	850
丹麦语	dansk	DANISH	850
荷兰语	nederlnd	DUTCH	850
芬兰语	suomi	FINNISH	850
法语	francais	FRENCH	850
德语	deutsch	GERMAN	850
希伯来语	hebrew	HEBREW	862
冰岛语	islensk	ICELANDIC	850
意大利语	italiano	ITALIAN	850
挪威 Bokmal 语	norbook	BM_NORWEGIAN	850
挪威 Nynorsk 语	norntn	NN_NORWEGIAN	850
葡萄牙语	portugal	PORTUGUESE	850
俄语	russian	RUSSIAN	866
西班牙语	espana	SPANISH	850
瑞典语	svensk	SWEDISH	850
瑞士德语	dschweiz	SWISS_GERMAN	850
泰语	thai	THAI	850
英国英语	uk	UK_ENGLISH	850
美国英语	us	US_ENGLISH	850

这些文件是根据它们的扩展名来区分的。

内容	扩展名
词典	DIC
停止字列表	STW
缩写列表	ABR

修改停止字和缩写文件

每种语言都有一个停止字文件和一个缩写文件。要了解编辑这些文件的含义，参见第29页的『为何要对文本文档创建索引』。

提示

在开始编辑这些文件之一前，应创建一个副本。

OS/2 和 Windows 系统上的停止字和缩写文件位于：

drive:\dmb\db2tx\dict

在 AIX、HP-UX 和 SUN-Solaris 系统上，停止字和缩写文件位于：

DB2TX_INSTOWNER/db2tx/dicts

使用您自己的编辑器来编辑这些文件。它们使用 CCSID 850，因此在开始之前要确保您的应用程序 CCSID 也设置为 850。

除去想要创建索引的字和缩写。添加不想创建索引的字。

第5章 使文本可搜索

第25页的『第3章 入门』通过由 DB2 Text Extender 引导您了解一个简单的示例来熟悉使文本成为可搜索的过程。本章会更详细地描述使文本可搜索的过程，并描述您在开始之前应考虑的所有方面。

使文本可搜索的步骤有：

1. 作完善的准备工作
2. 启动 DB2 Text Extender 命令行处理器
3. 连接数据库
4. 启用数据库以进行文本搜索
5. 启用文本表以进行文本搜索用（如果对每个文本列创建一个索引，则不需要此步骤）
6. 启用一个文本列以进行文本搜索

在使文本可搜索之前作准备

提示

仔细地阅读本节。它列示了在使文本可搜索之前需要了解的选项。

- **对于 UNIX 用户，设置简要表**

如果您不是 DB2 Text Extender 实例所有者，请参考第16页的『配置』以了解如何设置简要表

- **要为整个表创建一个索引吗？**

必须决定是为整个文本表创建一个索引，还是为每个文本列创建一个单独的索引。第44页的『为表创建一个或若干个文本索引』将帮助您作出决定。

- **熟悉您的文档**

在使文档成为可搜索时，必须指定它们的 CCSID、语言和文本格式。有关更多信息，参见第29页的『第4章 计划搜索需要』。

- **决定所需的文本索引类型**

您所需要的索引类型是由想要进行的搜索种类（精确、模糊，等等）和文档是 SBCS 还是 DBCS 来确定的。可在第29页的『第4章 计划搜索需要』中找到更多信息。

在使文本可搜索之前作准备

- **决定存储索引的位置**

在使文档可搜索时，DB2 Text Extender 会创建一个文本索引。必须指定想要存储索引的目录。确保该处将有足够的磁盘空间（参见第44页的『计算索引的大小』）。

- **检查缺省数据库名**

DB2 环境变量 DB2DBDFT 中的缺省数据库名（参见第17页的『环境变量』）是 DB2 Text Extender 使用的名称（如果在使文本可搜索时未指定数据库名的话）。

- **设置文本配置**

文本配置确定了下列各项的缺省设置：索引 CCSID、文档语言、文档格式、索引类型、索引更新频率、表空间名和索引目录。

可在使文本可搜索时覆盖这些设置，但是预先正确地设置这些缺省设置会更加方便。安装 DB2 Text Extender 时的初始文本配置设置在第18页的『文本配置设置』中作了描述。要更改安装设置并设置您自己的缺省值，使用第125页的『CHANGE TEXT CONFIGURATION』。

- **设置部分支持**

如果需要将搜索限制在文档的某个特定部分，阅读第47页的『使用结构文档（部分支持）』以了解如何在文档模型文件中指定模型。

- **修改停止字和缩写列表**

阅读第29页的『为何要对文本文档创建索引』和第53页的『修改停止字和缩写文件』以了解停止字列表和缩写列表的概念，并决定是否在开始创建索引之前修改它们。

一旦已经收集了信息并作出了第55页的『在使文本可搜索之前作准备』中所述的决定，就可以准备使文本可搜索的过程了。

启动 DB2 Text Extender 命令行处理器

摘要

何时	可选。在每个对话开始时。
命令	db2tx
权限	任何

可在操作系统提示符处输入 DB2 Text Extender 客户机命令。这些命令类似于 DB2 命令，但是，在这些命令前面不是加上 db2，而是加上 db2tx。

为每个 DB2 Text Extender 命令加上前缀 db2tx 的另一个方法是启动 DB2 Text Extender 命令行处理器。数据库连接的一个优点就是不会在每个命令后断开连接；您仍然会处于连接状态。（如果没有命令行处理器，您会在每次执行 db2tx 命令时自动重新连接，您必须确保想要连接的数据库的名称是在 DB2DBDFT 环境变量中设置的。）

1. （仅 Windows）：启动 DB2 命令行处理器

在操作系统提示符处，输入：

```
=>db2 cmd
```

2. 启动 DB2 Text Extender 命令行处理器

在 UNIX 系统中，在操作系统提示符处输入：

```
=>db2tx
```

在 Windows 系统中，在 DB2 提示符处输入：

```
db2=>db2tx
```

会显示 db2tx 提示符：

```
db2tx=>
```

而所有后续命令都被解释为 DB2 Text Extender 命令。

要脱离此方式，输入：

```
db2tx=>quit
```

对于 Windows 95 和 Windows 98，可以使用上述的 db2tx 命令，或者可以单击“开始”按钮，指向“程序”，指向“DB2 客户机 Windows 95/98 版”，然后单击“DB2 命令窗口”。

如果脱离此步骤，可通过为 DB2 Text Extender 命令加上前缀 db2tx 来直接从操作系统发出这些命令。这是从操作系统提示符处发出的命令的示例：

```
=>db2tx enable database
```

提示

若正从客户机上工作，则必须处于 DB2 Text Extender 命令环境中且使用连接语句的完整格式，并指定用户 ID 和口令。

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

启动 DB2 Text Extender 命令行处理器

命令行处理器帮助

要显示命令列表，输入：

```
db2tx ?
```

要显示个别命令的语法，输入：

```
db2tx ? command
```

例如：

```
db2tx ? CHANGE TEXT CONFIGURATION
```

连接至数据库

摘要

何时	可选。要启动与数据库的对话（该数据库不是在 DB2DBDFT 环境变量中指定的缺省值）。
命令	CONNECT
权限	对数据库的 CONNECT

在可在 DB2 Text Extender 对话中发出进一步的命令之前，必须连接至数据库。如果在未与数据库连接的情况下就发出命令，DB2 Text Extender 会自动将您连接至 DB2DBDFT 环境变量中指定的缺省数据库。可通过使用 DB2 Text Extender CONNECT TO 命令显式地连接至数据库。

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

会显示以下内容（如 AIX 上所示）：

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

注：如果随后通过为 DB2 Text Extender 命令加上前缀 db2tx 在命令行处理器外部发出了该命令，则当前数据库连接会丢失，而创建与缺省数据库的新连接。

若正在使用 OS/2 客户机，并且这是启动系统之后第一次与数据库相连，则“用户简要表管理”(UPM)提示您输入用户 ID 和口令。若您已经注册，则 UPM 采用来自此对话的用户 ID 和口令，并尝试与服务器上的远程数据库相连。

启用数据库

摘要

何时	对包含要搜索的文本列的每个数据库进行一次。
命令	db2tx=>ENABLE DATABASE
权限	SYSADM 或 DBADM

此命令不带任何参数。它准备已连接的数据库以供 DB2 Text Extender 使用。

此命令还对 DB2 说明了 DB2 Text Extender 的函数和 DB2 Text Extender 单值类型。这些是稍后搜索文本时所使用的 SQL 函数。在第185页的『第11章 搜索函数』中对它们进行了描述。这些说明适用于所有将来的对话。

将创建一个目录视图 TEXTINDEXES，它跟踪启用的文本列。参见第106页的『使用 DB2 Text Extender 目录视图』。

此命令为数据库创建文本配置信息，这包含索引、文本和处理特性的缺省值。在第18页的『文本配置设置』中对它们进行了描述。

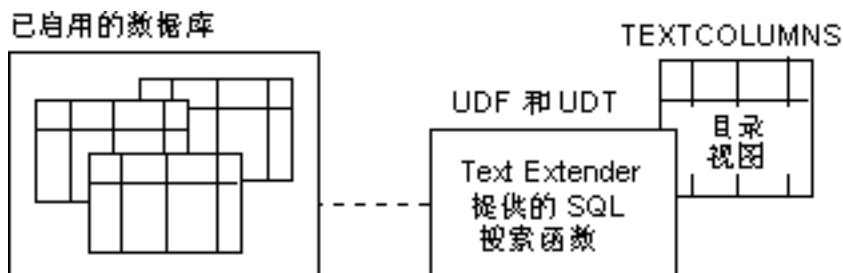


图 8. 启用数据库

一旦启用了数据库，该数据库就将一直保持启用状态，直到被禁用。要取消由 ENABLE DATABASE 所作的更改，请参考第109页的『禁用数据库』。

提示

若使用环境变量 `DB2TX_INSTOWNER`，则在启用数据库之前，必须将该环境变量设置为实例拥有者的名称。这对于 UNIX 用户尤其重要，因为在 UNIX 中此变量是缺省设置的。

如果稍后决定卸下启用的数据库，应首先禁用它以确保说明的 DB2 Text Extender 函数、目录视图等等已被除去。

启用文本表（可选）

摘要

何时	可选。使用此命令以仅为表中的所有文本列创建一个公共索引。参见第44页的『为表创建一个或若干个文本索引』。
命令	<code>db2tx=>ENABLE TEXT TABLE ...</code> （参见示例）
权限	对表的 ALTER、SELECT、UPDATE

此步骤确定是表中的所有文本列具有一个公共索引，还是具有多个索引，也就是说，每个文本列都具有单独的索引。

要获得公共索引，运行 `ENABLE TEXT TABLE`，然后对每个文本列运行 `ENABLE TEXT COLUMN`。要获得单独的索引，可跳过 `ENABLE TEXT TABLE`，并对每个文本列只运行 `ENABLE TEXT COLUMN`。这显示在图9和第63页的图10中。

在此步骤期间，DB2 Text Extender 创建了一个空文本索引，该文本索引对于所有后续启用的文本列是公共的 指定索引类型、更新索引的频率以及存储索引的目录。未指定的任何参数的缺省值都取自于文本配置设置。

提示

如果某个设置（如索引类型）对于大多数文本列都相同的话，则使用文本配置信息来指定缺省设置。参见第18页的『文本配置设置』。

此步骤还创建一个空的日志表，用于记录在表中添加、更改或删除的文档。创建触发器以保持更新日志表。

不能对表（该表已包含对 DB2 Text Extender 启用了的文本列）运行 `ENABLE TEXT TABLE`。

要删除由 `ENABLE TEXT TABLE` 创建的索引，参见第109页的『禁用文本表』。

提示

若稍后决定卸下已启用的文本表，则应首先禁用它以确保除去索引、日志表，等等。

示例

以下示例启用了文本表 `DB2TX.MYTABLE`:

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

索引特性的缺省值取自于文本配置设置。

下一示例显式地设置为表创建的公共索引的特性。

对于 UNIX 服务器:

```
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
```

对于 OS/2、Windows NT 或 Windows 2000 服务器:

```
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
```

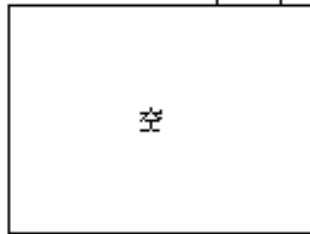
该示例设置索引类型和索引目录，然后设置索引更新频率，以便在星期一至星期五的 12:00:00 或 15:00:00 时更新索引（若至少有 100 个文本文档排队的话）。

启用文本表（可选）

1. 启用文本表

	A		B
	文本		文本

公共索引

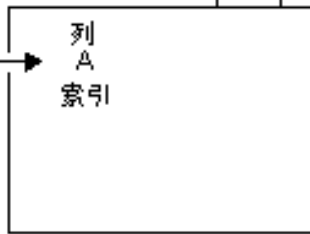


记录表

2. 启用文本列 A

	A		B
	文本		文本

公共索引



记录表

3. 启用文本列 B

	A		B
	文本		文本

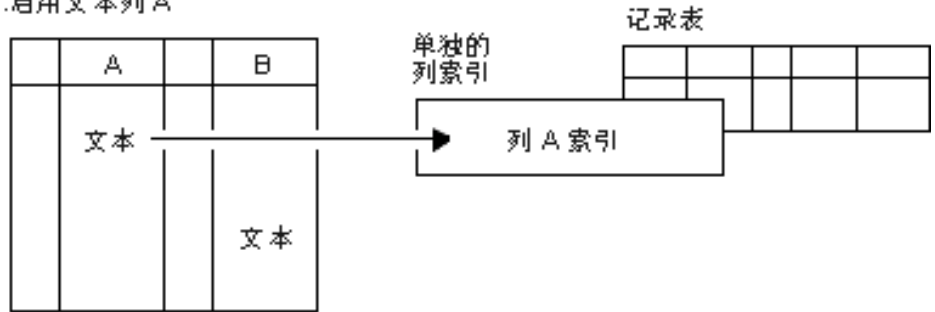
公共索引



记录表

图 9. 为表中的所有文本列创建公共索引

1. 启用文本列 A



2. 启用文本列 B

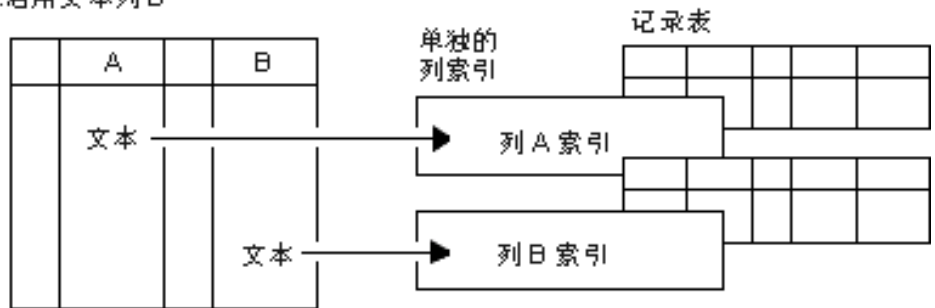


图 10. 为每个文本列创建单独的索引

启用文本列

摘要

何时

对包含要搜索的文本的每一列进行一次。

命令

`db2tx=>ENABLE TEXT COLUMN ...` (参见示例)

权限

对表的 ALTER、SELECT、UPDATE

提示

如果某个设置 (如索引类型) 对于大多数文本列都相同的话, 则使用文本配置信息来指定缺省设置。参见第18页的『文本配置设置』。

启用文本列

要取消由 `ENABLE TEXT COLUMN` 所作的更改，使用 `DISABLE TEXT COLUMN` 命令。要禁用表中所有已启用的文本列，使用 `DISABLE TEXT TABLE` 命令。

启用文本列时，会将一个句柄列添加至表中，设置文档信息（格式、语言、CCSID），并将创建日志表和索引。

添加句柄列

在此步骤中，DB2 Text Extender 将向表中添加一个 60 个字节的 `VARCHAR` 句柄列 – 该列包含与正在启用的文本列相关联的句柄 这些句柄包含关于相关文本列和相关外部文件中的文本的信息。此信息包括唯一的文档 ID、文档的语言、格式和 CCSID 以及索引名。

提示

此版本的 DB2 Text Extender 不能使用通过使用较早的发行版创建的索引。必须禁用然后重新启用所有启用了 DB2 Text Extender 的数据库和表。

DB2TX.SAMPLE

DOCID	AUTHOR	SUBJECT	DATE	COMMENT
数据	数据	数据	数据	文本

图 11. `DB2TX.MYTABLE` 表的结构 – 启用之前

包含文本块的列是 `COMMENT`。在可搜索此列中的文本之前，必须准备数据库和 `COMMENT` 列以供 DB2 Text Extender 使用。

在此准备步骤之后，`DB2TX.MYTABLE` 表就包含附加的句柄列。

DB2TX.SAMPLE

DOCID	AUTHOR	SUBJECT	DATE	COMMENT	COMMENTHANDLE
数据	数据	数据	数据	文本	文本句柄

图 12. `DB2TX.MYTABLE` 表的结构 – 启用之后

注：当随后搜索文本时，将句柄列而不是文本列指定为要搜索的列。

设置文档信息

指定在一般情况下在此文本列中存储的文本文档的类型：它们的格式（如 ASCII）、语言以及 CCSID。可在文本配置设置中指定此信息的缺省值。参见第18页的『文本配置设置』。

创建日志表

在此步骤中，创建了日志表和称作 LOGIXnnnnnn 的视图，其中 IXnnnnnn 是索引名（可从目录视图得到）。若文本配置中指定了缺省表空间，则把日志表存储在那里；否则，将其存储在 DB2 系统缺省表空间中。要优化性能和磁盘空间使用，可以指定要用于日志表的另一表空间。

还创建了触发器，每当添加或更改列中的文档，触发器就会向日志表添加信息。此信息导致在下一次索引发生时索引这些文档。

若添加或更改了外部文件，这些触发器不知道这些更改。在这样的情况下，要使触发器将信息添加至日志表，可使用 UPDATE 语句，如第96页的『更新外部文件的索引』中的示例所示。

若在创建索引期间发生错误（如，找不到排队等候索引的文档），则所谓的错误事件被添加至日志表中且可显示出来，如第103页的『显示错误事件』中所述。

提示

若在此步骤中用完了日志空间，参见第67页的『启用大型表中的文本列』以获取可能的解决方案。

在分区数据库中，每个表都被指定给表空间和节点组。将日志表指定给表空间是很重要的，而该表空间属于与启用的用户表相同的节点组。DB2 Text Extender 会在 ENABLE 命令运行期间对此进行检查。

创建索引

若想要每个文本列都有一个单独的索引，也就是说，您已跳过步骤 ENABLE TEXT TABLE，在此步骤中，DB2 Text Extender 为文本列创建单独的索引。指定索引类型、更新索引的频率以及存储索引的目录。另一方面，若希望整个表只有一个索引，则您已运行 ENABLE TEXT TABLE 并指定了索引参数；若在此处重复它们，则它们会被忽略。

启用文本列

使用 `UPDATEINDEX` 关键字来确定是立即开始指定文本列中的文本文档的索引，还是在下一次安排定期索引时才开始索引。若不使用此关键字，则采用文本配置设置中指定的值。

为文本列创建各种类型索引。 您可以为一个文本列创建多个索引。这在下列情况下很有用：通过将某文本列与不同的索引类型（如语言和 Ngram 索引）相关联，从而允许对同一文本列进行语言和模糊搜索。为此，可再次运行 `ENABLE TEXT COLUMN`，并且不仅指定要创建的其他类型的索引，而且指定唯一句柄列名。

提示

若正在使用分区数据库，并且决定更改节点组的配置并启动表行的重新分发，则必须重建文本索引和日志表，也就是说，必须禁用然后再次启用该节点组的所有启用 `DB2 Text Extender` 的表。

示例

下例启用了表 `DB2TX.MYTABLE` 中的文本列 `COMMENT`，并对创建的句柄列指定名称 `COMMENTHANDLE`：

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

文本信息的缺省值以及索引特性的缺省值都取自于文本配置设置。

下一个示例显式地设置 `COMMENT` 列中文档类型的值。索引特性的缺省值取自于文本配置设置。

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

下一个示例显式地设置为 `COMMENT` 列创建的索引的特性的值。该示例设置索引类型和索引目录，并设置索引更新频率，以便在星期一至星期五的 12:00:00 或 15:00:00 时更新索引（若至少有 100 个文本文档排队的话）。文本信息的缺省值取自于文本配置设置。

对于 UNIX 服务器：

```
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
```

对于 OS/2、Windows NT 或 Windows 2000 服务器:

```
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
```

启用大型表中的文本列

若正在使用一个行长度很长的表，则请记住，启用文本列将添加类型为 DB2TEXTH (VARCHAR 60) 的句柄列。若该表正在接近由 DB2 确定的其最大行长度，这可能很有效。

另外，当启用大型表中的文本列时，使用 DB2 UDB REORGANIZE 实用程序来检查是否需要重新组织该表。当首次启用大型表时，下列步骤会使索引进行得更快:

1. 使用 NOUPDATE 选项来启用表。这创建句柄，但是尚未对文档创建索引。
2. 使用 DB2 UDB REORGANIZE 实用程序来重组表。
3. 通过运行 UPDATE INDEX 来创建索引。

当启用文本列或外部文件时，DB2 Text Extender 将句柄列添加到该表中并初试化句柄值，从而导致 DB2 UDB 日志项被写入。若有非常大量的日志项要写入，则对于 DB2 UDB，日志空间可能会不够用。

有两种方法处理这种情况；由于性能方面的原因，第一种方法会好些:

- 通过使用 DB2 UDB UPDATE DB CFG 命令修改 LOGPRIMARY、LOGSECOND 和 LOGFILSIZ 的数据库配置参数以增加可用的日志空间。下列值是凭经验而定的值；可能需要更改它们以适合您的安装。

```
LOGSECOND 50
```

确保 LOGPRIMARY 和 LOGSECOND 之和不大于 128。还应该增加应用程序堆栈的大小。

```
APPLHEAPSZ 512
```

- 通过使用第16页的『配置』中所描述的 COMMITCOUNT 配置参数来强制 DB2 UDB 执行中间 COMMIT。您所指定的值指示在多少个插入或更新语句之后，

启用文本列

DB2 Text Extender 将发出 DB2 UDB 落实语句这就减少了日志表所需的大小，尽管它同时增加了启用步骤所需的时间。

启用不支持的数据类型的文本列

文本列必须是要由 DB2 Text Extender 启用的 CHAR、GRAPHIC、VARGRAPHIC、LONG VARGRAPHIC、DBCLOB、VARCHAR、LONG VARCHAR 或 CLOB。如果这些文档位于另一类型的列中（如用户定义单值类型 (UDT)），则必须提供这样一个函数，该函数将用户类型作为输入，且使输出类型为 CHAR、GRAPHIC、VARGRAPHIC、LONG VARGRAPHIC、DBCLOB、VARCHAR、LONG VARCHAR 或 CLOB。

使用 ENABLE TEXT COLUMN 中的 FUNCTION 关键字来指定此函数的名称。

示例：您打算在表中存储压缩文本。

1. 为该文本创建用户定义单值类型 (UDT):

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

2. 创建一个表并将文本插入该表中:

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

要启用文本列以供 DB2 Text Extender 使用:

1. 创建一个用户定义函数 (UDF)，例如 UNCOMPRESS，该函数接收类型为 COMPRESSED_TEXT 的值，而返回对应的解压缩文本，例如，CLOB(10M) 值。
2. 使用 FUNCTION 关键字来启用文本列以标识 UNCOMPRESS UDF:

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

启用包含数据类型 DATALINK 的列

DB2 Text Extender 支持数据类型 DATALINK。DB2 的 DataLink 技术为存储在 DB2 中的数据和存储在外部文件中的数据提供了参考完整性。它是通过提供一种新的 DATALINK 数据类型做到这一点的，该数据类型保存了对存储在数据库外部的对象的参考。存储在 DB2 表中的 DATALINK 值按照“统一资源定位器”(URL) 来对包含文件和文件名的 DataLink 服务器的名称进行编码。

如果 DATALINK 数据是在外部文件而不是在 DB2 表中，则遵循『启用外部文本文件』中的说明。

在启用包含 DATALINK 数据类型的文本列之前，很重要的一点就是要运行安装验证样本（参见第16页的『准备样本数据库以进行安装验证』）。

AIX 示例： 以下语句显示在 AIX 中如何使用数据类型 DATALINK 将 DataLink 值插入到文本列中。

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

在本示例中，*dlfs.boeblingen.ibm.com* 是文件服务器名，*test* 是文件系统名，而 *documents/letter1* 是路径和文件名。

确保文件系统是以与 URL 中所指定的完全相同的名称安装在 DB2 服务器上的。

Windows NT 或 Windows 2000 示例： 以下语句显示在 Windows 中如何使用数据类型 DATALINK 将 DataLink 值插入到文本列中。

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

在本示例中，*cdrive* 是存储有该文件的驱动器的共享名。

如果在对存储在 DataLink 列中的文档创建索引时遇到问题，则应确保可使用由 DB2 提供的 API 存取 DB2 服务器中的文档。

启用外部文本文件

此选项仅用来维护与先前发行版的兼容性。如果是对未存储在 DB2 表中的文件创建索引，首选方法是使用具有 DATALINK 数据类型的列。参见第68页的『启用包含数据类型 DATALINK 的列』。

摘要

何时	可选。 对与要搜索的外部文件相关联的每个表进行一次
命令	db2tx=>ENABLE TEXT FILES ... （参见示例）
权限	对表的 ALTER、SELECT、UPDATE

启用外部文本文件

DB2 Text Extender 不仅可以在存储在 DB2 UDB 表中的文本中进行搜索，也可以在存储在文件中的文本文档中进行搜索。若想要搜索外部文件中的文本，此准备步骤是必需的。与外部文本文件相关联的表不可由命令 `ENABLE TEXT TABLE` 启用过。

创建索引、创建日志表以及设置文档信息的方式与第63页的『启用文本列』中所描述的方式相同。

提示

1. 若在此步骤中用完了日志空间，参见第67页的『启用大型表中的文本列』以获取可能的解决方案。
2. 若正在启用的表是分区的，并且使用具有多个物理节点的节点组，则应确保在表列中正在引用的外部文件位于它们各自的表分区所驻留的节点上。

类型 `DB2TEXTFH` 的句柄列被添加至现存的 `DB2 UDB` 表中。句柄列将存放对外部文件的引用，每个句柄都包含索引和文档信息（`CCSID`、格式和语言）。

有关说明，参见第78页的『外部文件句柄』。

可用与启用文本列相同的方法指定附加参数（如缺省索引特性）。

创建索引之后，可以移动或删除外部文件。仍可对文件进行搜索。可在表中插入新行并使用 `UPDATE INDEX` 来以新的文件引用更新索引。

若启用的表正在使用具有多个物理节点的节点组，则应确保正在表列中引用的外部文件位于表分区驻留的节点上。

示例

1. 创建至少具有一列的表 `DB2TX.EXTFILE`，或者使用现存的表。
2. 将句柄列 `FILEHANDLES` 添加至表 `DB2TX.EXTFILE` 中

```
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. 初始化句柄

- 对于新表中的每一行:

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

- 对于现存表，更新句柄列以反映外部文件引用，并指定外部文件的名称：

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

提示

不要将 INIT_TEXT_HANDLE 用于更新引用外部文件的句柄列。

4. 更新索引

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

结束对话

现在已经完成了准备要搜索的文本文档的各个步骤。

若启用文本列时对 UPDATEINDEX 关键字指定了 NOUPDATE，则 DB2 Text Extender 不立即对文本创建索引，而是等到下一次定期索引时才进行。要现在更新索引，参见第95页的『更新索引』。

当完成了对文档的索引时，可开始检索信息，如第73页的『第6章 如何进行搜索』中所述。

提示

使用 GET INDEX STATUS 来确定何时完成了索引。

要终止 DB2 Text Extender 命令处理器，输入：

```
db2tx=>quit
```

结束对话

第6章 如何进行搜索

DB2 Text Extender 提供了一些 SQL 函数，使您能够在 SQL 查询中包括文本搜索子查询。除那些在 SQL 中常用的函数之外，还提供了这些函数，此处提到它们时时作为 DB2 Text Extender 函数的。

参考第185页的『第11章 搜索函数』以获取这些函数的语法的说明。

在查找之前，阅读第40页的『搜索类型』，并同时使用 GET INDEX SETTINGS，以了解与您要其中进行查找的文本相关的索引类型。根据索引类型的不同，搜索可产生不同的结果。

在本章的示例中，假定索引类型是语言索引。

本章描述:

- 样本 DB2 Text Extender 函数
- 样本表 DB2TX.SAMPLE
- 外部文件句柄
- 设置函数路径以向 SQL 提供对 DB2 Text Extender 函数的存取
- 使用 CONTAINS、NO_OF_MATCHES 和 RANK 来对文本进行搜索
- 使用 CONTAINS 示例在 DB2 Text Extender 函数 中指定搜索变元
- 使用 CONTAINS 和 REFINE 细化先前搜索
- 使用 INIT_TEXT_HANDLE、CCSID、FORMAT 和 LANGUAGE 在句柄中设置并抽取信息
- 使用 SEARCH_RESULT 来改进搜索性能。

何处可找到搜索函数的语法示例

DB2 Text Extender 提供了命令行处理器输入文件 txsample.udf。它包含对样本表运行的 DB2 Text Extender 搜索函数的示例。使用此文件查看 DB2 Text Extender 文本准备和搜索 UDF 的语法示例，以及在搜索变元中使用的语法示例。

要运行此文件，输入:

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

样本表 DB2TX.SAMPLE

样本表 DB2TX.SAMPLE

许多搜索语法示例中都用到样本表 DB2TX.SAMPLE。您可以使用 DB2 Text Extender 来自己试用这些样本

要创建表 DB2TX.SAMPLE, 参见第16页的『准备样本数据库以进行安装验证』。

在表6中显示了从 DB2TX.SAMPLE 表中抽取的一个样本。

表 6. 从示例表 DB2TX.SAMPLE 中抽取的一个样本

DOCID	AUTHOR	SUBJECT	DATE	COMMENT
doc 5	RSSHERM at CHGVMIC1	LIBDB2E.A error	1995-07-25 -20.13.59	Customer is getting a 'No such file or directory' on LIBDB2E.A. It does not appear to be the same error message that relates to the asynchronous I/O driver. He is using beta 4 on 3.2.5. I have had him compare the permissions and ownership of /usr/lpp/db2_02_01/lib files with mine, and they are now the same. His .profile and ENV also look good. He has, unfortunately, COMMITTED the install. What else could be wrong.
doc 6	EDWARDSC at SYDVM1	Lowercase Userid and Password from DDCS/2	1995-07-25 -20.15.20	After rechecking, the instance where I had problems with case-sensitivity was using a DB2/2 gateway to MVS. It didn't like it when I passed a lower case userid (didn't care about passwd). Connection was only successful if I actually typed an upper case userid. So, I guess this doesn't help your situation. Sorry.

表 6. 从示例表 DB2TX.SAMPLE 中抽取的一个样本 (续)

DOCID	AUTHOR	SUBJECT	DATE	COMMENT
doc 7	SKY at TOROLAB4	ODBC & Stored Procedures	1995-07-25 -20.42.27	<p>There are two sets of sample programs explaining the use of Stored Procedures using CLI (ODBC).</p> <p>The C file inpsrv2.c (placed on the server), and the C file inpli2.c (placed on the client) make up the sample that demonstrates using stored procedures for input. The files outsrv2.c and outcli2.c make up the sample that demonstrates using stored procedures for output.</p> <p>These files are part of the .../sqllib/samples/cli files. The MAKE file will automatically build them and transfer the server file to the correct subdirectory.</p>
doc 8	ADAMACHE at TOROLAB2	DB2SYS.DLL access violation	1995-07-25 -21.13.22	<p>Did you have a previous beta version installed? If so, did you remove it using Software Installer?</p> <p>Did you remove the database directories (SQLDBDIR and SQL00001, etc.) from previous beta drivers?</p>

样本表 DB2TX.SAMPLE

表 6. 从示例表 DB2TX.SAMPLE 中抽取的一个样本 (续)

DOCID	AUTHOR	SUBJECT	DATE	COMMENT
doc 9	ADAMACHE at TOROLAB2	CREATE DB = SYS3175: db2sysc.exe in db2eng.dll	1995-07-25 -21.40.09	<p>Many DB2/2 beta users delete a previous beta with Software Installer, install beta 5 (or golden code now), create a database, and get: SYS3175: db2sysc.exe in db2eng.dll</p> <p>This happens because the directory format has changed between beta4 and beta5. Our DB2/2 installation does not migrate the sqlbdir directory between beta drivers. You should remove all occurrences of sqlbdir and sql000x directories and \sqllib\db2\sqlbdir directory.</p> <p>What you should do is delete the previous beta with Software Installer, remove all occurrences of sqlbdir and sql000x directories and \sqllib\db2\sqlbdir directory, and then install the new code.</p>
doc 10	RSSHERM at CHGVMIC1	DB2/NT - SNA support	1995-07-25 -22.10.15	Will DB2/NT be able to act as both a server to CAE/WIN clients and also as a client (hopping) to DB2/6000 and/or DB2/MVS over an SNA network? The other alternative would be DRDA from DB2/NT to DB2/6000 and/or DB2/MVS - again via SNA, which I assume is supported?

以下是显示第一列和最后一列的表结构的一部分:

DB2TX.SAMPLE

DOCID	COMMENT
doc 1	Customer is ...
doc 2	After rechecking ...

图 13. DB2TX.SAMPLE 表结构

包含要搜索的文本的列是 COMMENT。在可以通过此列中的文本进行搜索之前，您必须通过 DB2 Text Extender 使用 ENABLE TEXT COLUMN 命令准备 COMMENT 列以供使用

在准备步骤之后，DB2TX.SAMPLE 表如下所示：

DB2TX.SAMPLE

DOCID	COMMENT	COMMENTHANDLE
doc 1	Customer is ...	X'..handle..'
doc 2	After rechecking ...	X'..handle..'

图 14. 启用之后的 DB2TX.SAMPLE 表

现在该表有句柄附加列，并且每个文本对象都有代表它的独特句柄。

稍后在已启用的文本列中插入文本时，插入触发器将为它创建一个句柄。

DB2TX.SAMPLE

DOCID	COMMENT	COMMENTHANDLE
doc 1	Customer is ...	X'..handle..'
doc 2	After rechecking ...	X'..handle..'

由 ENABLE TEXT COLUMN 创建的句柄

Inserted row:

doc 11	I have installed ...	X'..handle..'
--------	----------------------	---------------

由一个插入触发器创建的句柄

图 15. 由触发器创建已插入行的句柄

句柄包含下列信息：

- 文档 ID
- 相关索引的名称和位置
- 文档信息：CCSID、格式和语言。

由 DB2 Text Extender 提供的 SQL 函数将句柄看作参数，并且将文本作为该表的 SQL 处理的一部分进行存储、访问、搜索和操作。

外部文件句柄

外部文件句柄

DB2 Text Extender 不仅可以在存储在 DB2 UDB 表中的文本中执行搜索，还可以在存储在其他位置的文本文件中执行搜索。第69页的『启用外部文本文件』描述准备步骤，该准备步骤使在未存储于 DB2 UDB 表中的文本文档中执行搜索成为可能。在此步骤中，ENABLE TEXT FILES 命令为外部文件句柄创建类型为 DB2TEXTFH 的句柄列。该句柄列被添加到当前表中。

例如，可以创建一个表，该表包含作者姓名和文档创建日期列。

使用 INIT_TEXT_HANDLE 初始化文件句柄。每个句柄不仅包含文档 ID、相关索引的名称和位置和文档信息（CCSID、格式和语言），还包含对外部文件的引用。

设置当前函数路径

►► SET CURRENT FUNCTION PATH [=] DB2TX, ... ◀◀

使用 SQL 语句 SET CURRENT FUNCTION PATH 来将 DB2TX 添加至当前路径名，以使 SQL 能够查找 DB2 Text Extender 函数。若决定不执行此操作，则可以通过对 CONTAINS 函数输入 DB2TX.CONTAINS 来直接限定函数名。

本章中的示例对 DB2 Text Extender 函数使用限定格式您可以按这些语句的原样使用这些示例语句，而无需设置当前函数路径。

提示

记住在每次连接至数据库时要设置当前函数路径。

搜索文本

►► CONTAINS (—handle—, —search-argument—) ◀◀
 | NO_OF_MATCHES |
 | RANK |

►► SEARCH_RESULT (—schema—, —table—, —handle—, —search-argument—) ◀◀

本节描述如何使用随 DB2 Text Extender 提供的 SQL 函数在包含文本的 DB2 数据库中进行搜索。它告诉您如何：

- 执行查询
- 确定在一个文本文档中找到了多少匹配
- 获得所找到的文本文档的等级。

第92页的『提高搜索性能』中描述了 SEARCH_RESULT 的用法。

这些 DB2 Text Extender 函数中的每一个函数会在文本索引中搜索出现的搜索变元。例如，如果表中有 100 000 个文本文档，则会调用 CONTAINS、RANK 或 NO_OF_MATCHES 函数 100 000 次。但却不搜索文本索引 100 000 次。而是在第一次调用该函数时，创建了一个所有包含搜索词条的文档的内部列表；后续的函数调用会确定有关的文档是否在该列表中。

提示

使用 DB2 Text Extender 函数在表中进行搜索时，一定要将句柄列（而不是文本列）传送至该函数。若尝试在文本列中搜索，则 SQL 用一条信息作为响应，指示数据类型错误，例如：

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

若在发出 ENABLE TEXT TABLE 或 ENABLE TEXT COLUMN 命令之后立即搜索文本，则将发生 RC_SE_EMPTY_INDEX 错误，指示由该命令创建的索引尚未存在。创建索引所花费的时间取决于某些因素，如已索引的文档数、以及执行该索引的系统的性能。所需时间可以从几分钟到几小时，并应在系统负荷较轻时执行创建，如在晚上。

若此信息出现，尝试稍后再次搜索，或使用 GET INDEX STATUS 以检查是否发生了索引错误。

执行查询

本示例示范 CONTAINS 函数如何在以句柄标识的文档中搜索文本。若文本符合搜索变元，则返回 1，否则返回 0。

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

本示例中，在 COMMENTHANDLE 列中的句柄所引用的文本中搜索词条 compress。在 COMMENTHANDLE 列中的句柄指示对 COMMENT 文本创建索引的位置。

提示

若已为表或列创建了混合大小写的标识符，则记住必须将这些名称括在双引号中。例如：

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

若从命令行指定 DB2 UDB select 语句，则操作系统命令行语法分析器将从该命令字符串中删除特殊字符（如双引号），所以您必须使用一个反斜杠来屏蔽这些特殊符号。例如：

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

搜索并返回所找到的匹配数

使用 NO_OF_MATCHES 函数确定在每个文本文档中找到搜索条件的频率。

```
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 返回一个整数值。

搜索并返回所找到的文本文档的等级

RANK 是一个绝对值，它指示某个文档相对于其他找到的文档来说，满足搜索条件的程度。该值指示在文档中找到的、与文档的大小相关的匹配数。可通过使用 RANK 函数来获取所找到的文档的等级。

下面是一个示例：

```
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 返回一个在 0 和 1 之间的“双精度”值。

指定搜索变元

将在 CONTAINS、NO_OF_MATCHES、RANK 和 HANDLE_LIST 中使用搜索变元。本节使用 CONTAINS 函数来显示 DB2 Text Extender 函数中搜索变元的一些不同示例。

搜索若干词条

在一个搜索变元中可以有多个词条。组合几个搜索词条的一个方法是用逗号将它们组合在一起，如下所示：

```

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

```

搜索变元的这种格式查找包含任何搜索词条的文本。在逻辑项中，用 OR 运算符来连接这些搜索词条。

用布尔运算符 AND 和 OR 来进行搜索

(另见第86页的『用布尔运算符 NOT 进行搜索』。)

使用布尔运算符 “&” (AND) 和 “!” (OR) 可以将一些搜索词条与其他搜索词条组合在一起。例如：

```

db2=>SELECT DATE, SUBJECT
      FROM DB2TX.SAMPLE
      WHERE DB2TX.CONTAINS (COMMENTHANDLE,
        '"compress" 0| "compiler"') = 1

```

可使用布尔运算符来组合若干词条：

```

db2=>SELECT DATE, SUBJECT
      FROM DB2TX.SAMPLE
      WHERE DB2TX.CONTAINS (COMMENTHANDLE,
        '"compress" 0| "compiler" & "DB2"') = 1

```

若使用多个布尔运算符，DB2 Text Extender 会从左到右计算它们，但 AND 逻辑运算符 (&) 优先于 OR 逻辑运算符 (!)。例如，若不包括圆括号，

指定搜索变元

```
"DB2" & "compiler" 0| "support" & "compress"
```

计算为:

```
("DB2" & "compiler") 0| ("support" & "compress")
```

因此在以下示例中必须包括圆括号:

```
"DB2" & ("compiler" 0| "support") & "compress"
```

若将布尔运算符与使用逗号分隔符连接的搜索词条组合在一起, 如下所示:

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

则将逗号看作是 OR 布尔运算符, 如下所示:

```
("compress" 0| "compiler") & "DB2"
```

搜索词条变体

若使用**精确索引**, 则 DB2 Text Extender 完全按您所输入的那样搜索这些词条。例如, 词条 `media` 仅查找包含“`media`”的文本。找不到包含有单个“`medium`”的文本。

若使用**语言索引**, 则 DB2 Text Extender 还查找词条变体, 如名词复数或动词的不同时态。

例如, 词条 `drive` 查找包含“`drive`”、“`drives`”、“`driving`”、“`drove`”和“`driven`”的文本。

搜索词条的各部分 (字符屏蔽)

屏蔽字符, 也称为“通配符”, 提供了一个使搜索更加灵活的方法。它们代表在搜索词条的前面、中间或后面的可选的字符。它们增加了搜索所找到的文本文档数。

提示

若使用屏蔽字符, 则不能使用 `SYNONYM FORM OF` 关键字。

若使用精确索引, 则屏蔽字符对查找词条变体特别有用。若使用语言索引, 则总能找到通过使用屏蔽字符找到的许多变体。

注意，不能将字分段（由通配符屏蔽的字）缩减至基本格式。因此，若要查找 `passe%`，则将不能查找字 “`passes`” 或 “`passed`”，因为在该索引中已将它们缩减至其基本格式 “`pass`”。要查找它们，必须搜索 `pass%`。

DB2 Text Extender 使用两个屏蔽字符：百分号 (%) 和下划线 (_):

- % 代表任意数目的任何字符。以下是一个在搜索词条的前面使用 % 作为屏蔽字符的示例:

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

此搜索词条查找包含诸如 “`username`”、“`file_name`” 和 “`table-name`” 的文本文档。

% 还可以代表整个字: 以下示例查找包含诸如 “`graphic function`” 和 “`query function`” 短语的文本文档。

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

- _ 代表在搜索词条中的一个字符: 以下示例查找包含 “`CLOB`” 和 “`BLOB`” 的文本文档。

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

查找已包含屏蔽字符的词条

若想要搜索包含 “%” 字符或 “_” 字符的词条，则必须将所谓的转义字符放在该字符的前面，然后用 `ESCAPE` 关键字标识该转义字符。

例如，要搜索 “`10% interest`”:

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

此示例中的转义字符是 “!”。

搜索任何顺序的词条

如果按以下示例中所示的方式搜索 “`hard disk`”，则仅当两个词条相邻并以所显示的顺序出现时，才能找到它们，而与您使用的索引类型无关。

指定搜索变元

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

要搜索任何顺序的词条，例如顺序“data disks and hard drives”，则使用逗号来分隔这些词条：

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

在同一个句子或段落中搜索词条

下面是一个搜索变元的示例，该搜索变元查找搜索词条出现在同一个句子中的文本文档：

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

还可以搜索一起出现的两个以上的字。在下一个示例中，对在同一段落中出现的若干个字进行搜索：

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

在结构化文档的节中搜索词条

下面是一个搜索变元的示例，该搜索变元查找这样的文本文档，其中的搜索词条 Williams 出现在结构化文档的 play 节的 author 子节中：文档结构由文档模型文件中描述的模型 play 指定。有关详情，参见第47页的『使用结构文档（部分支持）』。

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

搜索词条的同义词

对于语言索引，通过不仅查找指定的搜索词条，而且还查找具有类似意义的字，可以使搜索更加灵活。例如，在搜索“book”的同时还搜索其同义词，可能会有所帮助。要执行此操作，指定：

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

当使用 SYNONYM FORM OF 时，假定用 OR 逻辑运算符将词条的同义词连接在一起，也就是说，变元为：

```
"book" 0| "article" 0| "volume" 0| "manual"
```

同义词在随 DB2 Text Extender 提供的词典中 用于同义词的缺省词典总是使用 US_ENGLISH（美国英语），而不是在文本配置设置中所指定的语言。

可以通过指定其他语言来更改用于特定查询的词典。下面是一个示例：

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

提示

若在搜索词条中有屏蔽字符，或若搜索变元使用了 NOT，则不能使用 SYNONYM 关键字。

执行语言搜索

DB2 Text Extender 提供强大的语言处理功能，基于您提供的搜索词条执行搜索 这些语言功能在索引为语言时适用。这些语言功能在第235页的『第15章 对语言索引和精确索引的语言处理』中作了描述。

有关它的一个示例是搜索复数形式（如“utilities”）和查找“utility”。在开始搜索之前，使用英文词典将该复数形式缩减至其基本形式 utility。

然而，该英文词典中没有关于将其他语言的词条变体缩减至其基本形式的信息。要搜索其他语言的词条的复数，必须使用该种语言的词典。

例如，若指定 GERMAN，则可以搜索“geflogen” (flown)，并查找其基本形式“fliegen” (fly)— 的所有变体，不仅有“geflogen”，还有“fliege”、“fliegt”等等。

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

提示

当在不是“美国英语”的文档中搜索时，在搜索变元中指定语言，而不考虑缺省语言。

若总是指定搜索词条的基本形式，而不是其变体，则不需要指定语言。

要知道为什么，则考虑在对数据库中的文本创建索引时所发生的情况。若使用语言索引，则在这些词条存储在索引中之前，该词条的所有变体都将被缩减至其基本形式。这意味着，在 DB2TX.SAMPLE 表中，尽管词条“decompress”出现在 COMMENT 列的第一项中，“decompression”出现在第二项中，但其索引却只包含基本形式“decompress”，并且将此词条（或它的变体）标识为在两项中都存在。

稍后，若搜索基本形式“decompress”，则您将找到所有变体。然而，若搜索变体（如“decompression”），则不能直接查找。必须为该搜索指定适当的词典，以便能够首先将变体转化为其基本形式。

用布尔运算符 NOT 进行搜索

可以使用布尔运算符 NOT 从搜索中排除特定的文本文档。例如：

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

从搜索“compress”或“compiler”中排除了包含词条“DB2”的任何文本文档。

不能将 NOT 运算符与在第84页的『在同一个句子或段落中搜索词条』中所述的 IN SAME SENTENCE AS 或 IN SAME PARAGRAPH AS 组合在一起使用，也不能与第84页的『搜索词条的同义词』中所述的 SYNONYM FORM OF 一起使用。

只能与主搜索一起使用 NOT 运算符，也就是说，不能自由组合 &、() 和 NOT 运算符（参见第202页的『搜索变元语法』）

不允许使用 NOT 的示例：

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

允许的示例是：

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

模糊搜索

“模糊”搜索对拼写方式类似于搜索词条的字进行搜索。它适用于 Ngram 索引。

例如:

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

此类搜索可以找到出现拼写错误的字 `conpress`。

在示例“2”中，匹配级指定精确度等级。支持五个级别，其中级别 1 提供大约百分之二十的最松散匹配，级别 5 提供大约百分之九十的最紧密匹配。当在文档中可能存在错误拼写时使用模糊搜索，通常，当使用“光盘字符识别”设备创建文档或语音输入时使用它。

考虑字与短语的边界

“边界”搜索是为韩文开发的。它确保 DB2 Text Extender 在搜索期间考虑到字的边界。例如:

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

搜索发音类似的字

“声音”搜索查找发音类似于搜索变元的字。当文档包含听起来相近，但拼写不同的字时，这很有用。例如，发音为 `my-er` 的德语名有若干种拼写方法。

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

此搜索能够找到各处出现的“Meyer”、“Mayer”和“Maier”。

同义词典搜索

同义词典搜索是 DB2 Text Extender 的另一个功能强大的搜索词条扩展功能。所搜索的附加词条是取自于您自己构建的同义词典中，因此您可以直接控制它们。例如，搜索“database”，您同时可以找到如“repository”和“DB2”等词条。

指定搜索变元

此类型搜索用于某些特定区域上，该区域是要在其中进行频繁搜索的区域，也就是指为了产生更加显著有效地结果，值得花时间来构建一个同义词典的区域。

有关详情及如何构建同义词典的说明，参见第246页的『同义词概念』。第246页的图18中的示例将从数据库主题上的同义词典中抽取一小部分。它用于下列示例中，这些示例示范使用同义词典扩展的语法。

此示例采用词条“object relational database management system”，并扩展它，同时添加在“myterms”同义词典中找到的有关此词条的所有实例。此处，将“DB2”添加至该搜索。

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

下一个示例采用词条“document management system”，并扩展它，同时添加其所有同义词。有一个同义词是 - “library”。

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

自由文本和混合搜索

“自由文本搜索”是将其搜索词条表达为自由格式文本的搜索。使用要搜索的主题的自然语言描述短语或句子。在自由文本查询中，字的顺序是无关系的。此外，还支持所谓的词法相似性。在检索中，这些是某些确定的字对，它们出现在自由文本查询词条和文档集合中，使用特定的最低频率以及特定的最小间距。英文文档的间距是五个字。

注意，对于自由文本变元格式搜索字符串，不支持字符屏蔽或字屏蔽。

例如：

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

混合搜索是布尔搜索和自由文本搜索的组合。例如：

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

细化先前搜索

当搜索变元找到的匹配项太多时，应通过以“布尔 AND”关系将初始搜索变元与第二搜索变元组合在一起，而将其缩小或细化，通常这是有用的。

通过将结果存储在表中并且对此表进行下一个搜索可以细化搜索结果，而无需使用 REFINE 函数。然而，根据限定词条的个数，此方法的效率可能低于存储最后的搜索变元并使用 REFINE 的方法。

下列步骤显示如何执行搜索、并且然后使用 REFINE 函数细化它。REFINE 函数返回一个搜索变元，该变元为它的两个输入参数的“布尔 AND”组合。由 REFINE 函数返回的组合搜索变元是 LONG VARCHAR 类型的值。

1. 为旧的搜索变元创建一个表。

创建表 PREVIOUS_SEARCHES 以保留已执行的搜索的搜索变元。

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

PREVIOUS_SEARCHES

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

2. 搜索第一个搜索变元。

在样本表中搜索字“compress”。

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

将搜索变元插入到 PREVIOUS_SEARCHES 表中，以供以后的步骤使用。

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

PREVIOUS_SEARCHES

STEP	SEARCHARGUMENT
1	"compress"

3. 细化搜索。

假设该搜索返回了太多的文本文档，则通过使用 `REFINE` 函数将先前搜索词条与字 “compiler” 组合在一起来细化搜索。

```
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
```

将细化搜索变元插入到 `PREVIOUS_SEARCHES` 表中，以供以后的步骤使用。

```
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"

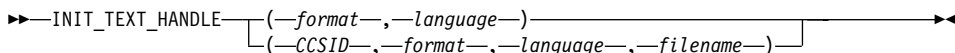
您可以重复此步骤直到所找到的文本文档数目足够少为止。

在句柄中设置并抽取信息

句柄包含其文本文档的 `CCSID`、格式和语言。外部文件句柄另外还包含指向外部文件的指针。当启用文本列或外部文件时，创建这些句柄。

此处所述的 `DB2 Text Extender` 允许您设置或更改句柄中的文本信息。

插入新文本时设置文本信息



当运行 `ENABLE TEXT COLUMN` 命令以启用已包含文本的文本列时，可以隐式地将文本的格式和语言设置为文本配置设置中所指定的值。然后将这些格式和语言设置存储在句柄中。若需要其他格式和语言值，则可在 `ENABLE TEXT COLUMN` 命令中显式地指定。

当运行 `ENABLE TEXT FILES` 命令时，还可以设置文档的 `CCSID` 和位置。

当稍后插入一个包含文本的行时，插入触发器创建一个句柄，并且将文本格式和语言设置为启用文本列时所使用的值。

要将格式和语言设置为与这些值不同的值，在 `INSERT` 命令中使用 `INIT_TEXT_HANDLE` 函数。当插入行时，`INIT_TEXT_HANDLE` 函数将创建一个部分初始化的句柄，该句柄包含指定的语言和格式值。然后，插入触发器在句柄中填充其他值。

在以下示例中，`INIT_TEXT_HANDLE` 在已初始化的句柄中预设置语言和格式。`INSERT` 命令将此句柄放在 `COMMENTHANDLE` 列中。

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

`INIT_TEXT_HANDLE` 返回的值的类型是 `DB2TEXTH` 或 `DB2TEXTFH`。

从句柄中抽取信息



下面是一个从句柄中抽取 `CCSID` 的示例：

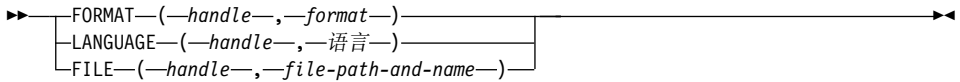
```
db2=>SELECT DISTINCT DB2TX.CCSID(COMMENTHANDLE)  
      FROM DB2TX.SAMPLE
```

用同样的方法，您可以抽取文本文档的格式或语言，或外部文件的位置。下面的示例说明 `FORMAT` 函数的用法。它返回样本表中的 `ASCII (TDS)` 文档的个数。

在句柄中设置并抽取信息

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

在句柄中更改信息



FORMAT 和 LANGUAGE 函数还可以在句柄中更改相应的说明。这些函数将已更改的句柄作为 DB2TEXTH 或 DB2TEXTFH 类型的值返回。

下列示例显示如何更改文本文档的语言设置。

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

再次使用 LANGUAGE 函数，会发现已经进行了以下更改：

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

提高搜索性能

SEARCH_RESULT 函数利用 DB2 的已赋值表函数的概念。该函数用在 SQL 语句的 FROM 子句中，并随指定搜索字符串的搜索结果返回一个中间表。搜索字符串的语法与第201页的『第12章 搜索变元的语法』中所述相同。与 CONTAINS 或 RANK 相比，此函数的优点是当涉及大型表时性能会显著提高。

返回的表的结构如下：

Column Name	Data type
HANDLE	DB2TX.DB2TEXTH, DB2TX.DB2TEXTFH
NUMBER_OF_MATCHES	INTEGER
RANK	DOUBLE

示例：

```
db2=>SET CURRENT FUNCTION PATH = db2tx
```

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

其中 T1.handle = T2.mydochandle

如果尚未设置当前路径，则示例的最后一行为：

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

`SELECT NUMBER_OF_MATCHES,RANK,HANDLE` 会使所有三项都返回, 但可以任意组合指定它们。例如, 您可能希望省略 `RANK`, 以避免进行与其相关的繁重处理。

若仅需要 `HANDLE` 值, 则可以只使用 `SELECT COUNT(*)`。

第7章 管理

本章描述如何维护文本索引、如何获取有关信息，以及如何反向文本准备过程。

维护文本索引

以下是维护任务：

- 更新索引
- 更改索引的设置
- 重设索引的状态
- 删除索引事件
- 重组索引。

可在任何时候和以任何顺序运行这些任务。

更新索引

摘要

何时 当必须立即更新索引而不必等待定期索引发生时。（参见第63页的『启用文本列』以了解有关进行周期性索引的详情。）

命令 UPDATE INDEX

权限 对表的 ALTER、SELECT、UPDATE

此示例更新公共索引表的索引：

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

此示例更新多索引表的列的索引：

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

使用此命令来立即更新索引，而不必等待下一次定期索引自动发生。当您已将几个文本文档添加至数据库中并想立即搜索它们时，该命令很有用。

维护文本索引

DB2 Text Extender 对此列（或表中的所有列）中已插入或已更改的文本文档创建索引，并从索引中将已删除的文档的各项除去 与索引相关联的日志表包含关于已插入、更新和删除的文档的信息。

更新外部文件的索引

日志表不自动包含有关对可能已索引的任何外部文件所作更改的信息（参见第69页的『启用外部文本文件』），如，用具有相同绝对路径名的较新版本来置换文档。在日志表中，DB2 Text Extender 监控不到在这样的文件上发生的更新，因为这些更新并不发生在 DB2 UDB 的范围之内

要在 DB2 Text Extender 索引中反映在外部文件上的更新，可执行下列操作：

1. 通过在相应的句柄列上发出更新语句（实际上无任何效果的）来强制将“更改”项置于日志表中：

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

其中 *filename* 是所更新的外部文件的绝对路径名。

2. 运行 UPDATE INDEX 使索引为最新，包括对外部文件所作的更改。

更改索引的设置

摘要

何时	当必须更改索引的更新频率时。
命令	CHANGE INDEX SETTINGS
权限	对表的 ALTER、SELECT、UPDATE

使用此命令来更改索引的更新频率。

更新频率

参见第45页的『更新索引』以获取更多信息。若不指定更新频率，则当前设置保持不变。

使用 UPDATEINDEX 关键字来确定是立即开始文本文档的索引，还是在下一次安排定期索引时才开始索引。若不使用此关键字，则当前设置保持不变。

示例

要更改索引的更新频率，以便在星期一至星期五的 12:00 或 15:00 更新它（若至少有 100 个文本文档排队的话）：

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

要停止索引的定期更新：

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

重设索引状态**摘要**

何时 当不能再搜索或更新索引时。

命令 RESET INDEX STATUS

权限 无

可能会发生某些情况，阻止您搜索索引或更新索引。第102页的『显示索引的状态』描述如何确定是否发生了其中一个这样的事件。RESET INDEX STATUS 重新激活索引，以便可再次使用它。

此示例重设公共索引表的索引的索引状态：

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

该语法让您重设特定文本列的索引状态。此示例重设多索引列表的索引的索引状态：

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

删除索引事件

摘要

何时 当您不再需要索引的日志表中的信息时。

命令 DELETE INDEX EVENTS

权限 无

若某事项阻止您搜索索引或更新索引，或者不能对文档创建索引，这就称为索引事件。有关索引事件的信息存储在索引的日志表中。它可帮助您确定问题的原因。当不再需要这些信息时，可以删除它们。

此示例删除公共索引表的索引的信息：

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

该语法还让您删除特定文本列的索引事件。此示例删除多索引表的索引的信息：

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

重组索引

摘要

何时 当 GET INDEX STATUS 指示应人工重组索引时。

命令 REORGANIZE INDEX

权限 无

若文本列经常更新，则在索引中进行搜索会变得效率低下。要使搜索再次变得有效率，则必须重组索引。虽然 DB2 Text Extender 知道索引何时需要重组并自动地在后台重组索引，但在以下情况下，需要使用 REORGANIZE INDEX 人工重组索引。您可以使用命令 GET INDEX STATUS 来了解索引是否需要重组。

虽然 REORGANIZE INDEX 运行时可以对索引进行搜索，但却不能执行索引更新。

此示例重组公共索引表的索引：


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

维护文本索引

此示例重组多索引表列的索引:

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

获取有用信息

本节描述用于显示有关下列各项的信息的命令:

- 数据库、表、列和文件的已启用状态
- 环境变量的设置
- 文本配置设置
- 索引状态
- 错误事件
- 索引设置
- 列的文本设置

显示已启用状态信息

摘要

何时 当您需要关于数据库、表、文本列或外部文件的已启用状态的信息时。

命令 GET STATUS

权限 无

输入:

```
db2tx GET STATUS
```

这是一个由 **GET STATUS** 显示的输出示例。它显示数据库的已启用状态，以及该数据库所包含的任何已启用的表、文本列或文本文件的已启用状态。

```
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

显示环境变量的设置

摘要

何时 当需要关于环境变量设置的信息时。

命令 GET ENVIRONMENT

权限 无

输入:

```
db2tx GET ENVIRONMENT
```

这是一个由 GET ENVIRONMENT 显示的输出示例。它显示 DB2 Text Extender 环境变量的当前设置。

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

显示文本配置设置

摘要

何时 当需要文本、索引和过程信息的缺省设置时。

命令 GET TEXT CONFIGURATION

权限 无

在第18页的『文本配置设置』中描述了这些设置。要更改它们，参见第19页的『更改文本配置』。

要显示文本配置，输入:

```
db2tx GET TEXT CFG
```

获取有用信息

这是一个由 GET TEXT CONFIGURATION 显示的输出示例。它显示当前文本配置设置。

```
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
```

显示索引的状态

摘要

何时	当需要确定是否可以搜索或更新索引时。
命令	GET INDEX STATUS
权限	无

可能会发生某些情况，阻止您搜索索引或更新索引。在这些情况下，信息被存储在索引的日志表中，可帮助确定出现问题的原因。因此，它对于检查索引的状态以及是否有任何信息可用都是很有用的。

此示例显示公共索引表的索引的索引状态：

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

该语法让您显示特定文本列的索引状态。此示例将获取多索引表列的索引的索引状态：

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

这是一个由 GET INDEX STATUS 显示的输出示例。

```
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
```

若索引分散在几个节点之间，则显示每个节点的节点信息。

搜索状态

指示是否可以使用指定的句柄列来搜索索引。若搜索不可用，则检查所指示的原因码以获取发生这种情况的原因的更多信息，然后使用 `RESET INDEX STATUS` 就能够再次使用索引。参见第289页的『第20章 错误事件原因码』。

更新状态

指示是否可以更新所指定的表或列的索引。若索引更新功能不可用，则检查所指示的原因码以获取关于发生这种情况的原因的更多信息，然后使用 `RESET INDEX STATUS` 就能够再次使用索引。

重组状态

指示是否可以重组所指定的表或列的索引。若重组功能不可用，则检查所指示的原因码以了解发生这种情况的原因的详情。重组不可用的常见原因是当前正在更新索引。

安排的文档

显示在队列中所列示的要创建索引（或者要从索引中删除的）的文档的数目。

索引的文档

显示已从安排的文档队列索引的文档的数目。

主索引文档

显示主索引中的文档数。

辅助索引文档

显示辅助索引中的文档数。

错误事件

显示在索引的日志表中可得到的事件的数目。可显示此信息，如『显示错误事件』中所述。当不再需要此信息时，可以删除它，如第98页的『删除索引事件』中所述。

显示错误事件

当索引期间发生问题时，如找不到计划要索引的文档，这些所谓的错误事件被写至索引的日志表中。

第289页的『第20章 错误事件原因码』中描述了事件返回码。

可以访问在称为 `db2tx.LOGIXnnnnnn` 的日志表的视图中的错误事件，其中 `IXnnnnnn` 是索引名，可从目录视图得到。

获取有用信息

要获取索引名:

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

错误事件视图的布局如下:

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

这是一个显示如何从索引记录访问信息的示例:

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

显示索引设置

摘要

何时	当需要关于索引的设置的信息时。
命令	GET INDEX SETTINGS
权限	无

此示例获取公共索引表的索引的索引设置:

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

此示例获取多索引表列的索引的索引设置:

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

若该表作为多索引表被启用, 则此命令显示表中的所有启用的文本列的索引设置。

对于公共索引表, 这是一个由 GET INDEX SETTINGS 显示的输出示例。公共索引表的输出对每个索引显示类似的信息。该语法让您请求特定文本列的索引设置。

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

```

若索引分散在几个节点中，则显示索引目录的节点信息。

显示列的文本设置

摘要

何时 当需要关于列的文本设置的信息时。

命令 GET TEXT INFO

权限 无

此示例获取公共索引表的索引的文本信息:

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

此示例获取多索引表的索引的文本信息:

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

该语法让您指定表名和句柄列的名称。

若在命令中只指定表名，则显示此表中已启用的每一列的文本信息。若还指定句柄列名，则仅显示该列的文本信息。

这是一个此命令为多索引表显示的内容的示例:

```

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

```

使用 DB2 Text Extender 目录视图

DB2 Text Extender 为每个数据库创建并维护称为 DB2TX.TEXTINDEXES 的目录视图。它是在运行 ENABLE DATABASE 时创建的。它包含关于对 DB2 Text Extender 启用的表和列的信息。

每当启用表、列或外部文件时，就会在 DB2TX.TEXTINDEXES 中创建新项目。每当使用 CHANGE INDEX SETTINGS 命令修改索引设置时，就会更新这些项目。若禁用列或表，则这些项目被删除。

可通过正常的 SQL 查询设施获得目录视图中的数据。但是，不能使用正常的 SQL 数据处理命令来修改目录视图。不能显式地创建或卸下目录视图。表7显示目录视图的内容。

表 7. DB2 Text Extender 目录视图

列名	数据类型	可空	说明
TABLESCHEMA	CHAR(8)	否	此项目所适用的表的模式。
TABLENAME	VARCHAR(18)	否	此项目所适用的表的名称。
COLUMNNAME	VARCHAR(18)	是	此表中已启用的列的名称。若已启用表，但是尚未启用任何列，则此值为空。
HANDLENAME	VARCHAR(18)	是	句柄列的名称。若在表 TABLESCHEMA.TABLENAME 中没有启用任何列，则此值为空。
INDEXNAME	CHAR(8)	否	在启用文本表或文本列期间创建的文本索引的名称。
LOGTABLE	VARCHAR(18)	否	索引 INDEXNAME 的日志表的名称。表 DB2TX.LOGTABLE 包含有关为下一次更新文本索引而安排的文本文档的信息，以及有关错误事件的信息。
INDEXTYPE	VARCHAR(30)	否	索引类型: LINGUISTIC、PRECISE 和 NGRAM。
MINIMUM	INTEGER	是	在执行索引更新前需要的最小索引更新请求数。参见第45页的『更新索引』。如果更新频率被设置为 NONE，则此值为空。
DAYS	VARCHAR(15)	是	计划更新时间的日部分。参见第45页的『更新索引』。如果更新频率被设置为 NONE，则此值为空。
HOURS	VARCHAR(75)	是	计划索引更新时间的小时部分。参见第45页的『更新索引』。如果更新频率被设置为 NONE，则此值为空。

表 7. DB2 Text Extender 目录视图 (续)

列名	数据类型	可空	说明
MINUTES	VARCHAR(185)	是	计划更新时间的分钟部分。参见第45页的『更新索引』。如果更新频率被设置为 NONE，则此值为空。
INDEXDIRECTORY	VARCHAR(254)	否	在文件系统中存储文本索引的目录的名称。
UPDATEONCREATE	VARCHAR(10)	否	值为“update”或“noupdate”（无论用 ENABLE TEXT TABLE 或 ENABLE TEXT COLUMN 中的 UPDATEINDEX 选项或在上一次 CHANGE INDEX SETTINGS 中指定了什么值）。
COMMONINDEX	VARCHAR(4)	否	值为“yes”（若表 TABLESCHEMA.TABLENAME 是公共索引表的话）。若表 TABLESCHEMA.TABLENAME 为多索引表，则值为“no”。
CCSID	SMALLINT	是	用启用文本列命令指定的文本列 TEXTCOLUMN 的 CCSID。若 TEXTCOLUMN 为空，则此值也为空。
LANGUAGE	VARCHAR(30)	是	处理文本列 TEXTCOLUMN 时所使用的词典的名称。若 TEXTCOLUMN 为空，则此值也为空。
FORMAT	VARCHAR(30)	是	对文本列 TEXTCOLUMN 指定的格式。若 TEXTCOLUMN 为空，则此值也为空。
FUNCTIONSCHEMA	CHAR(8)	是	在 ENABLE TEXT COLUMN 命令中使用 FUNCTION 选项指定的存取函数的模式。若未指定任何 FUNCTION 选项，则此值为空。
FUNCTIONNAME	VARCHAR(18)	是	在 ENABLE TEXT COLUMN 命令中使用 FUNCTION 选项指定的存取函数的名称。若未指定任何 FUNCTION 选项，则此值为空。
PROTOTYPEHANDLE	VARCHAR(60)	是	用于在性能函数中使用的句柄。它仅包含对整个文本列是公共的索引名。
INDEXOPTION	VARCHAR(30)	是	创建索引时使用的选项：CASE_ENABLED。
INDEXPROPERTY	VARCHAR(30)	是	创建索引时使用的特性：SECTIONS_ENABLED。
NODENUMBER	INTEGER	否	表分区的节点号。

反向文本准备过程

当准备文本以供 DB2 Text Extender 使用时，进行了某些管理的更改。本节描述帮助您反向此过程的功能。

反向文本准备过程

禁用文本列

摘要

何时 当您不再想在文本列中进行文本搜索时。

命令 DISABLE TEXT COLUMN

权限 对表的 ALTER、SELECT、UPDATE

示例:

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

禁用文本列时，将发生下列情况:

- 若这是一个多索引表，也就是说，该列有其自己的文本索引和日志表，那么，这些索引、日志表和日志表触发器将被删除。
- 若这是一个公共索引表，也就是说，所有文本列共享一个索引，那么，从公共索引中除去用于此列的文档的各个词条。若这是表中唯一剩余的已启用文本列，则索引、日志表以及日志表触发器被删除。

禁用文本文件

摘要

何时 当您不再想在一组外部文本文件中进行文本搜索时。

命令 DISABLE TEXT FILES

权限 对表的 ALTER、SELECT、UPDATE

示例:

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

禁用外部文本文件时，将发生下列情况:

- 删除此句柄列的索引。
- 删除日志表和触发器。

禁用文本表

摘要

何时 当您不再想在文本表中进行文本搜索时。

命令 DISABLE TEXT TABLE

权限 对表的 ALTER、SELECT、UPDATE

示例:

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

禁用文本表时，将发生下列情况:

- 若该表的文本列具有公共索引，则此索引将被删除。但是，如果每个文本列都有个别的索引，则文本列的所有索引将被删除。
- 用来自动记录要对哪些文本文档创建索引的公共日志表被删除。但是，如果每个文本列都有个别的日志表，则所有日志表都被删除。
- 用于维护日志表的触发器被删除。
- 句柄列的内容被设置为空。

禁用数据库

摘要

何时 当您不再想在此数据库中进行文本搜索时。

命令 DISABLE DATABASE

权限 对数据库的 SYSADM 或 DBADM

要禁用连接的数据库，输入:

```
db2tx DISABLE DATABASE
```

禁用数据库时，下列对象被删除:

- 在启用数据库时创建的 DB2 Text Extender 目录视图
- DB2 Text Extender 的 SQL 函数 (UDF) 和此数据库的 DB2 Text Extender 单值类型 (UDT) 的说明
- 与此数据库的任何文本表或文本列相关的所有索引

反向文本准备过程

- 用来自动记录要对哪些文本文档创建索引的日志表，以及用来维护日志表的触发器。

因为不能删除句柄列，且句柄列为单值类型，所以不删除某些单值类型。

第8章 使用 API 函数进行搜索和浏览

本章描述如何使用 DB2 Text Extender API 的搜索和浏览函数。有关这些函数的详细说明，参考第213页的『第13章 用于搜索和浏览的 API 函数』。在第233页的『第14章 样本 API 程序』中提供了使用 API 函数的程序的示例。本章还描述了一个样本浏览函数 DesBrowseDocument。

提示

在搜索之前，应阅读第40页的『搜索类型』。根据所使用的索引类型，搜索可以产生不同的结果。使用 GET INDEX SETTINGS，找到与您在其中进行搜索的文本相关的文本索引类型。

安装应用程序

因为某些 API 函数需要数据库连接句柄作为输入，所以使用 DB2 Text Extender API 的应用程序是一个 DB2 CLI 应用程序 这意味着必须为 DB2 CLI 应用程序考虑的规则，对使用 DB2 Text Extender API 的应用程序同样适用。

在应用程序中包括 des_ext.h，它是在 DB2 Text Extender 安装目录的 include 子目录中提供的。

要在您的应用程序中使用 DB2 Text Extender API，将程序链接至该 API。

链接 UNIX 应用程序

必须将库 libdescl.a 链接至您的应用程序。此库位于 DB2 Text Extender 安装目录的 lib 子目录中。

链接 OS/2 或 Windows 应用程序

所有 DB2 Text Extender API 函数都包含在动态链接库 desclapi.dll 中 将在调入库 desclapi.lib 中解析外部 API 函数调用。它们分别在 DB2 Text Extender 安装目录的 dll 和 lib 子目录中提供。

要将应用程序与 API 动态链接库 desclapi.dll 链接，使用 desclapi.lib 作为库文件。

API 函数概述

下面是搜索和浏览函数；第一个是搜索函数，其余都是浏览函数：

DesGetSearchResultTable
DesGetBrowseInfo
DesStartBrowseSession
DesOpenDocument
DesGetMatches
DesCloseDocument
DesEndBrowseSession
DesFreeBrowseInfo。

提示

许多 API 函数需要连接句柄 (hdbc)。必须使用 SQLConnect 函数提供此句柄，但这并不妨碍您从嵌入式 SQL 程序调用 DB2 Text Extender。 *DB2 Call Level Interface Guide and Reference* 描述如何将 CLI 语句与嵌入式 SQL 语句混合。

可将 API 函数用于：

- **搜索文本**

在此方案中，只需要搜索函数 DesGetSearchResultTable。它将搜索变元和句柄列名作为输入。它搜索文本，并将有关所找到文档的信息放入您先前准备好的结果表中。

在第113页的『搜索文本』中对此函数有更详细的描述。另见第233页的『第14章 样本 API 程序』。

- **浏览文本**

按显示的顺序使用下列函数：

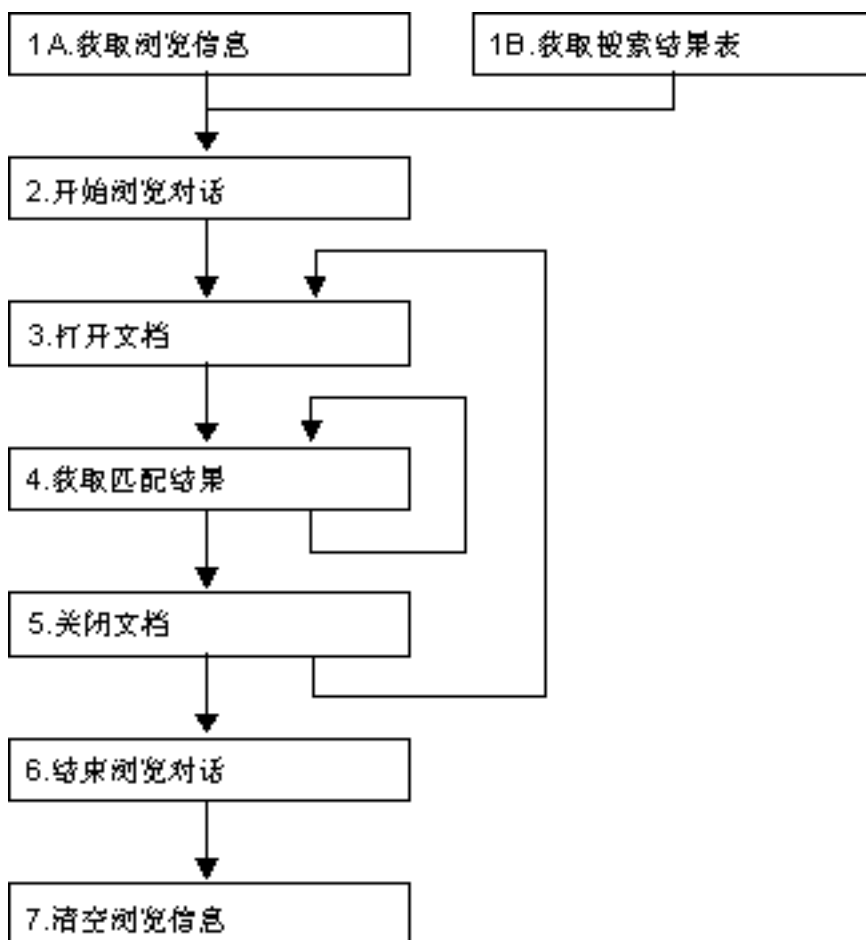


图 16. 使用 API 函数的顺序

这些函数获取突出显示的信息，然后启动一个浏览对话，以显示文本文档并突出显示找到的项。

在浏览对话中，可以使用同一个突出显示信息来打开和显示更多的文档。在第 14 页的『浏览文本』中对这些函数有更详细的描述。

搜索文本

有一个用于搜索文本的 API 函数：DesGetSearchResultTable 函数。

获取搜索结果表 (DesGetSearchResultTable)

DesGetSearchResultTable 函数接收用以在特定文本列中搜索文本文档的搜索变元，并将结果存储在表中。结果表包含每个找到的文档的句柄。根据所指定的搜索选项，还可以包含等级信息和匹配数。

还可以使用 RANK 和 NO_OF_MATCHES 函数来获取此信息。下面是一个示例：

```
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 只能在基本表上使用，但是，如果查询是纯文本查询，则它会比使用 DB2 Text Extender 函数要快；它直接到 DB2 Text Extender 服务器中获取等级和匹配数，循环次数等于找到的匹配文档数。另一方面，在 UDF 示例中，对表中的每一行调用一次 CONTAINS 函数；然后，再对每个适当的行调用 RANK 和 NO_OF_MATCHES 函数。对于每个找到的文档，执行三次独立的搜索。

输入

输入参数是：

- 数据库连接的句柄
- 要搜索的表
- 与要搜索的文本列相关的句柄列的名称
- 搜索变元
- 搜索选项
- 浏览选项（以返回浏览信息）
- 要在其中存储结果的表的名称。

输出

若指定了浏览选项，则此函数返回一个指向浏览信息的指针。

浏览文本

在第113页的图16中的这组函数找出要突出显示的项。然后它启动浏览对话，打开文档，并以数据流的形式获取匹配信息，数据流可由调用浏览器的应用程序进行语法分析。

获取浏览信息 (DesGetBrowseInfo)

DesGetBrowseInfo 函数接收搜索变元和句柄。它返回一个指向 DesStartBrowseSession 所需的浏览信息的指针。浏览信息包括所有要突出显示的项的列表。

获取浏览信息的另一个方法是在 DesGetSearchResultTable 函数中指定“浏览”选项。

输入

输入参数是:

数据库连接的句柄

句柄

搜索变元。

输出

此函数返回一个指向浏览信息的指针。

启动浏览对话 (DesStartBrowseSession)

DesStartBrowseSession 函数启动浏览对话，并创建浏览文本文档及突出显示其匹配所需的环境。它接收到一个指向浏览信息（它来自于 DesGetBrowseInfo 或 DesGetSearchResultTable）的指针，并且返回供其他浏览函数使用的浏览对话句柄。

输入

输入参数是:

指向来自 DesGetBrowseInfo 或 DesGetSearchResultTable 中的浏览信息的指针

用户 ID

口令。

输出

此函数返回一个浏览对话句柄。

打开文档 (DesOpenDocument)

DesOpenDocument 函数接收浏览对话指针、句柄和选项 DES_FAST 或 DES_EXTENDED，其中，选项指示用于突出显示所找到的项的语言处理类型。参见第244页的『步骤 2: 扩展匹配』。无需词典，DES_FAST 使用基本文本分析就可以确定要突出显示哪些项。DES_EXTENDED 使用扩展匹配。

浏览文本

`DesOpenDocument` 准备与句柄相应的文本文档，以获取该文档文本和突出显示信息，并返回用于迭代调用 `DesGetMatches` 的文档句柄。

输入

输入参数是：

来自 `DesStartBrowseSession` 的浏览对话句柄

文本句柄

匹配选项：`DES_FAST` 或 `DES_EXTENDED`。

输出

此函数返回供 `DesGetMatches` 和 `DesCloseDocument` 使用的文档句柄。

获取匹配 (`DesGetMatches`)

`DesGetMatches` 函数返回一个指针，指向文档句柄所描述的文本文档的突出显示信息。突出显示的信息是数据流。它包括文本上下文（至少一个段落）和该上下文中的突出显示文本的信息。该数据流在第220页的『数据流语法』中作了描述。应用程序可以使用用户自己的浏览器对该数据流进行语法分析和处理。

`DesGetMatches` 仅返回数据流的一部分，在输出结构中指示该部分的长度。

一个调用 `DesGetMatches` 的序列将获取整个文本文档内容。当到达文本文档的末尾时，返回一个指示符。

输入

输入参数是：

浏览对话句柄

来自 `DesOpenDocument` 的文档句柄。

输出

此函数返回一个指针，指向包含数据流部分及其长度的结构。

关闭文档 (`DesCloseDocument`)

`DesCloseDocument` 函数关闭由 `DesOpenDocument` 打开的文本文档，并在返回文档文本和突出显示信息期间释放已分配的存储器。

输入

输入参数是：

浏览对话句柄

来自 `DesOpenDocument` 的文档句柄。

输出

无。

结束浏览对话 (DesEndBrowseSession)

DesEndBrowseSession 函数结束由 DesStartBrowseSession 启动的浏览对话，并释放为该浏览对话分配的存储器。

输入

输入参数是:

一个浏览对话句柄。

输出

无。

释放浏览信息 (DesFreeBrowseInfo)

DesFreeBrowseInfo 函数释放由 DesGetBrowseInfo 分配给浏览信息的存储器。

输入

输入参数是:

指向浏览信息的指针。

输出

无。

第2部分 参考

第9章 客户机的文本准备和管理命令

命令	目的	页
CHANGE INDEX SETTINGS	更改索引的特性	123
CHANGE TEXT CONFIGURATION	更改文本配置设置	125
CONNECT	连接至数据库	128
DELETE INDEX EVENTS	从日志表中删除索引事件	129
DISABLE DATABASE	禁止 DB2 Text Extender 使用数据库	130
DISABLE TEXT COLUMN	禁止 DB2 Text Extender 使用文本列，并删除其相关的索引	131
DISABLE TEXT FILES	禁止 DB2 Text Extender 使用文本文件，并删除其相关的索引	132
DISABLE TEXT TABLE	禁止 DB2 Text Extender 使用表，并删除与该表相关的索引	133
ENABLE DATABASE	准备数据库以供 DB2 Text Extender 使用	134
ENABLE TEXT COLUMN	准备文本列供 DB2 Text Extender 使用，并为该列创建个别的文本索引	135
ENABLE TEXT FILES	准备文本文件供 DB2 Text Extender 使用，并为这些文件创建个别的文本索引	142
ENABLE TEXT TABLE	为表创建公共文本索引	145
GET ENVIRONMENT	显示环境变量的当前设置	149
GET INDEX SETTINGS	显示索引的特性	150
GET INDEX STATUS	显示索引的状态信息	152
GET STATUS	显示数据库、表和列的启用状态	153
GET TEXT CONFIGURATION	显示文本配置设置	154
GET TEXT INFO	显示文本列的文本信息	155
QUIT	从 DB2 Text Extender 命令行处理器方式退出	156
REORGANIZE INDEX	重组索引以改进搜索效率	157
RESET INDEX STATUS	重设索引的状态以允许再次使用它	158
UPDATE INDEX	更新文本索引	159

本章描述客户机的文本准备和管理命令的语法。第55页的『第5章 使文本可搜索』和第95页的『第7章 管理』描述如何使用这些命令。

在使用这些命令之前，通过输入命令 `db2tx` 来启动 DB2 Text Extender 命令行处理器。它使您进入交互式输入方式，在此方式中将所有后续命令解释为 DB2 Text Extender 命令。通常地，将在启动 DB2 的同时启动命令处理器。

要脱离此方式，输入 QUIT。

在 DB2 中，通过给命令加上 db2tx 前缀，可以从操作系统提示直接发出命令，并且可以将命令行处理器请求包括在 shell 脚本命令文件中。

提示

前缀为 db2tx 的命令将导致建立与环境变量 DB2DBDFT 中指定的缺省数据库的连接。以下命令序列不启用数据库 MYDATABASE，而是启用缺省数据库。

```
db2tx CONNECT TO MYDATABASE  
db2tx ENABLE DATABASE
```


CHANGE INDEX SETTINGS 命令

HANDLE handle-column-name

其索引更新频率将要被更改的句柄列的名称。若文本列有其自己的索引，也就是说，该索引是使用 `ENABLE TEXT COLUMN` 命令创建的，则需要此名称。

然而，若该索引是使用 `ENABLE TEXT TABLE` 命令创建的，也就是说，该表对所有文本列有一个文本索引，则忽略此关键字。

UPDATEFREQ 更新频率

索引更新频率，按何时进行更新以及必须在日志表中排队的最少文本文档数来指定。若在给定的日期和时间日志表中没有足够的文本文档，则不更新索引。

语法在第45页的『更新索引』中作了描述。

NONE 没有进行进一步的索引更新。这用于其中将没有进一步更改的文本列。

若不指定 `UPDATEFREQ` 关键字，则频率设置保持不变。

CHANGE TEXT CONFIGURATION

此命令更改启用数据库时所使用的文本配置的缺省设置。这些是文本配置设置。安装 DB2 Text Extender 时的初始文本配置设置在第18页的『文本配置设置』中作了描述

在已启用了—个特定的数据库之后，要更改该数据库的这些设置，使用第123页的『CHANGE INDEX SETTINGS』。

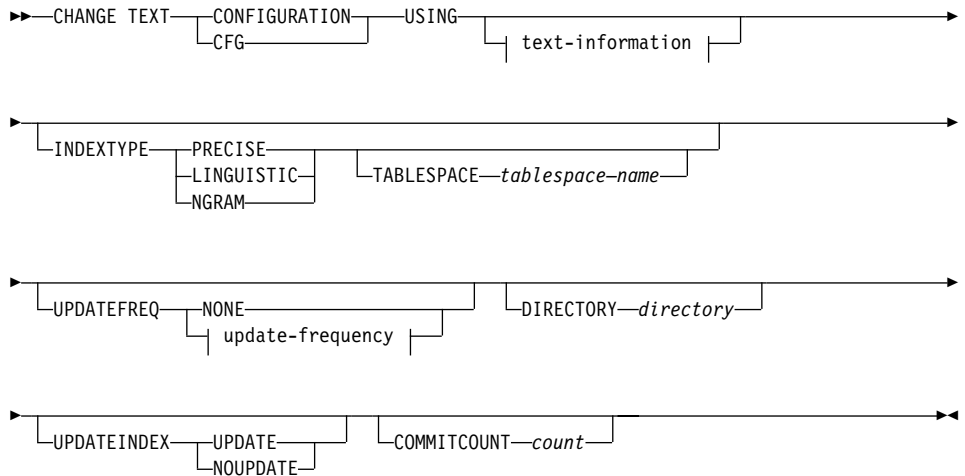
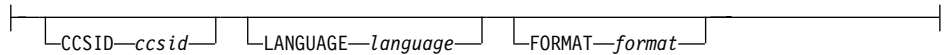
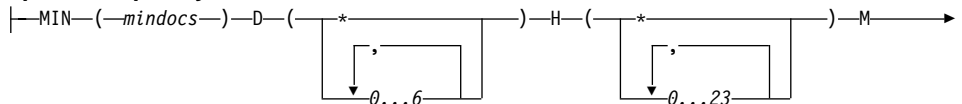
权限

您必须具有：

SYSADM 权限

DBADM 权限。

命令语法

**text-information:****update-frequency:**



命令参数

INDEXTYPE

要更改缺省索引类型，选择下列其中之一。有关更多信息，参见第40页的『搜索类型』。

PRECISE

完全按照在文本文档中出现的那样索引和搜索词条。

LINGUISTIC

在索引词条之前，将它们进行语言处理。搜索词条也在搜索开始之前进行语言处理。

NGRAM

通过对字符集进行语法分析来索引词条，而不是通过使用字典。若索引的文档包含 DBCS 字符，则此字典类型是必要的，尽管 Ngram 索引也可用于 SBCS 文档。

TABLESPACE tablespace-name

指定现存表空间的名称。表空间用来保存由 DB2 Text Extender 创建的索引特定表（如日志表）。对于大表，使用单独的表空间。若不指定表空间，则在 DB2 缺省表空间中创建那些表。

UPDATEFREQ update-frequency

索引更新频率，按何时进行更新以及必须在日志表中排队的最少文本文档数来指定。若在给定的日期和时间日志表中没有足够的文本文档，则不更新索引。

语法在第45页的『更新索引』中作了描述。

NONE 没有进行进一步的索引更新。这用于其中将没有进一步更改的文本列。

DIRECTORY 目录

将要在其中存储文本索引的目录。

UPDATEINDEX

一个关键字，它确定是否在使用此选项的命令已经完成之后立即对文本文档创建索引，而无需等待由 UPDATEFREQ 设置的下一次定期索引。这些命令是 ENABLE TEXT COLUMN 和 ENABLE TEXT FILES。

UPDATE

在已完成命令之后立即进行文本文档索引。

NOUPDATE

在由更新频率设置所设置的时间创建索引，这些更新频率设置是由 UPDATEFREQ 在此命令中指定的，或是由文本配置设置指定的。

COMMITCOUNT 计数

一个从 500 至 1 000 000 的值，该值指示在多少个插入或更新之后发出 DB2 UDB 中间落实语句。这可以避免当启用大表、列或大量外部文件时日志空间不足的情况。

CCSID ccsid

要在对文本文档创建索引时使用的“编码字符集标识符”。

有关可支持的 CCSID 的信息，参见第35页的『CCSID』。

LANGUAGE 语言

用来编写文本的那种语言。这确定当对文本文档创建索引以及当在文本文档中搜索时要使用的词典。第235页的『第15章 对语言索引和精确索引的语言处理』描述如何使用词典。

受支持的语言列示在第53页的表5中。

FORMAT 格式

存储的文本文档的类型，如 WordPerfect 或 ASCII。当对文档创建索引时，DB2 Text Extender 需要此信息。受支持的文档格式列示在第32页的『受支持的文档格式』中。

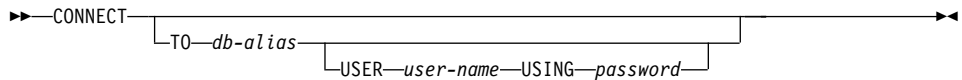
CONNECT

此命令将 DB2 Text Extender 与数据库相连。

权限

对数据库的 CONNECT 权限。

命令语法



命令参数

TO db-alias

要连接的数据库。其名称必须是数据库别名，而不是数据库名。

USER user-name

若未指定用户名，则从操作系统检索用户名。

USING 口令

仅当指定了用户名时，才需要口令。

CONNECT 无操作数

若不指定操作数并且没有相连的数据库，则 CONNECT 将隐式建立与环境变量 DB2DBDFT 中指定的数据库的连接。若不指定操作数并且有相连的数据库，则 CONNECT 显示关于当前数据库的信息。

用法注释

当在未连接数据库的情况下发出 DB2TX 命令时，DB2 Text Extender 连接至环境变量 DB2DBDFT 中指定的数据库。

要显式连接特定的数据库，发出 CONNECT TO 命令。

一次只能连接一个数据库；称之为当前数据库。在交互方式中，连接将一直持续，直至另一个 CONNECT TO 语句更改了该数据库，或发出了 QUIT 命令为止。在命令行方式中，CONNECT 命令不起作用。

若正在使用 OS/2 客户机，并且这是启动系统之后首次与数据库相连，则“用户简表管理”(UPM) 将提示您输入用户 ID 和口令。若您已注册，则 UPM 将采用此对话中的用户 ID 和口令，并且尝试连接远程服务器。参见第56页的『启动 DB2 Text Extender 命令行处理器』。

DELETE INDEX EVENTS

此命令从给定句柄列或表的索引的日志表中删除索引事件。

权限

无。

命令语法

```
▶▶—DELETE INDEX EVENTS—table-name—┬───┬───▶▶  
└───┬───▶▶  
HANDLE—handle-column-name—┬───▶▶
```

命令参数

table-name

在相连的数据库中的文本表的名称，将要从日志表中删除该数据库的错误事件。除非模式名与您的用户 ID 相同，否则该名称必须包括显式模式名（限定符）。

HANDLE handle-column-name

其信息将被从日志表中删除的句柄列的名称。

用法注释

若给出了句柄列名，则仅删除此列的索引事件。

要清除日志表，应在检查了事件发生的原因并尽可能地除去了产生错误的根源之后，删除索引事件。

DISABLE DATABASE 命令

DISABLE DATABASE

此命令重设 DB2 Text Extender 为数据库执行的任何准备工作，并且禁止 DB2 Text Extender 使用所有文本表。

权限

您必须具有：

SYSADM 权限

DBADM 权限。

命令语法

▶▶—DISABLE DATABASE—▶▶

命令参数

无。

用法注释

此命令重设连接的数据库，以使 DB2 Text Extender 无法再搜索它；也就是说，该命令禁用数据库中的所有 DB2 Text Extender 文本表和文本列。将重设数据库中为启用 DB2 Text Extender 文本表、文本列以及外部文件而做的所有修改：删除所有相关的文本索引，删除数据库中的 DB2 Text Extender 目录视图 TEXTCOLUMN，并删除所有 DB2 Text Extender 触发器。

DISABLE TEXT COLUMN

此命令禁止 DB2 Text Extender 使用文本列。

权限

对于该表，您必须至少具有下列特权的其中之一：

ALTER 特权

SELECT 特权

UPDATE 特权。

命令语法

►►—DISABLE TEXT COLUMN—*table-name*—HANDLE—*handle-column-name*—►►

命令参数

table-name

在相连的数据库中的文本表的名称，该文本表包含要禁止的列。除非模式名与您的用户 ID 相同，否则该名称必须包括显式模式名（限定符）。

HANDLE handle-column-name

要禁止 DB2 Text Extender 使用的句柄列的名称。

用法注释

删除索引。

删除用以记录句柄列中的更改（插入、更新和删除）的日志表。

删除向日志表中写入项目的触发器。

不更改句柄列。

DISABLE TEXT FILES

此命令禁止 DB2 Text Extender 使用一组外部文本文件。

权限

对于该表，您必须至少具有下列特权的其中之一：

ALTER 特权

SELECT 特权

UPDATE 特权。

命令语法

►►—DISABLE TEXT FILES—*table-name*—HANDLE—*handle-column-name*—◄◄

命令参数

table-name

在相连的数据库中的文本表的名称，该文本表包含要禁用的外部文本文件的句柄列。除非模式名与您的用户 ID 相同，否则该名称必须包括显式模式名（限定符）。

HANDLE handle-column-name

要禁止 DB2 Text Extender 使用的句柄列的名称。

用法注释

删除索引。

删除用以记录句柄列中的更改（插入、更新和删除）的日志表。还删除向日志表中写入项目的触发器。

DISABLE TEXT TABLE

此命令禁止 DB2 Text Extender 使用表中的所有文本列。

权限

对于该表，您必须至少具有下列特权的其中之一：

ALTER 特权

SELECT 特权

UPDATE 特权。

命令语法

►►—DISABLE TEXT TABLE—*table-name*—◄◄

命令参数

table-name

在相连的数据库中的文本表的名称，该文本表包含要禁用的列。除非模式名与您的用户 ID 相同，否则该名称必须包括显式模式名（限定符）。

用法注释

此命令使该表中的所有文本列不可供 DB2 Text Extender 使用。

若此表中的文本列是由 ENABLE TEXT COLUMN 个别地启用的，则该命令删除所有与这些文本列相关的文本索引。（要禁用文本列并个别地删除其相关文本索引，使用 DISABLE TEXT COLUMN 命令。）若此表中的文本列是由 ENABLE TEXT TABLE 一起启用的，则有一个对于所有文本列的公共索引。此命令删除该公共索引。

删除用以记录文本列中的更改（插入、更新和删除）的日志表。删除向日志表中写入项目的触发器。

ENABLE DATABASE 命令

ENABLE DATABASE

此命令允许当前数据库存储文本数据。

权限

您必须具有:

SYSADM 权限

DBADM 权限。

命令语法

►—ENABLE DATABASE—►

命令参数

无。

用法注释

在发出此命令之前，必须显式或隐式地连接数据库（参见第128页的『CONNECT』）。

此命令准备相连的数据库以供 DB2 Text Extender 使用 它是在数据库中启用 DB2 Text Extender 文本表或文本列之前的一个必要步骤。

ENABLE DATABASE 创建一个称为 DB2TX.TEXTINDEXES 的 DB2 Text Extender 目录视图，在第106页的『使用 DB2 Text Extender 目录视图』中有描述，还创建一个称为 DB2TX.TEXTCOLUMNS 的用于“性能”查询的目录视图。

它还创建文本配置设置，在第18页的『文本配置设置』中有描述。

还会进行某些其他工作，如 DB2 Text Extender 单值类型和 DB2 Text Extender 函数的说明。

ENABLE TEXT COLUMN

此命令使文本列能够供 DB2 Text Extender 使用。

权限

对于该表，您必须至少具有下列特权的其中之一：

- ALTER 特权
- SELECT 特权
- UPDATE 特权。

命令语法

►►—ENABLE TEXT COLUMN—*table-name*—*text-column-name*—FUNCTION—*function-name*—

►HANDLE—*handle-column-name*—text-information—

index-characteristics—UPDATEINDEX—UPDATE—NOUPDATE—

—TABLESPACE—*tablespace-name*—COMMITCOUNT—*count*—

text-information:

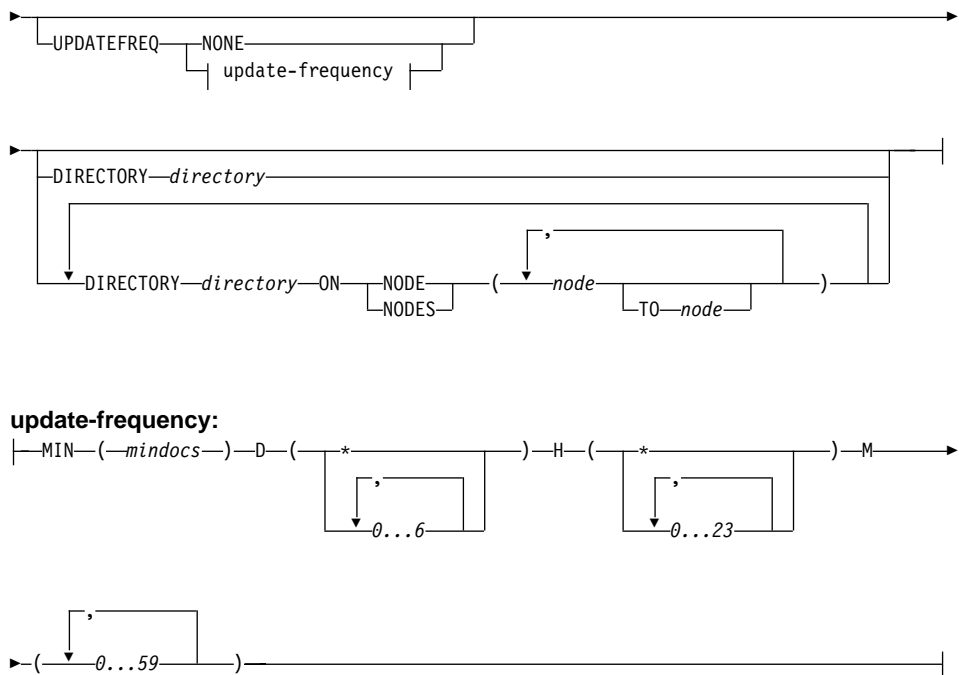
—CCSID—*ccsid*—LANGUAGE—*language*—FORMAT—*format*—

index-characteristics:

—INDEXTYPE—PRECISE—LINGUISTIC—NGRAM—INDEXOPTION—CASE_ENABLED—NORMALIZED—

—INDEXPROPERTY—SECTIONS_ENABLED—DOCUMENTMODEL—DOCUMENTMODELS—(—*model*—)

ENABLE TEXT COLUMN 命令



命令参数

table-name

在相连的数据库中，包含要启用的列的文本表的名称。除非模式名与您的用户 ID 相同，否则该名称必须包括显式模式名（限定符）。

text-column-name

要启用以供 DB2 Text Extender 使用的列的名称。此列的类型必须是 `CHAR`、`VARCHAR`、`LONG VARCHAR`、`CLOB`、`DBCLOB`、`GRAPHIC`、`VARGRAPHIC` 或 `LONG VARGRAPHIC`。若文档类型不是上述其中之一，则使用 `FUNCTION` 来转换文档类型。

FUNCTION function-name

用户定义函数的名称，DB2 Text Extender 使用此函数来存取在一个列中的文本文档，该列的类型不是 `CHAR`、`VARCHAR`、`LONG VARCHAR`、`CLOB`、`DBCLOB`、`GRAPHIC`、`VARGRAPHIC` 或 `LONG VARGRAPHIC`。有关更多信息，参见第68页的『启用不支持的数据类型的文本列』。

HANDLE handle-column-name

要添加至表以供 DB2 Text Extender 函数使用的句柄列的名称。

CCSID ccsid

要在对文本文档创建索引时使用的“编码字符集标识符”。

如果在对 Ngram 索引启用文本列时指定了 CCSID，则该 CCSID 必须与数据库的 CCSID 相同，且在搜索期间使用的 CCSID（数据库的 CCSID）必须与此 CCSID 相匹配。

要查找缺省 CCSID，使用：

```
db2tx get text cfg
```

安装缺省值为 数据库 CCSID。

若未指定此关键字，则使用在文本配置设置中所指定的 CCSID。忽略对文本配置设置的后续更改；所使用的值是启用该列时存在的值，而不是对文本文档创建索引时的值。

有关可支持的其他 CCSID 的信息，参见第35页的『CCSID』。

LANGUAGE 语言

用来编写文本的那种语言。这确定当对文本文档创建索引以及当在文本文档中搜索时要使用的词典。第235页的『第15章 对语言索引和精确索引的语言处理』描述如何使用词典。

此关键字对整个列指定语言一次。对于个别插入的文本文档来说，可使用在 INSERT 语句中的 INIT_TEXT_HANDLE 函数来覆盖此值。

若未指定此关键字，则使用在文本配置设置中所指定的语言。忽略对文本配置设置的后续更改；所使用的值是启用该列时存在的值，而不是对文本文档创建索引时的值。

受支持的语言列示在第53页的表5中

FORMAT 格式

存储的文本文档的类型，如 WordPerfect 或 ASCII。当对文档创建索引时，DB2 Text Extender 需要此信息。受支持的文档格式列示在第32页的『受支持的文档格式』中。

结构化文档支持的文档格式有：

- ASCII_SECTIONS

具有 ASCII_SECTIONS 格式的文档不包含嵌套部分。（关于嵌套部分的详情，参见第47页的『使用结构文档（部分支持）』。）一节的起始标记被下一个起始标记结束。
- HTML

ENABLE TEXT COLUMN 命令

为 HTML 文档提供了样本文档模型文件。它包含标准 HTML 定义的一个子集，您可以根据需要修改它。HTML 文档不能包含嵌套部分。

- XML

XML 文档的处理包括“文档类型定义”(DTD) 判定。对文档指定的模型将对照 DTD 进行检查。若文档框文件中定义的标记未在 DTD 中定义，就不对该文档创建索引。若未对可识别 DTD 定义模型，将不对该文档创建索引。XML 文档可包含嵌套部分。

对于这些格式，必须在文档模型文件中指定结构信息。参见第47页的『使用结构文档（部分支持）』。若指定了格式 TDS 和 INDEXPROPERTY SECTION_ENABLED，就会假设文档格式是 ASCII_SECTIONS。

模型文件中未定义的标记根据索引类型以常规方式创建索引。

此关键字对整个列指定格式一次。对于个别插入的文本文档来说，可使用在 INSERT 语句中的 INIT_TEXT_HANDLE 函数来覆盖此值。

若未指定此关键字，则使用在文本配置设置中所指定的格式。忽略对文本配置设置的后续更改；所使用的值是启用该列时存在的值，而不是对文本文档创建索引时的值。

INDEXTYPE

要创建的索引的类型。有关更多信息，参见第40页的『搜索类型』。

PRECISE

完全按照在文本文档中出现的那样索引和搜索词条。

LINGUISTIC

在索引词条之前，将它们进行语言处理。搜索词条也在搜索开始之前进行语言处理。

NGRAM

通过对字符集进行语法分析来索引词条，而不是通过使用字典。若索引的文档包含 DBCS 字符，则此索引类型是必要的，尽管 Ngram 索引也可用于 SBCS 文档。

若不指定 INDEXTYPE 关键字，则使用文本配置设置中的值。

Ngram 索引不支持 XML 格式的文档。

INDEXOPTION

创建索引时要使用的选项。

CASE_ENABLED

此选项仅可供 Ngram 索引使用。通常，Ngram 索引不允许区分大小写搜索。通过指定 CASE_ENABLED，确保索引文档的方法使

区分大小写的搜索成为可能。有关更多信息，参见第43页的『在 DBCS 文档进行模糊搜索或搜索』。

NORMALIZED

此选项仅可供精确索引使用。规则化精确索引与精确索引的不同点在于：

- 它不区分大小写；将所有不是大写的字转换为小写。
- 全大写的字不受停止字过滤影响；将对缩写（如 UK）创建索引。
- 通过试探性算法，可以将英语语言搜索词条扩充为包括引语格式，使得搜索 house 的同时还会搜索 houses。

INDEXPROPERTY SECTIONS_ENABLED DOCUMENTMODEL(S) model

选择的索引类型的特性。

SECTIONS_ENABLED 指定选择的索引类型可以包含关于文档结构的信息。

DOCUMENTMODEL/DOCUMENTMODELS *model* 指定要作为要进行索引的文档的缺省模型相关联的模型。若使用索引特性 SECTIONS_ENABLED，则必须使用模型名。若指定模型列表，则将第一个模型用作索引的缺省模型。若文档不引用模型，或搜索期间未指定模型，则创建索引期间将使用缺省模型。

可以用于模型名的字符是 a-z、A-Z 和 0-9。

指定的模型名必须与模型定义文件 `desmodel.ini` 中的模型定义相对应。

要更改与索引相关联的模型，

1. 使用 `DISABLE TEXT COLUMN` 来禁用索引
2. 使用 `ENABLE TEXT COLUMN` 来对文档重建索引，并指定另一模型名。

UPDATEFREQ update-frequency

索引更新频率，按何时进行更新以及必须在日志表中排队的最少文本文档数来指定。若在给定的日期和时间日志表中没有足够的文本文档，则不更新索引。

语法在第45页的『更新索引』中作了描述。

若不指定 `UPDATEFREQ`，则使用在文本配置设置中指定的缺省频率。

提示

若有许多表，则考虑避免使用缺省值。通过对各个表进行个别的更新频率设置，可以避免同时索引这些表而导致对系统资源的不必要的延长装入。

NONE 没有进行进一步的索引更新。这用于其中将没有进一步更改的文本列。

若已经通过 **ENABLE TEXT TABLE** 对整个表设置了更新频率，则忽略这些更新频率设置。

DIRECTORY 目录

将要在其中存储文本索引的目录路径。指定的路径与“*txinsnnn*”并置在一起，其中 *nnn* 是节点号。

这是 DB2 Text Extender 服务器运行于其上的系统的目录。如果该目录尚不存在，则创建该目录。如果该目录已经存在，确保在 UNIX 平台上，该目录具有实例拥有者的读/写许可权，并检查“受防护用户 ID”以了解它是否被指定为不同于在创建 DB2 实例时指定的“受防护用户 ID”。

若不指定 **DIRECTORY** 关键字，则使用在文本配置设置中的 **DIRECTORY** 设置的值。

若已经通过 **ENABLE TEXT TABLE** 对整个表进行了设置，则忽略此设置。

如果使用的是分区数据库，则可以对所有节点或某个范围内的节点指定一个公共索引路径名，或对每个节点指定不同的索引目录路径。

ON NODE node [TO node]

对其指定了目录路径名的节点的号码或节点的范围。

UPDATEINDEX

一个关键字，它确定是否在已完成此命令之后立即对与此句柄列相关的文本文档创建索引，而无需等待由 **UPDATEFREQ** 设置的下一次定期索引。

UPDATE

在已完成此命令之后立即进行文本文档的索引。

NOUPDATE

在由更新频率设置所设置的时间创建索引，这些更新频率设置是由 **UPDATEFREQ** 在此命令中指定的，或是由文本配置设置指定的。

若不指定此关键字，则采用文本配置设置中的值。

TABLESPACE tablespace-name

索引的表空间名，是在句柄列上以内部方式创建的。必须先前已经创建了该表空间。

COMMITCOUNT 计数

一个从 500 至 1 000 000 的值，该值指示在多少个插入或更新之后 DB2 UDB 必须发出中间落实语句。这可以避免当启用大表、列或大量外部文件时日志空间不足的情况。

用法注释

此命令将句柄列添加至指定的 DB2 表中。每个句柄列都与一个文本列相关，并且由 DB2 Text Extender 的函数使用。

若尚未启用此表以使其能够创建一个公共索引，则创建与此文本列相关的索引。

在该数据库中还创建一个日志表。该日志表用于记录对文本列的更改，即插入、更新和删除。对文本列定义插入、更新和删除触发器，以使日志表自动保持最新。

若启用的文本列属于某个表（该表是多节点节点组的一部分），则指定的索引目录必须可在所有物理节点上获得。若使用在文本配置中指定的缺省目录，则确保该路径可在该节点组的所有节点上获得。若这样不方便，则可以在 ENABLE TEXT COLUMN 命令中为每个节点指定一个特定路径。

若更改节点组的节点配置，该节点组包含为 DB2 Text Extender 启用的表，则必须重新索引该表

提示

若在此步骤中用完了日志空间，参见第67页的『启用大型表中的文本列』以获取可能的解决方案。

ENABLE TEXT FILES 命令

ENABLE TEXT FILES

此命令使得 DB2 Text Extender 能够在那些不在 DB2 UDB 数据库中的文本文件中搜索。

提示

若表中的文本列共享公共索引，则不能使用此命令，如第60页的『启用文本表（可选）』中所述。

权限

对于该表，您必须至少具有下列特权的其中之一：

ALTER 特权

SELECT 特权

UPDATE 特权。

命令语法

►►—ENABLE TEXT FILES—*table-name*—HANDLE—*handle-column-name*—►►►

┌──┴──┐ ┌──┴──┐
| text-information | | index-characteristics |
└──┬──┘ └──┬──┘

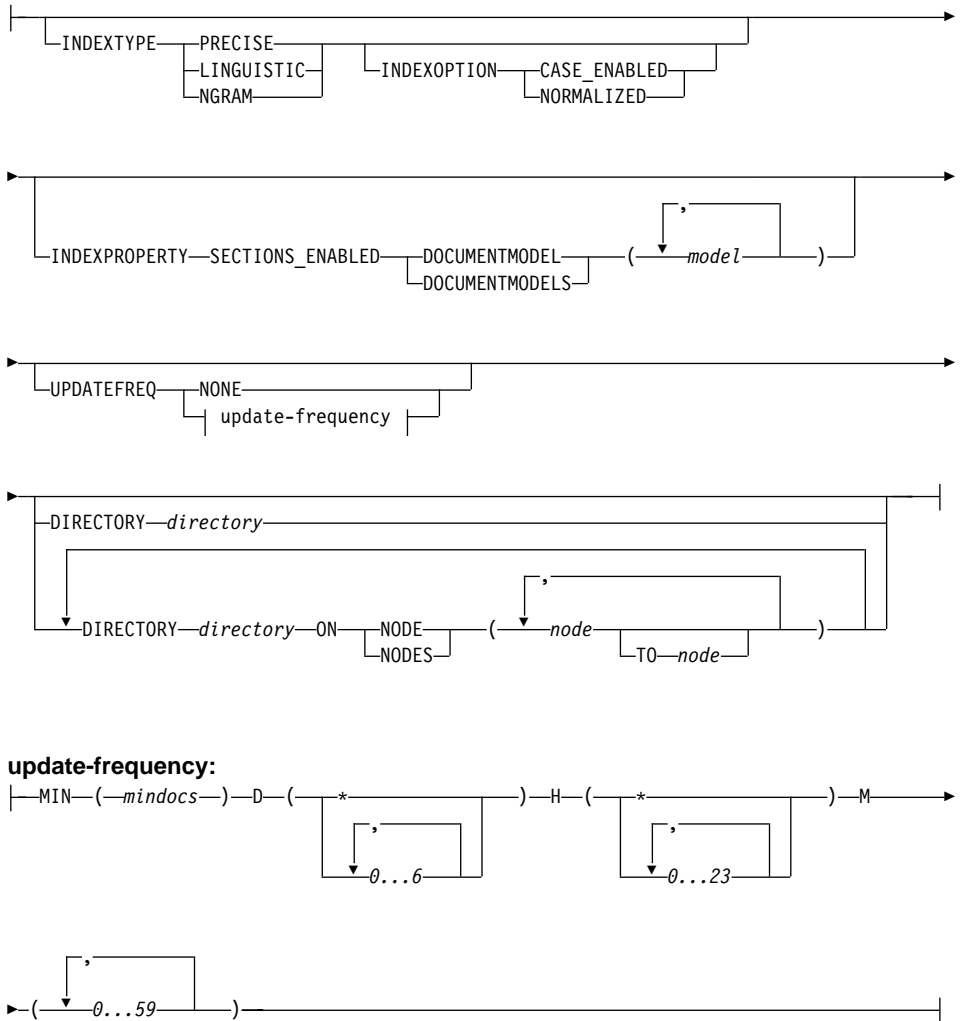
┌──┴──┐ ┌──┴──┐ ┌──┴──┐
| UPDATEINDEX—UPDATE— | | TABLESPACE—*tablespace-name*— |
└──┬──┘ └──┬──┘

┌──┴──┐
| COMMITCOUNT—*count*— |
└──┬──┘

text-information:

┌──┴──┐ ┌──┴──┐ ┌──┴──┐
| CCSID—*ccsid*— | | LANGUAGE—*language*— | | FORMAT—*format*— |
└──┬──┘ └──┬──┘ └──┬──┘

index-characteristics:



命令参数

table-name

在相连数据库中的文本表的名称，该文本表将与要索引的外部文本文件相关联。除非模式名与您的用户 ID 相同，否则该名称必须包括显式模式名（限定符）。

有关其他命令参数的说明，参见第135页的『ENABLE TEXT COLUMN』。

ENABLE TEXT FILES 命令

用法注释

此命令将句柄列添加至指定的 DB2 表中。每个句柄列都与一个外部文本文件的集合相关联，且由 DB2 Text Extender 的函数使用。创建一个与这些文件相关联的索引。

在启用了文本文件之后，使用 INIT_TEXT_HANDLE 初始化句柄列中的句柄。然后使用 UPDATE INDEX 填充索引。

若以前已经在 ENABLE TEXT FILES 中使用过一个句柄列名来标识文本列的句柄列，则不能再使用该名称。

虽然为记录对文件的更改创建了日志表，但是由于 DB2 UDB 不能控制这些文件，因而不知道这些更改，所以您必须人工激活触发器以记录这些更改。有关如何执行此操作的说明，参见第96页的『更新外部文件的索引』。

若在此步骤中用完了日志空间，参见第67页的『启用大型表中的文本列』以获取可能的解决方案。

DOCUMENTMODEL/DOCUMENTMODELS 模型 指定要作为要进行索引的文档的缺省模型相关联的模型。若使用索引特性 SECTIONS_ENABLED, 则必须使用模型名。若指定模型列表, 则将第一个模型用作索引的缺省模型。若文档不引用模型, 或搜索期间未指定模型, 则创建索引期间将使用缺省模型。

可以用于模型名的字符是 a-z、A-Z 和 0-9。

指定的模型名必须与模型定义文件 desmodel.ini 中的模型定义相对应。

要更改与索引相关联的模型,

1. 使用 DISABLE TEXT TABLE 来禁用索引
2. 使用 ENABLE TEXT TABLE 来对文档重建索引, 并指定另一模型名。

UPDATEFREQ update-frequency

索引更新频率, 按何时进行更新以及在日志表中必须有多少文本文档要排队来指定。若在给定的日期和时间日志表中没有足够的文本文档, 则不更新索引。

语法在第45页的『更新索引』中作了描述。

若不指定 UPDATEFREQ, 则使用在文本配置设置中指定的缺省频率。

NONE 没有进行进一步的索引更新。这用于其中将没有进一步更改的文本列。

提示

若有许多表, 则考虑避免使用缺省值。通过对各个表进行个别的更新频率设置, 可以避免同时索引这些表而导致对系统资源的不必要的延长装入。

DIRECTORY directory

将要在其中存储文本索引的目录路径。指定的路径与 “txinsnn” 并置在一起, 其中 *nnn* 是节点号。

这是 DB2 Text Extender 服务器运行于其上的系统的目录。如果该目录尚不存在, 则创建该目录。如果该目录已经存在, 确保在 UNIX 平台上, 该目录具有实例拥有者的读/写许可权, 并检查“受防护用户 ID”以了解它是否被指定为不同于在创建 DB2 实例时指定的“受防护用户 ID”。

若不指定 DIRECTORY 关键字, 则使用在文本配置设置中的 DIRECTORY 设置的值。

ENABLE TEXT TABLE 命令

若使用分区的数据库，则可以对所有节点或一系列节点指定一个公共索引路径名。或为每个节点指定不同的索引目录路径。

ON NODE node [TO node]

对其指定了目录路径名的节点的号码或节点的范围。

用法注释

创建了与此表中的所有文本列相关联的新的文本索引。当想要对一个表的所有文本列有一个公共索引、而不是对每个文本列单独有一个索引时，执行此操作。

当已经启用了表时，然后必须为想要在其中进行搜索的每个文本列运行 `ENABLE TEXT COLUMN`。

在数据库中创建一个日志表。该表用于记录更改，即在稍后启用的文本列中的插入、更新和删除。

当启用文本列时，创建监控文本更改的触发器，并自动在日志表中记录将需要索引哪些文档。

`DB2 Text Extender` 按 `UPDATEFREQ` 关键字指定的频率周期性地索引在日志表中列示的文本文档。

GET ENVIRONMENT

此命令显示环境变量的设置。

权限

无。

命令语法

▶▶—GET ENVIRONMENT—▶▶

命令参数

无。

用法注释

以下为所显示的环境变量：

DB2INSTANCE

DB2 UDB 实例名。

DB2DBDFT

数据库的缺省名称。

DB2TX_INSTOWNER

DB2 Text Extender 实例名。仅当连接至 UNIX 服务器时才需要此项。

DB2TX_INSTOWNERHOMEDIR

实例拥有者的主目录。仅当连接至 UNIX 服务器时才需要此项。

GET INDEX SETTINGS

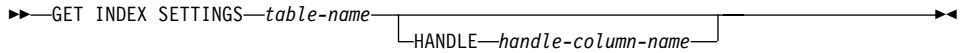
此命令显示索引的设置，显示下列各项：

- 索引类型
- 索引选项（可选）
- 更新索引选项
- 索引目录
- 更新频率
- 缺省模型。

权限

无。

命令语法



命令参数

table-name

在相连的数据库中的、其索引设置将被显示的文本表的名称。除非模式名与您的用户 ID 相同，否则该名称必须包括显式模式名（限定符）。

HANDLE handle-column-name

其索引设置将被显示的句柄列的名称。

用法注释

若启用该表作为多索引表，则此命令显示表中所有启用的文本列的索引设置。若提供了 *handle-column-name*，则此命令显示指定列的索引设置。

若该表为公共索引表，则显示公共索引的设置。若提供了 *handle-column-name*，则忽略它。

若启用表或列时使用了索引特性 `SECTIONS_ENABLED`，则命令 `GET INDEX SETTINGS` 显示索引的缺省模型。缺省模型是您在启用期间指定的模型名，或者是指定的一系列模型名中的第一个模型名。下面是一个示例：

Current index settings:

```
Index type          (INDEXTYPE) = LINGUISTIC  
Default model      (DOCUMENTMODEL) = mymodel
```

GET INDEX SETTINGS 命令

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

GET INDEX STATUS 命令

GET INDEX STATUS

此命令显示给定句柄列或表的下列索引状态信息:

- 搜索功能是否可用
- 索引更新功能是否可用
- 重组功能是否可用
- 调度的文档数
- 创建索引的文档数
- 主索引中创建索引的文档数
- 辅助索引中创建索引的文档数
- 错误事件。

权限

无。

命令语法

►► GET INDEX STATUS—*table-name* ─┬─►
└─┬─► HANDLE—*handle-column-name* ─┬─►

命令参数

table-name

在相连的数据库中的文本表的名称，该文本表包含其状态将被显示的文本列。除非模式名与您的用户 ID 相同，否则该名称必须包括显式模式名（限定符）。

HANDLE handle-column-name

其状态将被显示的句柄列的名称。

用法注释

对于多索引表，必须指定句柄列的名称。

GET STATUS

此命令显示有关数据库、表或文本列的已启用状态的信息。

权限

无。

命令语法

▶▶—GET STATUS—◀◀

命令参数

无。

用法注释

此命令显示数据库是否已启用、数据库中已启用的文本表的名称、已启用的文本列及其相关的句柄列的名称、以及外部文件句柄列的名称。

GET TEXT CONFIGURATION

此命令显示相连的数据库的文本配置的缺省设置。

要更改这些缺省设置，使用第125页的『CHANGE TEXT CONFIGURATION』。

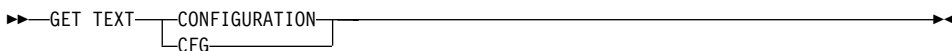
权限

对于该表，您必须至少具有下列特权的其中之一：

SELECT 特权

命令语法

► GET TEXT CONFIGURATION ◄



命令参数

无。

用法注释

有关文本配置信息的示例，参见第101页的『显示文本配置设置』。

GET TEXT INFO

此命令显示文本列的文本信息设置:

- CCSID
- 语言
- 格式。

权限

无。

命令语法

```
▶▶ GET TEXT INFO table-name [ HANDLE handle-column-name ] ▶▶
```

命令参数

table-name

在相连的数据库中的文本表的名称，该文本表包含其文本信息设置将被显示的文本列。除非模式名与您的用户 ID 相同，否则该名称必须包括显式模式名（限定符）。

HANDLE handle-column-name

其文本信息设置将被显示的句柄列的名称。

用法注释

若给出了句柄列名，则仅显示此列的文本信息。

若未给出句柄列名，则显示此表中每个已启用列的文本信息。

QUIT 命令

QUIT

此命令停止 DB2 Text Extender 命令行处理器并将控制返回至操作系统。

权限

无。

命令语法

▶—QUIT—▶▶

命令参数

无。

用法注释

与数据库的连接已终止。

REORGANIZE INDEX

若文本列经常更新，则在索引中进行搜索会变得效率低下。要使搜索重新变得有效率，就必须重组索引。虽然 DB2 Text Extender 知道索引何时需要重组并自动地在后台重组索引，但在以下情况下，需要使用 REORGANIZE INDEX 人工重组索引您可以使用命令 GET INDEX STATUS 来了解索引是否需要重组。

权限

无。

命令语法

```
►► REORGANIZE INDEX table-name ───────────────────────────────────►  
└── HANDLE handle-column-name ───────────────────────────┘
```

命令参数

table-name

相连的数据库中要重组其索引的文本表的名称。除非模式名与您的用户 ID 相同，否则该名称必须包括显式模式名（限定符）。

HANDLE handle-column-name

要重组其索引的句柄列的名称。

用法注释

对于多索引表，必须指定句柄列名。

虽然 REORGANIZE INDEX 运行时可以对索引进行搜索，但却不能执行索引更新。

RESET INDEX STATUS

当表或列的索引状态显示搜索不可用或更新不可用时，在索引期间发生了一个阻止您使用该索引的错误。

此命令重设索引状态以使您可以继续使用它。在重设索引状态之前，检查可能记录在索引的日志表中的任何错误（参见第103页的『显示错误事件』）。

权限

无。

命令语法

```
▶▶—RESET INDEX STATUS—table-name—HANDLE—handle-column-name—▶▶
```

命令参数

table-name

在相连的数据库中的文本表的名称，该文本表包含其状态将被重设的文本列。除非模式名与您的用户 ID 相同，否则该名称必须包括显式模式名（限定符）。

HANDLE handle-column-name

其状态将被重设的句柄列的名称。

用法注释

对于多索引表，必须指定句柄列名。

对于公共索引表，重设此表中每个已启用的列。

UPDATE INDEX

此命令立即开始索引。它使索引成为最新的，以反映与索引相关联的文本列的当前内容。

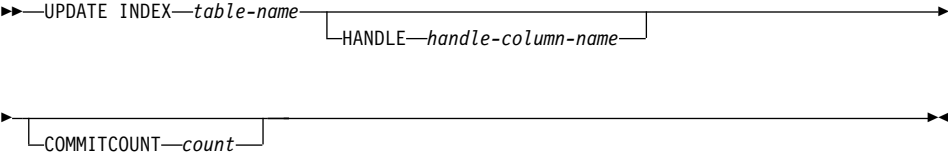
要在索引中反映对外部文件的更新，必须通过在相应的句柄列上发出一条更新语句，来在日志表中强行放入一个“更改”项目。有关示例，参见第96页的『更新外部文件的索引』。

权限

对于该表，您必须至少具有下列特权的其中之一：

- ALTER 特权
- SELECT 特权
- UPDATE 特权。

命令语法



命令参数

table-name

在相连的数据库中的文本表的名称，该文本表包含其索引将被更新的文本列。这也可以是公共索引表的名称。除非模式名与您的用户 ID 相同，否则该名称必须包括显式模式名（限定符）。

HANDLE handle-column-name

若这是公共索引表，则无需 *handle-column-name*，并忽略它。要更新的索引是与整个表相关联而不是与个别的文本列相关联。

若这是多索引表，则 *handle-column-name* 是其索引将被更新的句柄列的名称。

COMMITCOUNT count

一个从 500 至 1 000 000 的值，该值指示在多少个插入或更新之后 DB2 UDB 必须发出中间落实语句。这可以避免当启用大表、列或大量外部文件时日志空间不足的情况。

UPDATE INDEX 命令

第10章 服务器的管理命令

本章描述服务器的管理命令的语法。服务器管理包括这样一些任务，可以执行这些任务以启动、停止和检查 DB2 Text Extender 服务器的状态，以及创建样本数据库和样本表。第20页的『设置和维护 DB2 Text Extender 服务器』描述如何使用这些命令

命令	目的	页
TXICRT	创建 DB2 Text Extender 实例	162
TXIDROP	卸下 DB2 Text Extender 实例	164
TXILIST (仅限于 UNIX)	列示 DB2 Text Extender 实例 (仅限于 UNIX)	165
TXIPCLEAN (仅限于 UNIX)	在系统故障之后清除系统资源 (仅限于 UNIX)	166
TXNADD	添加 DB2 Text Extender 服务器	167
TXNCHECK	检查 DB2 Text Extender 数据库完整性	168
TXNDROP	卸下 DB2 Text Extender 服务器	169
TXSAMPLE	创建和启用样本表	170
TXSTART	启动 DB2 Text Extender 服务	171
TXSTATUS	显示搜索服务的状态	172
TXSTOP	停止 DB2 Text Extender 服务	173
TXTHESC	编译同义词典定义文件	174
TXTHESN	编译 Ngram 同义词典定义文件	176
TXTRACE	产生跟踪信息	178
TXVERIFY	创建和启用样本数据库	183
TXWIZARD	进行索引和搜索的示例程序 (仅限于 AIX、Windows NT 和 Windows 2000)	184

TXICRT 命令

TXICRT

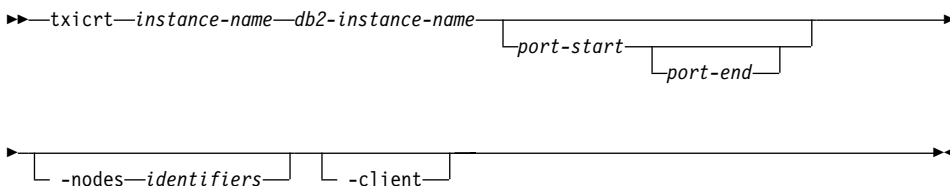
此命令创建 DB2 Text Extender 实例。

权限

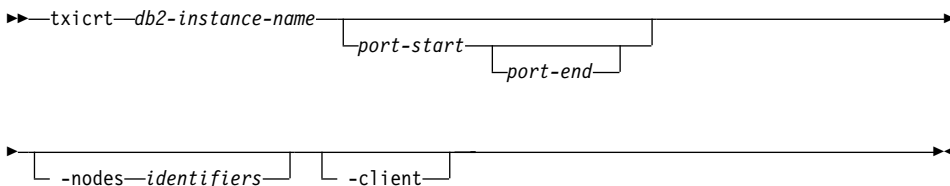
(仅限于 UNIX) 超级用户

命令语法

对于 UNIX:



对于 Windows NT 和 Windows 2000:



命令参数

instance-name

(仅限于 UNIX) 实例拥有者的用户 ID。

db2-instance-name

要与正创建的实例相关联的 DB2 实例的名称。

port-start/port-end

使用多个物理节点时，要使用的 TCP/IP 端口范围。若只使用逻辑节点，则不需要此项。

端口范围必须与您想要在其上工作的节点数相等。若使用一个节点，则仅指定 *port-start* 参数。要支持节点的子集，使用 *port-last* 参数。

若该实例是在单一分区数据库上创建的，则不要指定端口范围。

只使用尚未在 TCP/IP etc/services 文件中列示的端口号。不要对仅客户机安装指定端口号。

-nodes *identifiers*

节点标识符列表。示例: `-nodes 0 1 2`。

若您的表属于节点组中的特定节点, 使用此参数。

若未指定此参数, 则使用所有节点, 即在 `db2nodes.cfg` 中定义的所有节点。

不要对仅客户机安装指定节点标识符。

-client

若只安装了 DB2 Text Extender 客户机, 则指定此参数若不指定此参数, TXICRT 会自动配置 DB2 Text Extender 的客户机 / 服务器实例。

用法注释

仅在已创建了 DB2 Text Extender 实例的节点上才可能启用文本表或列。

必须已为用户 ID 创建了 DB2 实例。

在 UNIX 安装中, 此实例特定命令在 `/usr/lpp/db2tx_07_01/instance` 目录中。

在 Windows NT 或 Windows 2000 系统上, 缺省实例是在非分区数据库环境中的安装期间创建的。

TXIDROP 命令

TXIDROP

此命令将 DB2 Text Extender 实例与其所有索引一起卸下。

权限

(仅限于 UNIX) 超级用户

命令语法

对于 UNIX:

```
►►txidrop—instance-name—db2-instance-name—————►◄
```

对于 Windows NT 和 Windows 2000:

```
►►txidrop—instance-name—————►◄
```

命令参数

instance-name

要卸下的实例的名称。

db2-instance-name

与要卸下的实例相关联的 DB2 实例名。

用法注释

在卸下实例之前，禁用任何为它启用的数据库。

在 UNIX 安装中，此实例特定命令在 `/usr/lpp/db2tx_07_01/instance` 目录中。

TXILIST（仅限于 UNIX）

此命令列示了当前 DB2 Text Extender 版本的实例。

权限

SYSADM、DBADM

命令语法

▶—txilist—▶

命令参数

无。

用法注释

若安装了多个版本的 DB2 Text Extender，则此命令仅列示您当前正在使用的 DB2 Text Extender 版本（即，当前目录确定的版本）的实例。

TXIPCLEAN 命令 (仅限于 UNIX)

TXIPCLEAN (仅限于 UNIX)

此命令清除已注册用户的系统资源，即，它终止 `desssdem` 进程，清除共享内存、程序段信息和信号。

权限

`smadmin` 组的成员

命令语法

▶—txipclean—▶▶

命令参数

无。

用法注释

谨慎使用此命令；它不仅清除 `DB2 Text Extender` 资源，而且还清除了已注册用户的共享资源。始终注册为实例所有者以限制对实例资源的清除。

可能在某些时候您不能正确地启动或停止 `DB2 Text Extender`。在这种情况下，尝试在运行 `TXIPCLEAN` 之前首先停止 `DB2` 和 `DB2 Text Extender`。

TXNADD

此命令在指定的节点上创建一个附加的 DB2 Text Extender 服务器 TXSTART 命令接着启动该服务器。

权限

(仅限于 UNIX) SYSADM, DBADM

命令语法

►►—txnadd—*node-number*—*port*—►►

命令参数

node-number

要添加的节点的号码。完全按照在 `db2nodes.cfg` 文件中所指定的那样指定节点号。

port

要用于添加的节点的端口号。若要添加物理节点或节点配置包含物理节点，则必须指定端口号。

用法注释

使用 TXNCHECK 来检查为与数据库保持同步必须重新创建哪些索引。

提示

若对具有逻辑节点的节点配置添加物理节点，则必须首先卸下实例，然后重新创建该实例，并为所有节点指定端口号。

TXNCHECK 命令

TXNCHECK

此命令检查给定数据库的 DB2 Text Extender 完整性。

权限

(仅限于 UNIX) SYSADM, DBADM

命令语法

```
▶▶ txncheck database-name ▶▶
```

命令参数

database-name

要检查的数据库的名称。若未指定数据库名, 则使用 DB2 缺省数据库的名称。

用法注释

若更改了启用了 DB2 Text Extender 的表所在的节点组, 则必须重新索引该表使用 TXNCHECK 来显示因节点组的重新分发而需要重新索引哪些启用了 DB2 Text Extender 的表。

TXNDROP

此命令除去给定节点的 DB2 Text Extender 服务器。

权限

(仅限于 UNIX) SYSADM, DBADM

命令语法

▶—txndrop—*node-number*—————▶◀

命令参数

node-number

要卸下的节点的号码。完全按照在 `db2nodes.cfg` 文件中所指定的那样指定节点号。

用法注释

使用 TXNCHECK 来检查为与数据库保持同步必须重新创建哪些索引。

TXSAMPLE 命令

TXSAMPLE

此命令创建样本表，将样本英文文档装入列 COMMENT，并启用文本列。参见第 16 页的『准备样本数据库以进行安装验证』。

权限

(仅限于 UNIX) SYSADM, DBADM

命令语法

```
txsample database-name user-id password
```

命令参数

database-name

要在其中创建样本表的数据库的名称，已经由 DB2 Text Extender 启用。

user-id

仅当从客户机工作站工作时，才需要它。

password

仅当从客户机工作站工作时，才需要它。

用法注释

如果数据库 CCSID 不是 850，查看对 Ngram 索引类型启用步骤的示例，并更改 CCSID 以匹配您的数据库 CCSID。

提示

此命令还可在客户机工作站上使用。

TXSTART

此命令启动 DB2 Text Extender 服务。

权限

(仅限于 UNIX) SYSADM, DBADM

命令语法

► txstart NODENUM nodenum ►

命令参数

NODENUM nodenum

要启动的服务器的节点号。若未指定节点号，则启动所有服务器。

用法注释

运行此命令:

- 当在 SM 管理组中使用了用户 ID 进行注册时
- 无论何时停止和重新启动服务器系统时

如果在启动 DB2 Text Extender 服务时遇到问题，停止 DB2 并运行命令 TXIPCLEAN。参见第166页的『TXIPCLEAN (仅限于 UNIX)』。

TXSTATUS 命令

TXSTATUS

此命令显示 DB2 Text Extender 是否已启动并在运行。

权限

(仅限于 UNIX) SYSADM, DBADM

命令语法

► txstatus [NODENUM *nodenum*] ◄

命令参数

NODENUM *nodenum*

要检查其状态的服务器的节点号。若未指定节点号，则显示所有服务器的状态。

TXSTOP

此命令停止 DB2 Text Extender 服务。

权限

(仅限于 UNIX) SYSADM, DBADM

命令语法

►► txstop [NODENUM *nodenum*] ►►

命令参数

NODENUM nodenum

要检查其状态的服务器的节点号。若未指定节点号，则停止所有服务器。

用法注释

此命令不停止 DB2。

TXTHESC

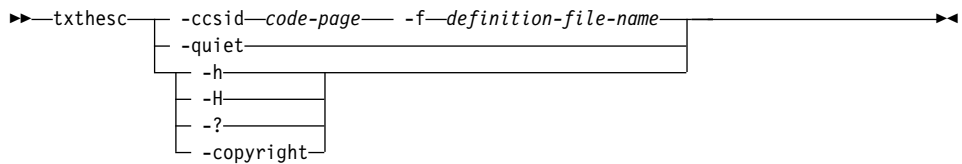
此命令编译同义词典定义文件。此同义词典只能用于对精确或语言索引的搜索。

有关编译 Ngram 同义词典定义文件，参见第176页的『TXTTHESN』。

权限

(仅限于 UNIX) 无。

命令语法



命令参数

-f *definition-file-name*

包含同义词典定义的 SGML 文件的名称。文件名必须包含文件的绝对路径或相对路径。

在定义文件所在的目录中生成同义词典词典。它与定义文件同名，但扩展名为 th1 至 th6。

提示

因为同义词典文件在同名时会被覆盖，因此对每个同义词典使用不同的目录。

-ccsid *code-page*

写同义词典定义文件所用的代码页。当前，仅支持代码页 850。

-quiet 不显示输出信息。

-copyright

返回产品的内部构建号。报告问题时，使用此编号。

-h, -H, or -?

显示帮助信息。

用法注释

使用此命令来将标准同义词典定义文件编译成二进制同义词典定义格式。定义文件必须处于 SGML 格式。

要使用编译过的同义词典文件，将其移至服务器实例的词典目录，然后在搜索期间指定文件的位置。

OS/2 和 Windows 系统上的词典目录是：

```
drive:\dmb\db2tx\dict
```

在 AIX、HP-UX 和 SUN-Solaris 系统上，词典目录是：

```
DB2TX_INSTOWNER/db2tx/dicts
```

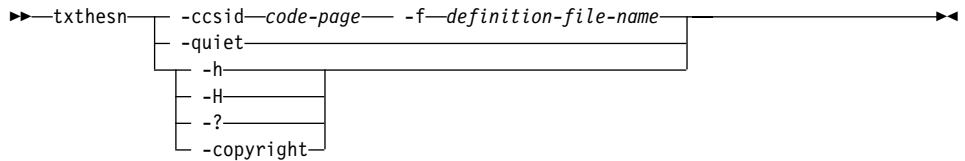
TXTHESN

此命令编译 Ngram 同义词典定义文件。此同义词典只能用于对 Ngram 索引的搜索。

权限

(仅限于 UNIX) 无。

命令语法



命令参数

-f *definition-file-name*

包含同义词典定义的文件名称。文件名必须包含文件的绝对路径或相对路径。文件名限制为 8+3 个字符。扩展名是可选的。

在定义文件所在的目录中生成同义词典词典。它与定义文件同名，但扩展名为 wdf、wdv、grf、grv、MEY、ROS、NEY、SOS 和 lkn，其中 *n* 是一个数字。

提示

因为同义词典文件在同名时会被覆盖，所以应对每个同义词典使用不同的目录。

-ccsid *code-page*

写同义词典定义文件所用的代码页。有关受支持的代码页的列表，参见第 35 页的『CCSID』。

-quiet 不显示输出信息。

-copyright

返回产品的内部构建号。报告问题时，使用此编号。

-h, -H, or -?

显示帮助信息。

用法注释

使用此命令来将同义词典定义文件编译成二进制同义词典定义格式。定义文件必须处于第254页的『创建 Ngram 同义词典』所述的格式。

要使用编译过的同义词典文件，将其移至服务器实例的词典目录，然后在搜索期间指定文件的位置。

OS/2 和 Windows 系统上的词典目录是：

```
drive:\dmb\db2tx\dict
```

在 AIX、HP-UX 和 SUN-Solaris 系统上，词典目录是：

```
DB2TX_INSTOWNER/db2tx/dicts
```

TXTRACE 命令

TXTRACE

此命令将处理信息写入共享内存中的跟踪缓冲区。此信息可用于跟踪故障。可将此信息以二进制从跟踪缓冲区写入文件，以便稍后在已经断开了跟踪时格式化它，或者可以在跟踪仍打开时格式化该信息并将它写入文件。

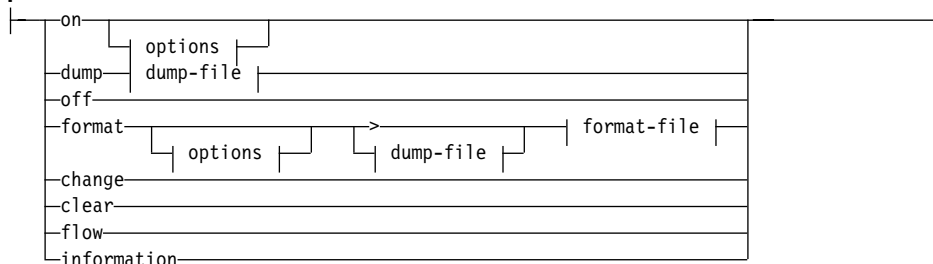
权限

(仅限于 UNIX) SYSADM, DBADM

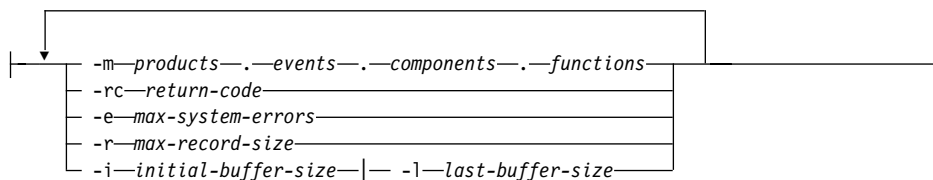
命令语法

► txtrace parameters ◄

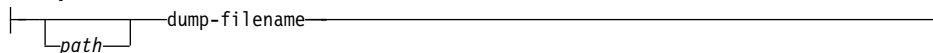
parameters:



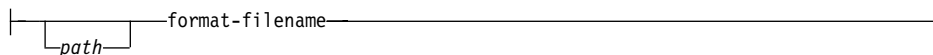
options:



dump-file:



format-file:



命令参数

注:

-u 选项也可用于所有 TXTRACE 参数以显示关于该参数的信息。

on 启动跟踪设施。

dump | dmp

将跟踪信息以二进制写入文件。

off 停止跟踪设施。

format | fmt

格式化二进制跟踪信息。可在关闭了跟踪时，通过指定转储文件名以及要保存格式化的跟踪信息的文件的名称来格式化转储文件。要在跟踪仍打开时从跟踪缓冲区直接格式化跟踪信息，输入：`destrc format > filename.tmp`。

change | chg

更改跟踪掩码、`maxSysErrors` 或 `maxRecordSize`。

clear | clr

清除跟踪。

flow | flw

显示跟踪的控制流。

information | info | inf

获得关于跟踪的信息。

options

在打开跟踪（这减少共享内存使用率）时或格式化跟踪信息时过滤跟踪信息。除非跟踪非常大，否则写入全部跟踪信息然后在格式化期间过滤该信息通常更容易。

-m 添加“掩码”以指定将要在跟踪中包括哪些事件、部件及函数。缺省值是跟踪每一件事。掩码有四个部分，以句点分隔；例如：`2.2-6.1,3.*` 可使用“-”作为分隔符以指定范围，或使用“;”作为分隔符以指定列表。例如：`2-6` 表示只包括其 ID 在从 2 至 6 的范围内的的事件。要只包括部件 2 和 6，指定 `2,6`

products

产品 ID。DB2 Text Extender 的产品 ID 是“2”。TextMiner 的产品 ID 是“3”。

events 要包括在跟踪中的事件类型的集合:

0 system_error

TXTRACE 命令

1	system_error
2	system_error
3	non-fatal_error
4	non-fatal_error
5	api_errcode
7	fnc_errcode
8	trap_error
10	api_entry
11	api_exit
13	api_retcode
15	api_data
30	fnc_entry
31	fnc_exit
33	fnc_retcode
35	fnc_data

components

要跟踪的部件。

DB2 Text Extender 的部件 ID 是:

1	COMMAND_LINE_INTERFACE
2	UDF
3	STORED_PROCEDURES
4	ADMINISTRATION
5	INDEX_CONTROL
6	LIBRARY_SERVICES
7	DES_PARSER
8	DES_DEMON
9	DES_API
10	SERVICES

TextMiner 的部件 ID 是:

1	automachine
2	bgproc (后台处理)
3	cluster
4	common
5	commsvc (公共服务)
6	communic (通信)
7	daemon
8	dsclient
9	environ (环境)

10	glue
11	idxcomm (索引构建, 公共部件)
12	libsrv (库服务)
13	search
14	trace
15	guru
16	indexbld (索引构建, 仅用于 tm)
17	indexeng (索引引擎, 仅用于 tm)
18	smsearch
19	search (搜索引擎, 仅用于 tm)
20	tmsearch
21	gtrcm (gtr, 公共部件)
22	gtrsrch (搜索, 仅用于 gtr)
23	gtridx (索引构建, 仅用于 gtr)

functions

星号 (*)。要跟踪的函数组。除非“IBM 支持中心”指导执行其他操作, 否则使用星号 (*) 跟踪所有函数。

-rc *return-code*

将 *return-code* 当作系统错误。

-e *max-system-errors*

整数。发生了这样多个错误之后停止跟踪。缺省值为 1, 指定出现第一个系统错误时, 抑制较低严重性事件的所有后续跟踪。当只对第一个主要错误感兴趣时, 这是可以接受的, 但是如果您想查看初始系统错误之后的全部跟踪, 则应指定较大的数字 (如 -e 50)。跟踪目的地是共享的内存。

-r *max-record-size*

整数。在已向跟踪文件写入了此数量的记录之后停止跟踪。缺省值为 16 KB。

-i *initial-buffer-size*

整数。从跟踪的开头保持此数量的记录。若指定了 -i, 则缺省值为 16 KB。在 UNIX 系统上, 建议缓冲区大小为 2 MB。

若 -i 或 -l 都没有指定, 则将 -l 作为缺省值。

若指定 -i, 则不出现绕回; 当记录容量超过 *max-record-size* 时, 即使清除所有跟踪项, 也不再写入任何跟踪项。要使新跟踪项能够写入, 增加缓冲区大小, 关闭跟踪然后再打开。

TXTRACE 命令

-l *last-buffer-size*

整数。从跟踪的结束保持此数量的记录。缺省值为 16 KB。在 UNIX 系统上，建议缓冲区大小为 2 MB。

路径 存储相应文件的目录。

dump-filename

包含二进制跟踪信息的文件的名称。

formatted-filename

包含格式化的跟踪信息的文件的名称。

示例

参见第22页的『跟踪故障』。

TXVERIFY

此命令创建一个样本数据库并启用它。

权限

(仅限于 UNIX) SYSADM, DBADM

命令语法

```
▶▶ txverify database-name user-id password ▶▶
```

命令参数

database-name

要在其中创建样本表的数据库的名称。

user-id

仅当从客户机工作站工作时，才需要它。

password

仅当从客户机工作站工作时，才需要它。

用法注释

此命令还可以在客户机工作站上使用。

TXWIZARD

此命令仅在 AIX、Windows NT 和 Windows 2000 系统中可用。它举例说明了如何对选择的表或列创建索引及如何对这些索引进行搜索。

命令语法

►—txwizard—◄◄

命令参数

无。

用法注释

对于能够使用 DB2 Text Extender 搜索功能，对所选表列进行搜索所需的步骤，本样本向导将指导您如何完成可使用随样本向导一起提供的 Java 源代码来构建您自己的文本准备和搜索应用程序，或了解如何在 Java 环境内使用和调用 DB2 Text Extender 功能。

在 AIX 中，向导的 Java 源代码存储在以下目录中：

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

在 Windows NT 和 Windows 2000 中，它存储在以下目录中：

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

第11章 搜索函数

DB2 Text Extender 提供了一些 SQL 函数，这些函数可在存储在 DB2 UDB 中的文本文档进行搜索，并使用搜索的结果。某些函数的参数是随 DB2 Text Extender 提供的，被称为单值类型的数据类型。

本章描述 DB2 Text Extender SQL 函数和单值类型。

DB2 Text Extender 提供一个 DB2 UDB 命令行处理器输入文件，称为 txsample.udf。它包含 DB2 Text Extender 函数的示例，这些函数是对第16页的『准备样本数据库以进行安装验证』中所述的样本表运行的。使用此文件来查看文本准备和搜索函数的语法示例。

DB2 Text Extender 单值类型

单值类型	源数据类型	注解
DB2TEXTH	VARCHAR(60) FOR BIT DATA	<p>文本句柄。 包含索引存储在文本列中的文本文档所需信息的变长字符串。句柄中的信息包括文档 ID、要在其中索引文本的服务器的名称、索引名，以及关于文本文档的信息。</p> <p>句柄存储在 DB2 Text Extender 创建并与每个文本列关联的列中。</p> <p>函数 HANDLE 和 INIT_TEXT_HANDLE 返回此数据类型。</p>
DB2TEXTFH	VARCHAR(210) FOR BIT DATA	<p>文件句柄。 包含索引外部文本文件所需信息的变长字符串，外部文件是指在 DB2 UDB 的控制之外存储的文件。文本句柄中的信息包括文档 ID、要在其中索引文本的服务器的名称、索引名、关于文本文件的信息，以及关于文件位置的信息。</p> <p>文件句柄存储在由 DB2 Text Extender 创建并与每组外部文件相关联的列中。</p> <p>函数 FILE 和 INIT_TEXT_HANDLE 返回此数据类型。</p>

DB2 Text Extender 单值类型

单值类型	源数据类型	注解
DB2TEXTHLISTP	VARCHAR(16) FOR BIT DATA	句柄列表指针。 一个指向句柄列表的指针，这些句柄与搜索所找到的文本文档相关联。 函数 HANDLE_LIST 返回此数据类型。
DB2TEXTFHLISTP	VARCHAR(16) FOR BIT DATA	句柄列表指针。 一个指向句柄列表的指针，这些句柄与搜索所找到的外部文件相关联。 函数 HANDLE_LIST 返回此数据类型。

DB2 Text Extender 函数概述

搜索函数	目的	页
CCSID	从句柄返回 CCSID	187
CONTAINS	在特定文档中进行文本搜索	188
FILE	返回或更改现存句柄中的文件的路径和名称	189
FORMAT	返回或更改句柄中的文档格式设置	190
HANDLE ¹	从句柄列表返回句柄	191
HANDLE_LIST ¹	搜索并返回句柄列表	192
INIT_TEXT_HANDLE	返回包含如格式及语言设置等信息的部分初始化的句柄	193
LANGUAGE	返回或更改句柄中的语言设置	194
NO_OF_DOCUMENTS ¹	返回在句柄列表中列示的文档数	195
NO_OF_MATCHES	搜索并返回找到的匹配数	196
RANK	搜索并返回找到的文本文档的等级值	197
REFINE	取一个搜索变元和一个细化搜索变元并返回一个组合搜索变元	198
SEARCH_RESULT	返回带有指定搜索字符串的搜索结果的中间表	199

第73页的『第6章 如何进行搜索』中提供了一些 DB2 Text Extender 用法的示例。

1. 这些搜索功能是较早发行版的 DB2 Text Extender 的功能部件。由于兼容性原因，这些功能继续受支持。但他们的功能已被 SEARCH_RESULT 搜索功能。

CCSID

CCSID 函数从句柄返回 CCSID (数据类型 SMALLINT)。这是用于索引相应的文本文档的 CCSID 参数。在第35页的『CCSID』中作了描述。通过 ENABLE TEXT COLUMN 命令为每个文本列设置它。

函数语法

►►—CCSID—(—*handle*—)—————►

函数参数

handle

一个表达式，其结果是 DB2TEXTFH 或 DB2TEXTH 类型的值。通常为要从中返回 CCSID 设置的句柄列的名称。

CONTAINS 函数

CONTAINS

CONTAINS 函数在特定文本文档中搜索文本。若文档包含该文本，它返回 INTEGER 值 1。否则，返回 0。

函数语法

►—CONTAINS—(—*handle*—,—*search-argument*—)——►

函数参数

handle

一个表达式，其结果是 DB2TEXTFH 或 DB2TEXTH 类型的值。通常为包含要搜索的文本文档的句柄的句柄列的名称。

search-argument

包含要搜索的词条、类型为 LONG VARCHAR 的字符串。参见第201页的『第12章 搜索变元的语法』。

FILE

FILE 函数执行下列其中之一：

- 返回句柄中的路径和文件名
- 更改句柄中的路径和文件名，并返回该路径和文件名。

返回的句柄是 DB2TEXTFH 类型的值。

函数语法

►► FILE (—*handle*—) —————►►

►► FILE (—*handle*—, —*file-name*—) —————►►

函数参数

handle

一个表达式，其结果是 DB2TEXTFH 类型的值。通常为要从中返回文件名的句柄列的名称。

file-name

一个 VARCHAR(150) 类型的字符串，指定将要与句柄相关联的外部文件的新绝对路径和文件名。例如，该路径可能是一个 LAN 驱动器或一个安装了 NFS 的驱动器。文件存取许可权必须允许 DB2 UDB 实例拥有者存取该文件。

FORMAT

FORMAT 函数执行下列其中之一:

- 返回在句柄中指定的文档格式
- 更改文档的句柄中的格式规范, 并返回更改了的句柄。

返回的文档格式是 VARCHAR(30) 类型的字符串。返回的句柄是 DB2TEXTFH 或 DB2TEXTH 类型。

这是用于索引相应的文本文档的格式参数。受支持的文档格式列示在第32页的『受支持的文档格式』中。

函数语法

(1)
▶▶—FORMAT——(—*handle*—)————▶▶

注:

- 1 返回格式值, 类型为 VARCHAR(30)。

(1)
▶▶—FORMAT——(—*handle*—,—*format*——)————▶▶

注:

- 1 返回一个句柄, 类型为 -DB2TEXTFH 或 DB2TEXTH。

函数参数

handle

一个表达式, 其结果是 DB2TEXTFH 或 DB2TEXTH 类型的值。通常为要从中返回或设置格式设置的句柄列的名称。

format

数据类型 VARCHAR(30) 的新文档格式设置。

若指定了格式, 则在句柄中设置此文档格式; 在此情况下, 返回的是句柄而不是格式设置。

HANDLE

HANDLE 函数返回 DB2TEXTFH 或 DB2TEXTH 类型的句柄，该句柄是按句柄列表中的顺序号选择的。

HANDLE 函数只用在 INSERT 语句中。

提示

此功能是较早版本的 DB2 Text Extender 的功能部件由于兼容性原因，这些功能仍然受支持。但 SEARCH_RESULT 搜索功能已替代了其功能。

函数语法

►►—HANDLE—(—*handle-list*—,—*integer*—)—————►►

函数参数

handle-list

一个表达式，其结果是 DB2TEXTHLISTP 或 DB2TEXTFHLISTP 类型的值。它由 HANDLE_LIST 函数返回。

这是一个指向搜索所找到的文档的句柄列表的指针。

integer

指示要返回列表中的哪一个句柄的 INTEGER 值。

HANDLE_LIST

HANDLE_LIST 函数使用搜索变元搜索文本文档，并且返回 DB2TEXTHLISTP 或 DB2TEXTFHLISTP 类型的值，指向找到的文档的句柄列表。

注：指向此列表的指针仅在使用此函数的 SQL 语句范围内是可存取的。

若未找到包含搜索词条的文本文档，则该句柄列表为空。使用 NO_OF_DOCUMENTS 函数查找该列表是否为空。

若搜索存储在基本表列中的文本，则每条语句仅使用一次此函数。若搜索存储在视图列中的文本，并且该视图是从多个基本表制成的（即联合视图），则使用每个句柄列的一个句柄作为 HANDLE_LIST 函数的句柄输入自变量。

提示

此功能是较早版本的 DB2 Text Extender 的功能部件由于兼容性原因，这些功能仍然受支持。但 SEARCH_RESULT 搜索功能已替代了其功能。

函数语法

►►HANDLE_LIST(—*handle*—,—*search-argument*—)◄◄

函数参数

handle

一个表达式，其结果是 DB2TEXTH 或 DB2TEXTFH 类型的值。通常为包含要搜索的文本文档的句柄的句柄列。也可以是从 TEXTCOLUMNS 目录视图中抽取的原型句柄。

search-argument

包含要搜索的词条、类型为 LONG VARCHAR 的字符串。参见第201页的『第12章 搜索变元的语法』。

INIT_TEXT_HANDLE

INIT_TEXT_HANDLE 函数返回一个部分初始化的句柄，该句柄包含文本的格式或语言的预设置值。可将它插入到句柄列中。当您添加一行，而该行包含格式和语言与在文本配置设置中所指定的格式和语言不同的文本时，这是有用的。

返回的句柄是 DB2TEXTH 类型的值。

若打算在存储在外部文件的文本中而不是在 DB2 UDB 表中搜索，则可以使用 INIT_TEXT_HANDLE 函数来返回一个完全初始化的句柄，该句柄包含文本的 CCSID、格式、语言以及该文件位置的预设置值。

返回的句柄是 DB2TEXTFH 类型的值。

使用 INIT_TEXT_HANDLE 函数插入或更新句柄值。

函数语法

►►—INIT_TEXT_HANDLE—(—*format*—,—*language*—)—————►►

►►—INIT_TEXT_HANDLE—(—*CCSID*—,—*format*—,—*language*—,—*file-name*—)—————►►

函数参数

format

一个类型为 VARCHAR(30) 的字符串，指定新文档格式设置。受支持的格式列示在第32页的『受支持的文档格式』中。

language

一个类型为 VARCHAR(30) 的字符串，指定新文档语言设置。受支持的语言列示在第53页的表5中。

file-name

一个 VARCHAR(150) 类型的字符串，指定将要与句柄相关联的外部文件的绝对路径和文件名。要具有对 UNIX 中的文件的存取权，DB2 UDB 实例所有者必须包括在文件存取许可权中。对于 OS/2 和 Windows 用户，文件存取许可权必须包括注册用户 ID。

LANGUAGE

LANGUAGE 函数执行下列其中之一：

- 返回在句柄中指定的文档语言
- 更改文档的句柄中的语言规范，并返回更改了的句柄。

返回的文档语言是 VARCHAR(30) 类型的字符串。返回的句柄是 DB2TEXTFH 或 DB2TEXTH 类型。

这是用于索引相应的文本文档的语言参数。 受支持的语言列示在第53页的表5中

函数语法

►► LANGUAGE ⁽¹⁾ (—*handle*—) ►►

注：

- 1 返回语言值，类型为 VARCHAR(30)。

►► LANGUAGE ⁽¹⁾ (—*handle*—, —*language*—) ►►

注：

- 1 返回一个句柄，类型为 -DB2TEXTFH 或 DB2TEXTH。

函数参数

handle

一个表达式，其结果是 DB2TEXTFH 或 DB2TEXTH 类型的值。通常为要从中返回或设置语言设置的句柄列的名称。

language

数据类型 VARCHAR(30) 的新文档语言设置。

若指定了 *language*，则在句柄中设置此文档语言；返回句柄而不返回语言设置。

NO_OF_DOCUMENTS

NO_OF_DOCUMENTS 函数返回一个 INTEGER 值，指示在搜索所找到的文本文档的列表中词条的个数。返回的值是在句柄列表中找到的项的个数。

提示

此功能是较早版本的 DB2 Text Extender 的功能部件由于兼容性原因，这些功能仍然受支持。但 SEARCH_RESULT 搜索功能已替代了其功能。

函数语法

►►—NO_OF_DOCUMENTS—(—*handle-list*—)—————►

函数参数

handle-list

一个表达式，其结果是 DB2TEXTHLISTP 或 DB2TEXTFHLISTP 类型的值。它由 HANDLE_LIST 函数返回。

这是一个指向搜索所找到的文档的句柄列表的指针。

HANDLE_LIST 和 NO_OF_DOCUMENTS 函数必须在同一个 SQL 语句中，这是因为列表仅存在于该语句的范围内。

NO_OF_MATCHES function

NO_OF_MATCHES

NO_OF_MATCHES 可以在文本文档中搜索，并返回一个 INTEGER 值，指示每个文档产生了多少个匹配项。

函数语法

►►NO_OF_MATCHES(—*handle*—,—*search-argument*—)◄◄

函数参数

handle

一个表达式，其结果是 DB2TEXTFH 或 DB2TEXTH 类型的值。通常为包含要搜索的文本文档的句柄的句柄列的名称。

search-argument

包含要搜索的词条、类型为 LONG VARCHAR 的字符串。参见第201页的『第12章 搜索变元的语法』。

RANK

RANK 可以在文本文档中搜索并对每个找到的文档返回一个等级值，指示搜索变元描述该找到的文档的程度。

RANK 返回 0 和 1 之间的 DOUBLE 值。等级值是绝对的，指示与其他找到的文档相比该找到的文档满足搜索条件的程度。该值指示在文档中找到的、与文档的大小相关的匹配数。

函数语法

►► RANK (—*handle*—, —*search-argument*—) ◀◀

函数参数

handle

一个表达式，其结果是 DB2TEXTFH 或 DB2TEXTH 类型的值。通常为包含要搜索的文本文档的句柄的句柄列的名称。

search-argument

包含要搜索的词条、类型为 LONG VARCHAR 的字符串。参见第201页的『第12章 搜索变元的语法』。

REFINE

REFINE 函数取两个搜索变元并返回一个类型为 LONG VARCHAR 的组合搜索变元，用布尔运算符 AND 将两个原始搜索变元连接起来。

函数语法

►►REFINE(—*search-argument*—,—*search-argument*—)◄◄

函数参数

search-argument

包含要搜索的词条、类型为 LONG VARCHAR 的字符串。参见第201页的『第12章 搜索变元的语法』。

搜索变元一定不能包含搜索参数 IS ABOUT、THESAURUS 或 EXPAND。

SEARCH_RESULT

SEARCH_RESULT 函数返回在中间表中搜索的结果。此函数可用于 SQL 语句的 FROM 子句中。

返回的表的结构如下：

Column Name	Data Type
HANDLE	DB2TX.DB2TEXTH, DB2TX.DB2TEXTFH
NUMBER_OF_MATCHES	INTEGER
RANK	DOUBLE

仅对中间表的选择的列生成值。选择计数 (*) 仅生成 HANDLE 列。因为等级值的计算消耗大量系统资源，所以若等级值不是必需的，则不应从中间表中选择等级值。

当处理大型表时，此函数比 CONTAINS 或 RANK 运行得要快。

函数语法

►—SEARCH_RESULT—(—schema—,—table—,—handle—,—search-argument—)————►

函数参数

schema

您正在其中进行搜索的表的模式名。

table 您正在其中进行搜索的表的名称。

句柄 对应于包含要搜索的文档的句柄列的名称。

search-argument

包含要搜索的词条、类型为 LONG VARCHAR 的字符串。参见第201页的『第12章 搜索变元的语法』。

示例

有关示例，参考第92页的『提高搜索性能』或查看第73页的『何处可找到搜索函数的语法示例』中提及的文件中所述的样本 DB2 Text Extender 函数。

SEARCH_RESULT 函数

第12章 搜索变元的语法

搜索变元是指在文本文档中搜索词条时所指定的条件。它包括一个或几个搜索词条和搜索参数。

在第81页的『指定搜索变元』中和名为 `txsample.udf` 的文件中提供了搜索变元的示例。它包含 `DB2 Text Extender` 函数的示例，这些函数是对第16页的『准备样本数据库以进行安装验证』中所述的样本表运行的。

使用搜索变元的 `DB2 Text Extender` 函数是：

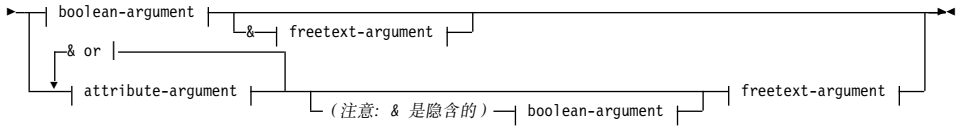
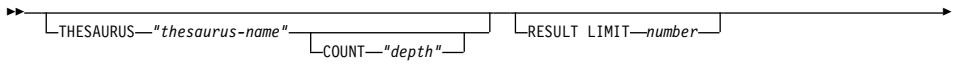
- `CONTAINS`。此函数使用搜索变元在特定文本文档中搜索文本。若文档包含该文本，它返回 `INTEGER` 值 1。否则，返回 0。
- `NO_OF_MATCHES`。此函数使用搜索变元在文本文档中搜索。它并返回一个 `INTEGER` 值，指示每个文档产生了多少个匹配。
- `RANK`。此函数使用搜索变元在文本文档中搜索。它为每个找到的文档返回一个值，指示搜索变元描述该找到的文档的程度。
- `REFINE`。此函数取两个搜索变元并返回一个类型为 `LONG VARCHAR` 的组合搜索变元，用布尔运算符 `AND` 将两个原始搜索变元连接起来。
- `HANDLE_LIST`。此函数使用搜索变元在文本文档中搜索。它返回 `DB2TEXTHLISTP` 或 `DB2TEXTFHLISTP` 类型的值，指向找到的文档的句柄列表。
- `SEARCH_RESULT`。此函数返回一个包含请求的信息（即，等级、匹配数以及句柄）的表。

使用搜索变元的 `API` 函数有：

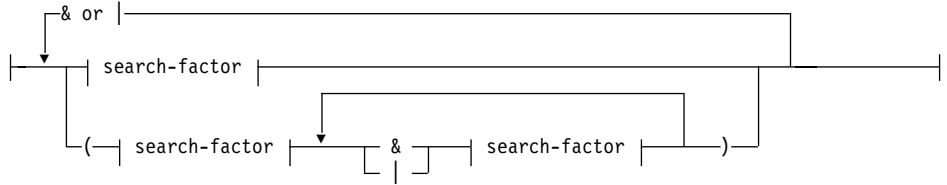
- `DesGetBrowseInfo`。此函数使用搜索变元以在由句柄标识的文本中搜索。它返回一个指向浏览信息的指针，该浏览信息是 `DesStartBrowseSession` 为突出显示词条所需的。
- `DesGetSearchResultTable`。此函数使用搜索变元以在由文本列标识的文本文档中搜索。找到的文本项的句柄数据写至结果表中。关于等级和匹配数的浏览信息也可写至结果表中。

搜索变元

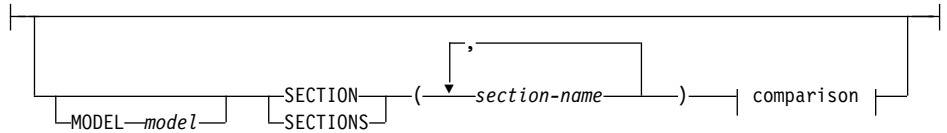
搜索变元语法



boolean-argument:



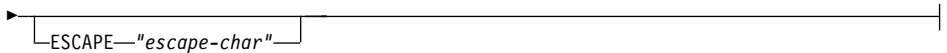
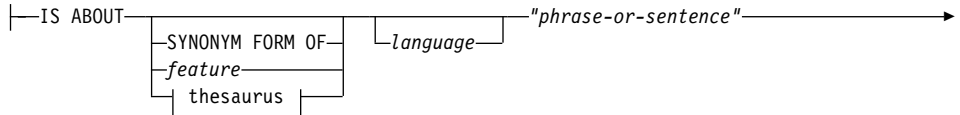
attribute-argument:



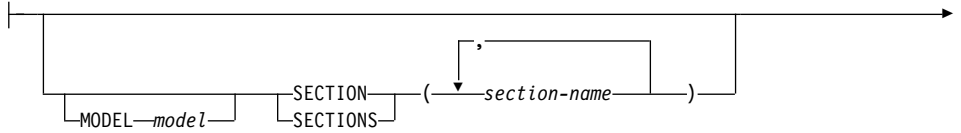
Comparison:



freetext-argument:

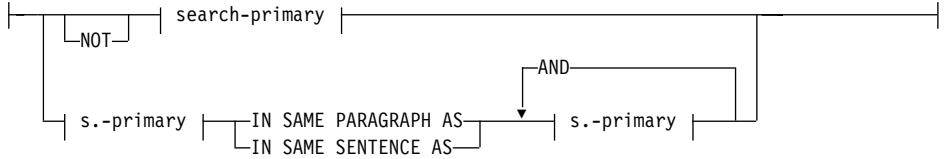


search-factor:

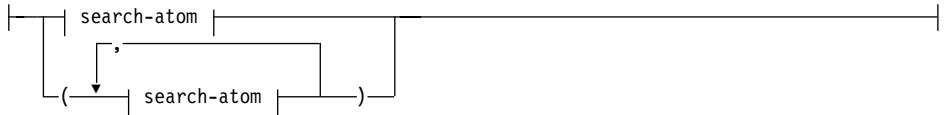


▶ **search-element**

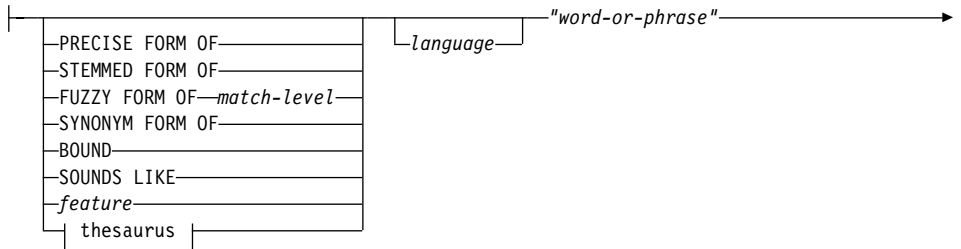
search-element:



search-primary:

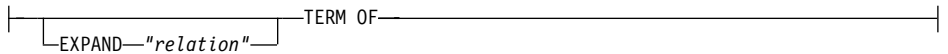


search-atom:



▶ **ESCAPE**—*"escape-character"*

thesaurus (if THESAURUS is specified):



搜索变元的语法

示例

在第81页的『指定搜索变元』中提供了示例。

搜索参数

IS ABOUT

一个让您指定自由文本搜索变元的选项，该自由文本搜索变元也就是指描述要查找的概念的自然语言短语或句子。参见第88页的『自由文本和混合搜索』。

MODEL *model*

用来指定要在搜索词条中使用的文档模型的名称。文档模型描述包含可标识节的文档的结构，以便可以个别地搜索这些节的内容。

模型名必须在第47页的『使用结构文档（部分支持）』中描述的文档模型文件中指定。可以使用通配符将模型名屏蔽起来。

若不指定模型，则使用创建索引期间指定的缺省模型。

SECTION(S) *section-name*

用来指定将只对其执行搜索的一节或多节的关键字。必须在文档模型文件中的模型中指定节名，如第47页的『使用结构文档（部分支持）』中所述。可以使用通配符 % 和 _ 将节名屏蔽起来。

一些节可以嵌套在其他节中，例如：

```
play/Act/Title=play/act/title
```

限制： 只有对存储在按 XML 格式启用的列中的文档，才有可能在嵌套的节中进行搜索。对于 Ngram 索引，只能搜索一个节名，并且不支持 XML 格式。

<*attribute-value*

在先前的节列表中列示了与属性的先前比较运算符一起使用的值。对于请求要在某个范围内的属性值的查询可以在一个属性条件中使用两个比较运算符。

使用同一条件的同一类比较运算符组合（例如，在第一个比较中使用 \geq ，在第二个比较中使用 $>$ ）是无效的。带有 = 运算符的二个比较的规范也是无效的。

= 请求将索引文档中的属性与下列属性值进行相等比较。

\geq 请求将索引文档中的属性与下列属性值进行“大于或等于”比较。

- > 请求将索引文档中的属性与下列属性值进行“大于”比较。
- ≤ 请求将索引文档中的属性与下列属性值进行“小于或等于”比较。
- < 请求将索引文档中的属性与下列属性值进行“小于”比较。

某些部分可以嵌套在其他部分之中，例如：

```
play/Act/Title=play/act/title
```

限制： 只有对存储在按 XML 格式启用的列中的文档，才有可能在嵌套的节中进行搜索。对于 Ngram 索引，只能搜索一个节名，并且不支持 XML 格式。

THESAURUS *thesaurus-name*

用来指定要用于扩充搜索词条的同义词典的名称的关键字。同义词典名是已经使用同义词典编译器 TXTHESC 或 TXTHESN 编译过的同义词典的文件名（不带有扩展名）。样本目录中存储有缺省同义词典 `desthes` 和 `desnthes`，其中，`desnthes` 是 Ngram 同义词典。还可以指定该文件的路径名。缺省路径名是词典路径。

COUNT *depth*

用于指定同义词典中词条的级别数（深度）的关键字，这些级别用于扩展给定关系的搜索词条。若不指定此关键字，则假定计数为 1。

RESULT LIMIT *number*

用来指定在结果列表中要返回的最大项数的关键字。数目是从 1 到 32767 的值。若使用自由文本搜索，则只考虑完整的搜索结果列表来对搜索结果列表进行等级评定。否则，仅根据列表项对该限制的搜索结果进行等级评定。

EXPAND *relation*

用来指定在 TERM OF 中指定的搜索词条和要用来扩充搜索词条的同义词典词条之间的关系（如 INSTANCE）的关键字。关系名必须与同义词典中所使用的关系对应。参见第246页的『同义词概念』。

对于 Ngram 同义词典，使用第254页的『创建 Ngram 同义词典』中所述的成员关系名。对于用户定义的成员关系，使用 `:RELATION n`，其中，*n* 是在 `:RELATED (number)` 中指定的成员关系号。

TERM OF *"word-or-phrase"*

要将其他搜索词条从同义词典添加至其中的搜索词条或多字搜索词条。

搜索变元的语法

search-factor

一个操作数，它可以与其他操作数组合以形成一个搜索变元。求值顺序是从左到右。逻辑 AND (&) 运算符优先于逻辑 OR (|) 运算符。示例：

```
"passenger" & "vehicle" | "transport" & "public"
```

计算为：

```
("passenger" & "vehicle") | ("transport" & "public")
```

要搜索：

```
"passenger" & ("vehicle" | "transport") & "public"
```

必须包括圆括号（如所显示的那样）。

NOT search-primary

让您将包含特定词条的文本文档排除在搜索之外的运算符。

当在搜索因子中使用 NOT 时，不能使用 SYNONYM FORM OF 关键字。

search-primary IN SAME PARAGRAPH AS search-primary

让您搜索出现于同一段落中的词条组合的关键字。

仅当词条“air”在同一段落中时，以下搜索变元才查找包含词条“traffic”的文本文档。

```
"traffic" IN SAME PARAGRAPH AS "air"
```

当在搜索因子中使用 NOT 时，不能使用 IN SAME PARAGRAPH AS 关键字。

search-primary IN SAME SENTENCE AS search-primary

让您搜索出现于同一句子中的词条组合的关键字。类似于 IN SAME PARAGRAPH AS。

AND search-primary

让您组合几个在同一句子或同一段落中要搜索的搜索主词条的关键字。

以下搜索变元在同一句子中搜索“forest”、“rain”、“erosion”以及“land”。

```
"forest" IN SAME SENTENCE AS "rain" AND "erosion" AND "land"
```

search-atom

若用逗号连接一系列搜索微元，则若在任何这些搜索微元中找到了词条，搜索就成功。每个搜索微元必须至少包含一个字或一个短语。

若找到这些搜索变元中的一个或多个，则以下语句为真。

```
CONTAINS (mytexthandle, '( "text",
                        "graphic",
                        "audio",
                        "video"))' = 1
```

PRECISE FORM OF, STEMMED FORM OF, FUZZY FORM OF, SYNONYM FORM OF, BOUND

表8显示对应于各种索引类型的选项。例如，对于语言索引，除 PRECISE FORM OF 之外的任何选项都适用。若指定 PRECISE FORM OF，则忽略它并取缺省值。

在表9中有对搜索词条处理的更详细描述。

表 8. 语言选项

搜索原子关键字	索引类型				
	语言	精确	精确规格化	Ngram	Ngram 且区分大小写
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=缺省设置 O=可用函数

表 9. Ngram 索引的搜索词条选项

搜索原子关键字	搜索词条处理				
	大小写		词干	匹配	
	区分	不区分		精确	模糊
PRECISE FORM OF	当启用可大小写时	X		X	
STEMMED FORM OF		X	X		
FUZZY FORM OF		X			X

X=缺省设置

若使用不可用于该索引类型的关键字，则忽略它，并且或者使用缺省关键字作为替代，或者返回信息。

PRECISE FORM OF

一个关键字，它导致完全按输入格式搜索跟在 **PRECISE FORM OF** 后面的那个字（或短语中的每个字），而不是先将该字缩至其词干形式。对于精确索引，这种搜索是区分大小写的；也就是说，使用大写和小写字符是有区别的。例如，若搜索 `mouse`，您找不到 `“Mouse”`。

对于精确索引来说，这是缺省选项。对于“精确规格化”索引，缺省搜索形式是不区分大小写。若对语言索引指定此关键字，则忽略它并假定 **STEMMED FORM OF**。

STEMMED FORM OF

一个关键字，它导致在执行搜索之前将跟在 **STEMMED FORM OF** 后面的那个字（或短语中的每个字）缩至其词干形式。这种搜索不区分大小写。例如，若搜索 `mouse`，就可以找到 `“Mouse”`。

将字缩至其词干形式的方法是与语言有关的。

示例：当使用“美国英语”词典时，用 `program compute system` 代替 `programming computer systems`，而当使用“英国英语”词典时，则用 `programme compute system` 代替。

此搜索短语能够查找 `“programmer computes system”`、`“program computing systems”`、`“programming computer system”`，等等。

对于语言索引来说，这是缺省选项。若对精确索引指定此关键字，则忽略它并假定 **PRECISE FORM OF**。

FUZZY FORM OF

一个执行“模糊”搜索的关键字，模糊搜索是指搜索与搜索词条拼写类似的词条。在由“光学字符识别”（OCR）程序创建的文档中搜索时特别有用。这类文档经常包括拼写错误的字。例如，OCR 程序可能会将字 `economy` 识别为 `economy`。

match-level: 从 1 到 5 的整数，指定类似的程度，其中 5 比 1 的类似程度更高。

SYNONYM FORM OF

一个关键字，它导致将跟在 **SYNONYM FORM OF** 后面的字或短语与其同义词一起搜索。由语言指定的词典或其他缺省词典提供这些同义词。

短语的同义词是指包含所有可能同义词的组的替代短语，可通过用原始短语的每个字的同义词之一来替换该字来获得。字顺序保持为在原始短语中的那样。

若对精确索引指定此关键字，则忽略它并假定 **PRECISE FORM OF**。

当在搜索因子中使用 **NOT**，或当要搜索的字或短语包含屏蔽字符时，不能指定此关键字。

BOUND

用于在使用韩文 **CCSID** 的文档中搜索的关键字。它导致搜索考虑字短语边界。若指定了语言，则忽略它；并假定韩文。

language

一个变量，它确定在索引和检索期间，在文本文档的语言处理中使用哪个词典。这不仅适用于语言索引，还适用于精确索引，因为这些索引使用词典来处理停止字。

语言处理包括同义词处理和字干处理。

受支持的语言列示在第53页的表5中。

注：当在不是美国英语的文档中搜索时，必须在搜索变元中指定语言而不考虑缺省语言。

"word-or-phrase"

要搜索的字或短语。可在字中使用的字符是与语言有关的。字是否需要以分隔符来分隔也是与语言有关的。对于英语和大多数其他语言，短语中的每个字必须以一个空格字符来分隔。

精确或语言搜索。DB2 Text Extender 既可以使用字或短语的精确形式搜索，也可以使用其变体进行搜索。若不在第207页的表8中指定一个选项，则根据所使用的索引类型来使用缺省语言选项。

要搜索包含了双引号的字符串，输入双引号两次。例如，要搜索文本 "wildcard" character，使用：

```
""wildcard"" character"
```

屏蔽字符。字可以包含下列屏蔽字符：

_（下划线）

代表任何单个字符。

%（百分号）

代表任何数目的任意字符。若一个字由单个 % 组成，则它代表一个任意长度的可选字。

搜索变元的语法

除使用单个 % 来代表一个可选字之外，一个字不能仅由屏蔽字符组成。

若使用屏蔽字符，则不能使用 SYNONYM FORM OF、功能部件或 THESAURUS。

ESCAPE *escape-character*

一个字符，它将下一个字符标识为要搜索的一个字符，而不是作为要用作屏蔽字符的字符。

示例：如果 *escape-character* 是 \$， \$%、\$_ 和 \$\$ 分别代表 %、_ 和 \$。不以 \$ 为前缀的任何 % 和 _ 字符代表屏蔽字符。

规则和限制摘要：

布尔运算

OR 后不允许 NOT。

FUZZY FORM OF

前 3 个字符必须匹配。若搜索微元中的字包含屏蔽字符，则不能使用。不能与 NOT 组合使用。只能与 Ngram 索引一起使用。

IN SAME PARAGRAPH AS

若在搜索因子中使用 NOT，则不能使用。

IN SAME SENTENCE AS

若在搜索因子中使用 NOT，则不能使用。

语言索引

防止使用 PRECISE FORM OF。将 STEMMED FORM OF 用作缺省值。可以使用屏蔽字符。搜索是区分大小写的。

屏蔽字符

防止使用 SYNONYM FORM OF 和 THESAURUS。

Ngram 索引

可以使用屏蔽字符，尽管未跟在非字母数字字符后面。除非索引启用了大小写并且使用了 PRECISE FORM OF，否则搜索不区分大小写。

NOT 防止使用 SYNONYM FORM OF、IN SAME PARAGRAPH AS 和 IN SAME SENTENCE AS。

PRECISE FORM OF

对于语言索引，忽略它。

精确索引

防止使用 STEMMED FORM OF 和 SYNONYM FORM OF。将 PRECISE FORM OF 用作缺省值。可以使用屏蔽字符。搜索区分大小写。

STEMMED FORM OF

对于精确索引，将被忽略，但是可用于包含“英语”文档的正规精确索引。

SYNONYM FORM OF

若搜索微元中的字包含屏蔽字符，则不能使用。不能与 NOT 组合使用。不能随精确索引一起使用。

第13章 用于搜索和浏览的 API 函数

DB2 Text Extender 提供 C 函数以搜索文本文档和浏览（显示）找到的文档 这些函数组成 DB2 Text Extender 应用程序设计接口 (API) 本章按字母次序描述这些 API 函数。

第111页的『第8章 使用 API 函数进行搜索和浏览』提供对这些函数的介绍，并描述如何可将它们一起使用。

函数	目的	页
DesCloseDocument	释放由 DesOpenDocument 分配的存储器。	214
DesEndBrowseSession	关闭浏览对话并释放由 DesStartBrowseSession 分配的存储器。	215
DesFreeBrowseInfo	释放由 DesGetBrowseInfo 分配的存储器。	216
DesGetBrowseInfo	使用搜索变元在文档中搜索文本并创建浏览信息。	217
DesGetMatches	返回指向由文档句柄描述的文本文档的突出显示信息的指针。突出显示的信息是数据流。它包括文本上下文（至少一个段落）和该上下文中的突出显示文本的信息。	220
DesGetSearchResultTable	采用搜索变元以在给定的文本列中搜索文本文档，并将结果存储在用户提供的表中。它还可以返回浏览信息。	225
DesOpenDocument	接收到浏览对话指针、句柄，以及选项 DES_FAST 或 DES_EXTENDED，该选项指示要用于突出显示找到的词条的语言处理的类型。准备与句柄相应的文本文档以获取文档文本和突出显示的信息，并返回用于重复调用 DesGetMatches 的文档句柄。	229
DesStartBrowseSession	使用来自 DesGetBrowseInfo 的浏览信息打开浏览对话，并且返回供其他浏览函数使用的浏览对话句柄。	231

提示

许多 API 函数需要连接句柄 (hdbc)。必须使用 SQLConnect 函数提供此句柄，但这并不妨碍您从嵌入式 SQL 程序调用 DB2 Text Extender。 *DB2 Call Level Interface Guide and Reference* 描述如何将 CLI 语句与嵌入式 SQL 语句混合。

DesCloseDocument API 函数

DesCloseDocument

目的

关闭由 DesOpenDocument 打开的文本档，并在返回文档文本和突出显示的信息期间释放已分配的存储器。

语法

```
DESRETURN  
DesCloseDocument  
    (DESBROWSESESSION BrowseSession,  
     DESHANDLE DocumentHandle);
```

函数自变量

表 10. DesCloseDocument 自变量

数据类型	自变量	使用	说明
DESBROWSESESSION	<i>BrowseSession</i>	输入	浏览对话句柄。
DESHANDLE	<i>DocumentHandle</i>	输入	标识打开的文本档的 DesOpenDocument 返回的句柄。

返回码

```
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
```

限制

仅在通过调用 DesOpenDocument 打开了文本档之后，才可调用此函数。

DesEndBrowseSession

目的

结束由 DesStartBrowseSession 启动的浏览对话并释放分配给该浏览对话的存储器。

语法

```
DESRETURN
DesEndBrowseSession
(DES_BROWSE_SESSION BrowseSession);
```

函数自变量

表 11. DesEndBrowseSession 自变量

数据类型	自变量	使用	说明
DES_BROWSE_SESSION	<i>BrowseSession</i>	输入	浏览对话句柄。

用法

此函数不释放由 DesGetBrowseInfo 分配给浏览对话的存储器。此存储器包含仍可用于另一浏览对话的浏览信息。要释放此存储器，调用 DesFreeBrowseInfo。

返回码

```
RC_SUCCESS
RC_INVALID_SESSION
RC_INVALID_PARAMETER
RC_SE_UNEXPECTED_ERROR
```

限制

仅在通过调用 DesStartBrowseSession 启动了浏览对话之后，才可调用此函数。

DesFreeBrowseInfo

目的

释放由 DesGetBrowseInfo 分配给浏览信息的存储器。

语法

```
DESRETURN  
DesFreeBrowseInfo  
(DESBROWSEINFO BrowseInfo );
```

函数自变量

表 12. DesFreeBrowseInfo 自变量

数据类型	自变量	使用	说明
DESBROWSEINFO	<i>BrowseInfo</i>	输入	浏览信息

返回码

RC_SUCCESS

RC_INVALID_PARAMETER

限制

仅在通过调用 DesGetBrowseInfo 给浏览信息分配了存储器之后，才可调用此函数。

DesGetBrowseInfo

目的

接收搜索变元以搜索由句柄标识的文本。它返回一个指向浏览信息的指针，该信息是 DesStartBrowseSession 需要的，用于突出显示找到的词条。

语法

```

DESRETURN
DesGetBrowseInfo
(SQLHDBC          hdbc,
SQLCHAR          *pHandle,
DESUSHORT        HandleLength,
char             *pSearchArgument,
DESSMALLINT      ArgumentLength,
DESBROWSEINFO   *pBrowseInfo,
DESMESSAGE       *pErrorMessage);
  
```

函数自变量

表 13. DesGetBrowseInfo 自变量

数据类型	自变量	使用	说明
SQLHDBC	<i>hdbc</i>	输入	数据库连接句柄。
SQLCHAR *	<i>pHandle</i>	输入	指向已从数据库抽取的句柄的指针。
DESUSHORT	<i>HandleLength</i>	输入	<i>pHandle</i> 的长度。此处不能使用 DES_NTS。
char *	<i>pSearchArgument</i>	输入	指向指定想要查找的信息的文本搜索变元的指针。
DESSMALLINT	<i>ArgumentLength</i>	输入	<i>pSearchArgument</i> 的长度（不包括空字节结束符）或 DES_NTS。
DESBROWSEINFO *	<i>pBrowseInfo</i>	输出	指向包含浏览文本文档所需数据的浏览信息的指针。此指针被传送给 DesStartBrowseSession。
DESMESSAGE *	<i>pErrorMessage</i>	输出	由实施方案定义的信息文本。如果出现错误，则 DB2 Text Extender 返回错误码和错误信息。应用程序分配大小为 DES_MAX_MESSAGE_LENGTH 的缓冲区。如果 <i>pErrorMessage</i> 是空指针，则不返回任何错误信息。

用法

应用程序必须在调用 DesGetBrowseInfo 之前创建与数据库的连接。

DesGetBrowseInfo API 函数

对于指向搜索变元的指针，使用 *char** 而不是使用 *SQLCHAR**。这是因为参数值可能不是来自于数据库的。

对于 SQL 数据类型和 C 数据类型之间的映像，必须对句柄使用 SQL 符号名 *SQL_VARBINARY*。指向 *Handle* 值的 C 表达式的宿主变量类型是 *SQLCHAR**。

DB2 Text Extender 将存储器分配给浏览信息。应用程序必须通过调用 *DesFreeBrowseInfo* 释放存储器和相关资源。

因为 *Handle* 值是位数据且包含若干个 '\0' 字符，所以必须指定 *pHandle* 的长度。

pSearchArgument 中的搜索变元在第201页的『第12章 搜索变元的语法』中作了描述。

返回码

RC_SUCCESS
RC_NO_BROWSE_INFO

RC_ALLOCATION_ERROR
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

警告：下列返回码指示函数已返回结果，但它可能不是所期望的结果。

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

限制

仅当已经创建了至数据库的连接并且使用了 DB2 Text Extender 函数从该数据库抽取句柄之后，才可调用此函数。

DesGetMatches

目的

返回包含由文档句柄描述的文本文档的突出显示信息的数据流。参见『数据流语法』。突出显示信息包括文本上下文（至少一个段落）和该上下文中的突出显示文本的信息。

DesGetMatches 仅返回数据流的一部分，在输出结构中指示该部分的长度。

一个调用 DesGetMatches 的序列将获取整个文本文档内容。当到达文本文档的末尾时，返回 RC_SE_END_OF_INFORMATION。

语法

```

DESRRETURN
DesGetMatches
    (DESBROWSESESSION BrowseSession,
    DESHANDLE DocumentHandle,
    DESMATCHINFO *pMatchInfo,
    DESULONG *pMatchInfoLength,
    DESMESSAGE *pErrorMessage);
    
```

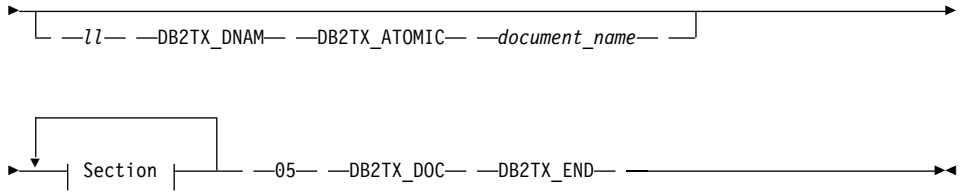
函数自变量

表 14. DesGetMatches 自变量

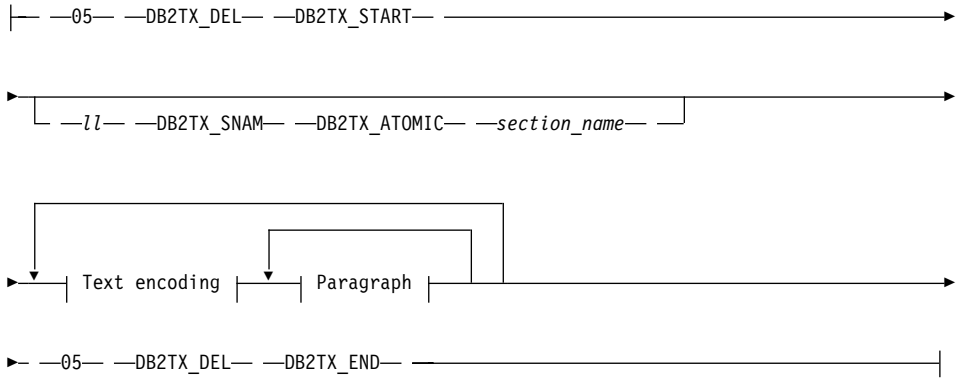
数据类型	自变量	使用	说明
DESBROWSESESSION	<i>BrowseSession</i>	输入	浏览对话句柄。
DESHANDLE	<i>DocumentHandle</i>	输入	由 DesOpenDocument 返回的文档句柄。
DESMATCHINFO *	<i>pMatchInfo</i>	输出	指向包含接收到的数据流部分的缓冲区的指针。DesGetMatches 分配该缓冲区。
DESULONG *	<i>pMatchInfoLength</i>	输出	pMatchInfo 所指向的数据流部分的长度。
DESMESSAGE *	<i>pErrorMessage</i>	输出	由实施方案定义的信息文本。如果出现错误，则 DB2 Text Extender 返回错误码和错误信息 应用程序分配大小为 DES_MAX_MESSAGE_LENGTH 的缓冲区。如果 pErrorMessage 是空指针，则不返回任何错误信息。

数据流语法

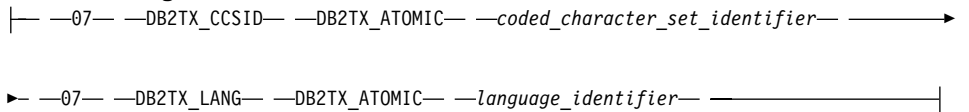
▶ —05— —DB2TX_DOC— —DB2TX_START— →



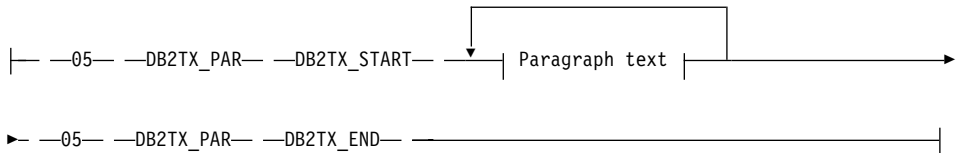
Section:



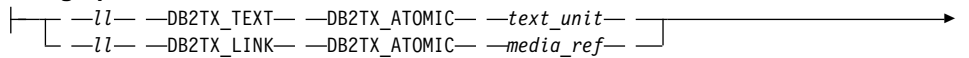
Text encoding:



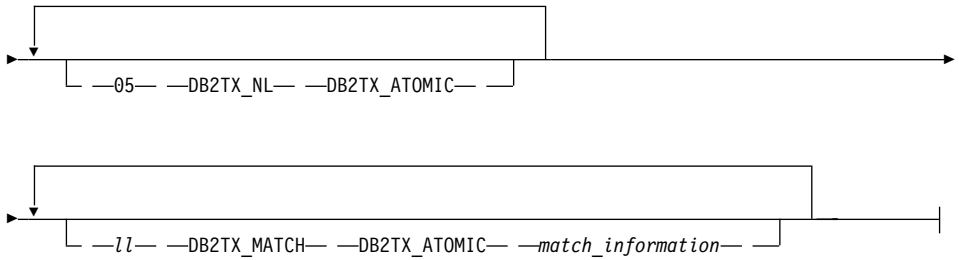
Paragraph:



Paragraph text:



DesGetMatches API 函数



语法图中的每个段（如 05 DB2TX_DOC DB2TX_START）都以一个整型长度字段开始，它在图中是一个明确的数字（如 05），或者是一个变量（如 11）。数据段的长度包括两个字节的长度字段。

注：该长度的格式是大尾结构格式。

每个段包括下列其中一个 1 字节的类型标识符：

DB2TX_START

指示段的开头，如文档或段落。

DB2TX_END

指示段的结尾。

DB2TX_ATOMIC

指示跟在后面的项是原子，如文档名或语言标识符。

每个数据流项的长度是两个字节的。它们是：

DB2TX_DOC

指示文档的开头和结尾。

DB2TX_DNAM

文档名。若未指定名称，则使用文档的标识符。

DB2TX_DEL

指示文档元素的开头和结尾。当前受支持的文档元素的类型是文本节。

DB2TX_SNAM

指定文本节的名称。目前，DB2 Text Extender 只支持一个文本节并且自动提供缺省名。若您指定节名，则该缺省名被忽略。

DB2TX_PAR

指示当前节内的文本段落的开头和结尾。

DB2TX_TEXT

指定当前段落内的一个文本部分。通常，文本单元包含一行文本，并且 DB2TX_NL 项跟在 TEXT 项的后面；但文本行还可能被分成若干部分，每个部分在其自己的 DB2TX_TEXT 项中指定。

该文本使用与当前段落相关的 CCSID 和语言。

DB2TX_LINK

指定 DB2 Text Extender 超媒体引用 它使用当前段落的 CCSID。

DB2TX_NL

指示当前段落中新行的开头。

DB2TX_MATCH

包含当前文本部分中匹配的发生信息。以二进制数对序列的形式提供信息。每对中的第一个数字是当前文本部分内匹配的位移，第二个数字是该匹配的长度（以字符计）。给定的长度可能超过指定的文本部分。位移和长度都是以大尾结构格式指定的两个字节的值。

DB2TX_CCSID

后续段落中文本的 CCSID ，直到段落前出现新的 DB2TX_CCSID 项为止。返回下列 CCSID:

DB2TX_CCSID_00500

用于 Latin-1 EBCDIC 代码页 500 中的文本。

DB2TX_CCSID_04946

用于 Latin-1 ASCII 代码页 850 中的文本。

DB2TX_CCSID_00819

用于 ASCII 代码页 819 中的文本。

这些 CCSID 的符号名是在随 DB2 Text Extender 一起提供的文件 DES_EXT.H 中定义的。该两字节二进制值是以大尾结构格式指定的。

DB2TX_LANG

后续段落中文本的语言标识符，直到段落前出现新的 DB2TX_LANG 项为止。随 DB2 Text Extender 一起提供的文件 DES_EXT.H 为受 DB2 Text Extender 支持的所有语言标识符定义符号名 该两字节二进制值是以大尾结构格式指定的。

用法

当到达文本文档的末尾时，DesGetMatches 返回 RC_SE_END_OF_INFORMATION。

DesGetMatches API 函数

返回码

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

警告：下列返回码指示函数已返回结果，但它可能不是所期望的结果。

RC_SE_DICTIONARY_NOT_FOUND

限制

仅在通过调用 DesOpenDocument 打开了文本文档之后，才可调用此函数。

DesGetSearchResultTable

目的

使用搜索变元以搜索由文本列标识的文本文档。找到的文本项的句柄数据写至结果表中。关于等级和匹配数的浏览信息也可写至结果表中。

语法

```

DESRETURN
DesGetSearchResultTable
(SQLHDBC          hdbc,
char              *pTableSchema,
DESSMALLINT      TableSchemaLength,
char              *pTableName,
DESSMALLINT      TableNameLength,
char              *pColumnName,
DESSMALLINT      ColumnNameLength,
char              *pSearchArgument,
DESSMALLINT      ArgumentLength,
char              *pResultSchema,
DESSMALLINT      ResultSchemaLength,
char              *pResultTableName,
DESSMALLINT      ResultTableNameLength,
DESSEARCHOPTION  SearchOption,
DESBROWSEOPTION  BrowseOption,
DESBROWSEINFO    *pBrowseInfo,
DESMESSAGE       *pErrorMessage);

```

函数自变量

表 15. *DesGetSearchResultTable* 自变量

数据类型	自变量	使用	说明
SQLHDBC	hdbc	输入	数据库连接句柄。
char *	pTableSchema	输入	要搜索的基本表的模式。
DESSMALLINT	TableSchemaLength	输入	pTableSchema 的长度（不包括空字节结束符）或 DES_NTS。
char *	pTableName	输入	指向要搜索的基本表的名称的指针。
DESSMALLINT	TableNameLength	输入	pTableName 的长度（不包括空字节结束符）或 DES_NTS。
char *	pColumnName	输入	指向要由期望的文本搜索寻址的列的名称的指针。该列的类型必须是 DESTEXTH。
DESSMALLINT	ColumnNameLength	输入	pColumnName 的长度（不包括空字节结束符）或 DES_NTS。

DesGetSearchResultTable API 函数

表 15. DesGetSearchResultTable 自变量 (续)

数据类型	自变量	使用	说明
char *	pSearchArgument	输入	指向文本搜索变元的指针。
DESSMALLINT	ArgumentLength	输入	pSearchArgument 的长度 (不包括空字节结束符) 或 DES_NTS。
char *	pResultSchema	输入	指向包含结果表的模式的指针。
DESSMALLINT	ResultSchemaLength	输入	pSchemaName 的长度 (不包括空字节结束符) 或 DES_NTS。
char *	pResultTableName	输入	指向结果表的名称的指针, 该结果表是先前创建的、要在其中存储搜索结果的表。有关此表的结构, 参见第227页的图17。
DESSMALLINT	ResultTableNameLength	输入	pResultTableName 的长度 (不包括空字节结束符) 或 DES_NTS。
DESSEARCHOPTION	SearchOption	输入	<p>一个选项, 它确定您请求等级信息、匹配数, 还是仅请求匹配文本文档的句柄。</p> <p>DES_RANK DES_MATCH DES_RANKANDMATCH DES_TEXTHANDLEONLY</p> <p>此选项确定结果表的内容, 如『用法』中所述。</p>
DESBROWSEOPTION	BrowseOption	输入	保留。
DESBROWSEINFO *	pBrowseInfo	输出	指向浏览信息的指针或是空指针, 这视 BrowseOption 的值而定。
DESMESSAGE *	pErrorMessage	输出	<p>由实施方案定义的信息文本。如果出现错误, 则 DB2 Text Extender 返回错误码和错误信息。应用程序分配大小为 DES_MAX_MESSAGE_LENGTH 的缓冲区。如果 pErrorMessage 是空指针, 则不返回任何错误信息。</p>

用法

必须由调用 DesGetSearchResultTable 的应用程序创建与数据库的连接。

名称 pResultTableName 指已预先创建的结果表。样本目录中的实用程序 DESRESTB 创建文本句柄的结果表。调用此函数之后, 结果表将包含标识匹配搜索变元的文本值的信息。这是结果表的结构:

结果表

TEXTHANDLE	RANK	MATCHES

图 17. 结果表的结构

TEXTHANDLE 的数据类型是 DB2TEXTH 或 DB2TEXTFH。RANK 的数据类型是 DOUBLE。MATCHES 的数据类型是 INTEGER。

pSearchArgument 中的搜索变元在第201页的『第12章 搜索变元的语法』中作了描述。

若 *BrowseOption* 的值是 *BROWSE*，则 DB2 Text Extender 从位于服务器上的 DB2 Text Extender 搜索引擎返回浏览信息 *pBrowseInfo* 指向作为 DesStartBrowseSession 的输入的浏览信息。若 *BrowseOption* 的值是 *NO_BROWSE*，则 *pBrowseInfo* 指向空。

返回码

```

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

```

DesGetSearchResultTable API 函数

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
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

警告：下列返回码指示函数已返回结果，但它可能不是所期望的结果。

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

接收到浏览对话指针、句柄和选项 (DES_EXTENDED 或 DES_FAST)，该选项指示应使用词典还是不使用词典来分析文本文档。它准备与句柄相应的文本文档以获取文档文本和突出显示的信息，并返回用于重复调用 DesGetMatches 的文档句柄。

语法

```

DESRETURN
DesOpenDocument
(DES_BROWSE_SESSION BrowseSession,
SQLCHAR *pHandle,
DES_USHORT HandleLength,
DES_MATCH_MODE MatchMode,
DES_HANDLE *pDocumentHandle,
DES_MESSAGE *pErrorMessage);

```

函数自变量

表 16. DesOpenDocument 自变量

数据类型	自变量	使用	说明
DES_BROWSE_SESSION	<i>BrowseSession</i>	输入	浏览对话句柄。
SQLCHAR *	<i>pHandle</i>	输入	指向从数据库抽取的句柄的指针。
DES_USHORT	<i>HandleLength</i>	输入	<i>pHandle</i> 的长度 (不能使用 DES_NTS)。
DES_MATCH_MODE	<i>MatchMode</i>	输入	确定是否使用词典来查找突出显示信息的方式。 DES_FAST 不使用词典 DES_EXTENDED 使用词典
DES_HANDLE *	<i>pDocumentHandle</i>	输出	以迭代方式调用 DesGetMatches 的文档句柄。
DES_MESSAGE *	<i>pErrorMessage</i>	输出	由实施方案定义的信息文本。如果出现错误，则 DB2 Text Extender 返回错误码和错误信息。应用程序分配大小为 DES_MAX_MESSAGE_LENGTH 的缓冲区。如果 <i>pErrorMessage</i> 是空指针，则不返回任何错误信息。

DesOpenDocument API 函数

用法

DES_FAST 和 DES_EXTENDED 涉及查找在所浏览的文本中要突出显示的词条的语言处理的使用。有关详情，参见第244页的『用于浏览的语言处理』。指定 DES_FAST 以使用基本文本分析，并指定 DES_EXTENDED 以使用扩展的匹配。

对于 SQL 数据类型和 C 数据类型之间的映像，必须对句柄使用 SQL 符号名 SQL_VARBINARY。指向 *TextHandle* 值的 C 表达式的宿主变量类型是 SQLCHAR*。

DB2 Text Extender 将存储器分配给浏览信息。应用程序必须通过调用 DesFreeBrowseInfo 释放此存储器和相关资源。

因为 *TextHandle* 值是位数据且包含若干个 '\0' 字符，所以必须指定 *pHandle* 的长度。

调用者必须具有对包含 *pHandle* 引用的文本文档的表的读存取权。

返回码

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

限制

仅在通过调用 DesStartBrowseSession 启动了浏览对话之后，才可调用此函数。

DesStartBrowseSession

目的

启动浏览对话，创建浏览文本文档及突出显示其匹配所需的环境。它接收到一个指向浏览信息（它来自于 DesGetBrowseInfo 或 DesGetSearchResultTable）的指针，并且返回供其他浏览函数使用的浏览对话句柄。

语法

```

DESRETURN
DesStartBrowseSession
(
  DESBROWSEINFO      BrowseInfo,
  char                *pUserId,
  DESSMALLINT        UserIdLength,
  char                *pPassword,
  DESSMALLINT        PasswordLength,
  DESBROWSESESSION  *pBrowseSession,
  DESMESSAGE         *pErrorMessage);
  
```

函数自变量

表 17. DesStartBrowseSession 自变量

数据类型	自变量	使用	说明
DESBROWSEINFO	<i>BrowseInfo</i>	输入	指向浏览和突出显示文本文档中的匹配所需的信息的指针。由 DesGetSearchResultTable 或 DesGetBrowseInfo 返回该指针。
char *	<i>pUserId</i>	输入	数据库的用户 ID
DESSMALLINT	<i>UserIdLength</i>	输入	数据库的用户 ID 的长度
char *	<i>pPassword</i>	输入	数据库的口令
DESSMALLINT	<i>PasswordLength</i>	输入	数据库的口令的长度
DESBROWSESESSION *	<i>pBrowseSession</i>	输出	供其他浏览功能使用的浏览对话的句柄。
DESMESSAGE *	<i>pErrorMessage</i>	输出	由实施方案定义的信息文本。如果出现错误，则 DB2 Text Extender 返回错误码和错误信息。应用程序分配大小为 DES_MAX_MESSAGE_LENGTH 的缓冲区。如果 <i>pErrorMessage</i> 是空指针，则不返回任何错误信息。

用法

此函数打开浏览文本文档的浏览对话。提示您输入用户 ID 和口令以检查您存取数据库的权限。

DesStartBrowseSession API 函数

通过调用 DesEndBrowseSession 关闭浏览对话。

BrowseInfo 取决于搜索变元以及用于构建浏览信息的基本文本列。

返回码

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

限制

在调用此函数之前，必须用适当的“浏览选项”调用 DesGetBrowseInfo 或 DesGetSearchResultTable。

第14章 样本 API 程序

DB2 Text Extender 提供了一个样本程序 DESSAMP1.C，它位于 SAMPLES 目录中。

DESSAMP1.C 是一个示例程序，它使用 DesGetSearchResultTable 函数和您自己的浏览器程序。它遵循在第113页的图16中显示的 API 函数调用的顺序。

需要对已启用的数据库和已启用的文本列有存取权。要运行此程序，执行下列操作：

1. 可选。将源文件 DESSAMP1.C 复制至客户机上的本地目录。
2. 使用提供的 makefile（对于 OS/2、Windows NT 和 Windows 2000，它是 dессamp.mak，对于 UNIX 系统，它是 dессamp）来编译和链接样本文件。
3. 运行实用程序 DESRESTB，在打算使用样本代码的数据库中创建结果表：

`DESRESTB database-name`

此表用于存储信息（例如搜索结果）；它具有下列结构：

列	数据类型
HANDLE	DB2TX.DB2TEXTH 或 DB2TX.DB2TEXTFH
RANK	DOUBLE
MATCHES	INTEGER

第15章 对语言索引和精确索引的语言处理

DB2 Text Extender 在下列检索范围内提供语言处理:

- **索引。**当 DB2 Text Extender 分析文档以抽取要存储到文本索引中的词条时, 对文本进行语言处理, 以便为索引抽取正确的词条 这样做是为了使检索更简单、更快捷。
- **检索。**当 DB2 Text Extender 搜索整个文档索引以查找包含指定的搜索词条的文档时, 对搜索词条也进行了语言处理以使它们与索引的词条相匹配。
- **浏览。**当您浏览在搜索后找到的文档时, 使用语言处理来突出显示在文档中找到的词条。

创建索引时的语言处理

当 DB2 Text Extender 对文档创建索引和检索时, 它将对文本进行语言分析。就象您在下表中看到的那样, 语言处理量取决于索引的类型。对于 Ngram 索引, 不应应用语言处理。

用来对文档创建索引的语言处理包括:

- 基本文本分析
 - 识别词条 (符号化)
 - 将词条规范为标准格式
 - 识别句子
- 将词条缩减至其基本格式
- 停止字过滤
- 字符分解 (划分复合词)。

表18概述了当索引类型为**语言**并且没有请求其他索引特性时, 如何对词条创建索引。

表 18. 为语言索引抽取词条

文档文本	索引中的词条	语言处理
Mouse Käfer	mouse kaefer	基本文本分析 (规范化)
mice swum	mouse swim	缩减至基本格式

创建索引时的语言处理

表 18. 为语言索引抽取词条 (续)

文档文本	索引中的词条	语言处理
system-based Wetterbericht	system-based, system base wetterbericht, wetter bericht	字符分解
a report on animals	report animal	停止字过滤。 停止字是: a、 on

通过比较, 表19概述了当索引类型为**精确**时如何对词条创建索引。

表 19. 为精确索引抽取词条

文档文本	索引中的词条	语言处理
Mouse Käfer	Mouse Käfer	不进行规范化
mice swum	mice swum	不缩减至基本格式
a report on animals	report animals	停止字过滤。 停止字是: a、 on
system-based Wetterbericht	system-based Wetterbericht	不进行字符分解

基本文本分析

DB2 Text Extender 在处理基本文本分析时不使用电子词典。

识别包含非字母数字字符的词条

对文档创建索引时, 即使词条中包含非字母数字字符, 也能识别这些词条, 例如: “\$14,225.23”、“mother-in-law” 和 “10/22/90”。

下列各项也被认为是词条的一部分:

音调符号

货币符号

数字分隔符 (如 “/” 或 “.”)

电子邮件地址 (仅限英文) 中的 “@” 字符

“+” 号。

还使用特定的语言规则来识别包含下列内容的词条:

- 罗马语中的重音前缀，就象法语中的 l'aventure。
- 日期、时间和数值的国家标准格式。
- 替代词条（例如任务 / 责任），在英语中使用 “/” 来指示。
- 意大利单词中的结尾撇号，如 securita'。在打印的意大利语文本中，结尾撇号是很常见的，当字符集中不包括带音调符号的字符时，则在该字符后面输入音调符号；例如，“à” 被打印成 “a”。

将词条规范为标准格式

规范化将混合大小写的词条和包含音调符号或特殊字符的词条缩减为标准格式。当索引类型为语言索引时，将缺省执行此操作。（在精确索引中，字母的大小写保持不变 — 搜索是区分大小写的）。

例如，对词条 Computer 创建的索引是 computer，大写字母被改为小写字母。当搜索词条 computer 时，不仅会找到 computer，还将找到 Computer。在索引期间进行规范化的结果是按同样的方式对词条创建索引，而不管它们在文档中是否大写。

不仅在创建索引期间应用规范化，在检索期间也应用它。在搜索之前，搜索词条中的大写字母被更改为小写字母。例如，当搜索词条是 Computer 时，在搜索中所使用的词条却是 computer。

音调符号和特殊字符也是按类似的方法进行规范化的。école 的任何变体，例如 École，将找到 école、Ecole 等等。Bürger 将找到 buerger，Maße 将找到 masse。

若搜索词条中包括屏蔽字符（通配符），则在处理屏蔽字符之前进行规范化。例如：Bür_er 变成 buer_er。

识别句子

可以搜索同一句子中出现的词条。为此，在创建索引期间将对每个文档进行分析，以了解每个句子在何处结束。

DB2 Text Extender 提供了两种类型的句子结束识别方法:

- Universal Unicode Tokenizer，这适用于不是 Arabic 和 Hebrew 的语言。

这是更简单、但是更快速的方法。tokenizer 查找句点、感叹号或问号，前面加上记号字符，例如，字母，后跟空白、标签或换行符。要检查这确实是句子结尾，而不是用句点表示的缩写结尾，则应检查特定于语言的缩写列表。

- 对于 Arabic 和 Hebrew，采用基于 POE 的 tokenizer 方法

创建索引时的语言处理

此 tokenizer 在语言方面更高级，但是需要处理更长时间。该 tokenizer 主要是通过标点符号匹配、以及从特殊的输入类型和单词数中获得提示来查找句子结尾的。

利用基于 POE 的 tokenizer 来识别句子

基于 POE 的 tokenizer 通过使用标点符号规则和涉及到缩写处理的特定于语言的处理来确定句子（或句子段）的边界，但是其功能级别随语言不同而有很大区别。大多数使用单字节代码页的语言都有相关联的随 POE 一起提供的“缩写附录词典”。因为双字节语言通常不采用带有句点的缩写，所以“缩写附录词典”不可用于这些语言。

主要是通过标点符号匹配来确定句子结尾的。下表列示了终止标点字符及其“图形字符全局标识符”（GCGID）。

SBCS 字符的 GCGID	DBCS 字符的 GCGID	说明
SP110000	SP110080	句点
SP020000	SP020080	感叹号
SP150000	SP150080	问号
SP140000	N/A	问号（希腊问号）
N/A	JQ730080	双字节圆点

除非发生下列情况之一，否则，终止标点字符（例如，句点、感叹号或问号）都表示句子结尾：

- 在终止标点字符后面还跟有下表中所列示的结束标点字符（例如，右括号）。

SBCS 字符的 GCGID	DBCS 字符的 GCGID	说明
SP070000	SP070080	反括号
SP040000	SP040080	双引号
SP050000	SP050080	单引号
SP180000	SP070083	尖角双引号
N/A	SM140080	反花括号
N/A	SM080080	反方括号
N/A	JQ720080	直角单引号
N/A	JQ720081	直角双引号
N/A	SP200080	反钩形单引号
N/A	SP220080	反钩形双引号
N/A	SP070081	反甲壳方括号
N/A	SP070082	反尖角单引号

SBCS 字符的 GCGID	DBCS 字符的 GCGID	说明
N/A	SP070084	反转角括号
N/A	SP370080	垂直反单引号
N/A	SP370081	垂直反直角双引号
N/A	SP250084	垂直反直角括号
N/A	SP250080	垂直反括号
N/A	SP350080	垂直反花括号
N/A	SP250081	垂直反甲壳方括号
N/A	SP250083	垂直反尖角双引号
N/A	SP250082	垂直反尖角单引号

注: POE 不会将标有 N/A 的项目视作结束标点字符, 且只有在中文中才支持垂直结束标点字符。

示例:

...this sentence ends with two parentheses.))

在该示例中, 第二个括号被认为是句子结尾。然而, 在德语中, 若反引号后面跟有逗号, 则不认为该右引号表示句子结尾。

- 终止标点字符后面跟有另一个终止字符。示例:

This is a strong exclamation

最后一个感叹号被认为是句子结尾。

- 终止标点字符前面有数字或标点字符, 并且后面又跟有数字字符。这就防止了诸如 '1.25' 和 '.314' 这样的字符串来结束句子。
- 终止标点字符是一个句点和不允许出现在句子末尾的缩写的一部分。对每种语言都执行了限制缩写处理。
- 终止标点字符是一个句点并且后面没有跟空白字符 (例如, 空格或换行符)。这就避免了将类似 'III.IV' 这样的标题认为是要结束句子。

基于 POE 的 tokenizer 还执行缩写处理来确定句点是缩写的一部分, 还是表示句子结尾。可以将缩写添加至缩写附录词典中。若没有将任何词典传送到基于 POE 的 tokenizer 中, 则后面跟有句点的所有单个字母都被认为是缩写; 没有进行其他缩写处理。

文本到底是不是缩写通常容易混淆, 因为容易将缩写误认为是正常的单词后面跟有一个句点。例如, 考虑下列句子中的字符 "no." :

Enter the no. of exemptions you are claiming.
Answer each question with yes or no.

创建索引时的语言处理

但是，即使知道某些文本确实是缩写，但仍然容易混淆它是否要结束一个句子。某些缩写决不会结束一个句子，而某些缩写有时会结束句子。例如，考虑下列句子中的缩写 "Hwy.":

```
The drive along Hwy. 1 to Santa Cruz was beautiful.  
Many people speak highly of the Pacific Coast Hwy.
```

因为缩写可能会混淆，并且因为某些缩写可能不会出现在句子末尾，所以 POE 试图区分发现的缩写。若发现句点是一个有时会结束句子的缩写的一部分，则需要进一步执行处理。若 POE 确定缩写不在句子末尾，则用来表示句点的记号与缩写文本的记号连接在一起。否则，用来表示句点的记号仍然是一个独立的记号。

基于 POE 的缩写处理使用三组标准来确定一个句点是否是缩写的一部分：

- 将后面跟有句点的所有单个字母都认为是缩写。单字母缩写被划分为可能是句子结尾。
- 始终认为输入“缩写附录词典”中所包含的的单词是缩写。在该附录中发现的缩写是否可以结束一个句子是由与该附录中的单词相关联的信息来确定的。例如，“Mr.”在“美国英语”“缩写附录词典”中被认为是一个不能结束句子的缩写，而“etc.”是有时可以结束一个句子的缩写。
- 其长度为 2 到 6 个字符、后面跟有句点、并且在任何输入词典或“缩写附录词典”中都找不到的任何单词，也被认为是缩写。这是用来处理类似于“Jml.Comp.Ling.”这样的情况。由词典查表所确定的缩写通常都认为可能会结束句子。

若一个缩写被标识为可能结束句子，则 POE 会检查该缩写后面的文本以便确定该缩写是否在当前句子的结尾，即检查下一个单词是否以大写字母开头。

若缩写后面跟有两个或多个新行、新句子或新段落数据元素，则 POE 假定已经到了句子末尾。另外，若后续文本是一个倒置的问号或感叹号，则将一个句子结尾标记插入到输出中。

若 POE 确定句点是不结束句子的缩写的一部分，则它将继续搜索句子定界符。否则，在使句子结束之前，它将检查其他终止标点字符异常情况（后面跟有终止标点或结束标点）。

将词条缩减至其基本格式（主题化）

例如，在语言索引中，可以搜索 mouse，从而找到 mice。将词条缩减至其基本格式后再创建索引；对词条 mice 创建的索引是 mouse。稍后，当您使用搜索词条 mouse 时，就可以找到该文档。如果搜索 mice，也可以找到该文档。

其结果是，不管文档中出现词条 mouse 的哪一个变体，或者将哪一个变体用作搜索词条，您都将找到关于 mice 的信息的文档。

同样的，动词的不同时态都将缩减至其动词原形；例如，bought 将变成 buy。

停止字过滤

停止字就是在文档中频繁出现的字，例如介词和代词，因此不适合作为搜索词条。这些字都列示在与每种词典相关联的停止字列表中，在索引过程中将它们排除。

停止字处理是不区分大小写的。因此，停止字 about 还将排除句子中的第一个字 About。可以修改在各种语言中所提供的停止字列表。

Ngram 索引没有停止字列表。

字符分解（划分复合词）

日耳曼语，例如德语或荷兰语，其复合词相当丰富，例如 Versandetiketten，它的意思是邮件（Versand）标签（Etiketten）。这些复合词可以划分为它们的组成部分。

对于精确索引，复合词仍被作为一个字来创建索引。对于语言索引，复合词是在创建索引期间划分的。若具有语言索引，则在搜索时将划分复合词。

只要复合词的组成部分包含在文档中的一个句子中，而不管以任何顺序出现，都能找到它们。例如，当搜索德语单词 Wetterbericht（天气报告）时，也会找到包含短语 Bericht über das Wetter（关于天气的报告）的文档。

如果出现下列情况，则尝试划分词条：

- 该词条的语言使用复合词
- 该词条具有某种最小长度
- 该词条本身并不在电子词典中，但它的常用复合词在电子词典中，例如德语单词 Geschäftsbericht（商业报告）在德语词典中。

若发现有可能进行划分，则将词条的组成部分都缩减为它们的基本格式。下面是丹麦语、德语和荷兰语中的一些示例：

复合词	组成部分
børsmæglerselskab	børsmæglerselskab børs mægler selskab

复合词	组成部分
Kindersprachen	kindersprache kind sprache
probleemkinderen	probleemkinderen probleemkind kind probleem

用于检索的语言处理

查询处理的目的在于使搜索词条更简化，以便提高搜索的检索率，即，可以找到更多相关文档。对查询词条可以执行两种基本操作以达到该目的；这些操作是扩展和缩减。另外，一些搜索词条操作同时涉及到扩展和缩减。

- 扩展将从搜索词条中取出单字词条或多字词条，并将它们与一组替代搜索词条关联，其中每一个词条本身可能就是多字词条。在 DB2 Text Extender 的查询语言中，源表达式和目标表达式的设置组成了 OR “布尔”表达式。扩展保持源词条不变，在某种程度上，它们与索引类型无关。下列是一些扩展操作：

同义扩展

同义词扩展

- 缩减将搜索词条更改为比用户所指定的格式更通用的格式。因为它更改了搜索词条，因此，缩减将取决于索引类型，以确保更改的词条相匹配。因此，DB2 Text Extender 根据查询所指向的索引类型来生成缩减信息，以下是缩减方法：

主题化（参见第240页的『将词条缩减至其基本格式（主题化）』）

规范化（参见第237页的『将词条规范为标准格式』）。

停止字（参见第241页的『停止字过滤』）。

- 一些操作既更改搜索词条，又使用一组替代词条来扩展搜索词条。由于固有的缩减，这些操作还是取决于索引中所包含的信息。下列操作就属于此类情况：

字符和字屏蔽

声音扩展。

同义词

同义词是语义上相关的词。通常，这些字与搜索词条具有相同的词性（例如名词、动词等）。同义词是从每种语言的单独文件中得到的。它们通常以基本格式返回，通常不是多字词条，但有例外情况。查找同义词时，搜索词条总是缩减为其基本格式。下面是一个词在三种语言中的部分同义词：

- 英语

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

- 法语

mot:

expression parole terme vocable lettre billet missive épître
plaisanterie

- 德语

Wort:

Vokabel Bezeichnung Benennung Ausdruck Begriff Terminus
Ehrenwort Brocken Bekräftigung Versprechen Zusicherung Gelöbnis
Beteuerung Manneswort Schwur Eid Ausspruch

同义词扩展

使用通过特定关系可以找到的同义词词条，可以扩展搜索词条。这些关系可能是分层的（例如“精确词条”关系）、相关联的（例如“相关词条”关系），或者也可能是同义词关系。同义词词条可能是（通常是）多字词条。

第246页的『同义词概念』更详细地描述了同义词扩展。

当完成了同义词典查表时，没有规格化搜索词条（启动词条）。按照索引类型，从同义词典查表中产生的单词被缩减至其基本格式。

声音扩展

声音扩展通过一组发音相似的字来扩展单个的字。当不知道要搜索的词条的正确拼写时，声音扩展特别有用。

字符和字屏蔽

屏蔽是一种非语言扩展技术，在这种技术中，利用满足正则表达式的所有已索引字的逻辑和来置换该正则表达式。屏蔽的表达式及其任何扩展都不受主题化、停止字抽取或任何其他扩展技术的支配。这样会有一些影响，例如，当用屏蔽词条 `swu*` 来搜索不规则动词形式 `swum` 时，对于精确索引来说是匹配的，但是对于语言索引来说则是不匹配的，因为在语言索引中，此格式已被主题化为 `swim`。

如果使用字掩码，则性能可能会下降，特别是在大型索引中进行搜索时更是如此。

当您浏览在搜索后所找到的文档时，也要使用语言处理。它分两个阶段完成：

1. 基本文本分析：规范化和词条扩展
2. 扩展匹配。

步骤 1: 规范化和词条扩展

执行第一步时不使用电子词典。

规范化

在第236页的『基本文本分析』中描述了如何进行规范化。

词条扩展

词条扩展是将词条缩减为其基本格式的逆向过程。若索引是语言索引，则在开始搜索之前，搜索词条将缩减至其基本格式。

同样，若是语言索引，则文档的词条在被添加到索引中之前，将缩减至其基本格式。因此，根据词条的基本格式就可以找到文档。

然而，在浏览所找到的文档时，您期望看到突出显示的基本格式的所有变体。要突出显示这些变体，则要扩展所找到的基本词条。

可以生成在字典中所找到的每个词条的所有变体（词尾变体）。以下是为德语单词 gehen（走）生成的词尾变体：

```
gegangen geh   gehe   gehen   gehend  gehest  gehet  gehst  
ging       ginge  gingen  gingest  inget   ingst   ingt   geht
```

步骤 2: 扩展匹配

第二步是扩展匹配，只有当基本文本分析和规范化都不能突出显示所找到的词条时，才使用扩展匹配。扩展匹配可找到更多的模糊匹配。

通过指定 DES_EXTENDED 作为 DesOpenDocument API 函数中的参数，来选择扩展匹配。

扩展匹配使用与执行语言索引时所使用的语言处理相同的语言处理。

在下列情况下，扩展匹配可以找到附加的匹配：

- 搜索词条包括屏蔽字符，且具有词尾变体。

已处理了屏蔽字符，已对搜索词条执行了词干缩减，已找到相应的文档。不进行扩展匹配，就不会突出显示与指定的搜索标准相匹配的文本。

示例：文档包含有词尾变体的词条 `swam`。

- 索引期间，此词条将缩减至 `swim`。
 - 若搜索词条是 `swi%`，则将找到上述文档，因为词干缩减是 `swim`。
 - 若没有扩展匹配，则仅突出显示与词条 `swi%` 相匹配的那些字。利用扩展匹配，则也突出显示有词尾变体的词条 `swam`。
- 若已索引了复合词。

当用德语编写的文档中包含组合字并且正使用语言索引来索引它时，文档索引将保持组合字的组成部分及其本身保持不变。当搜索组合字的一部分时，将找到包含该组合字的文档，但是，若没有扩展匹配的话，则不能突出显示该字。

示例：包含德语单词 `Apfelbaum`（苹果树）的文档。

- 在语言索引期间，此字将缩减至 `apfel` 和 `baum`。
 - 当为了找到词条 `baum` 而搜索索引时，在整个索引中将找到词条 `Baum` 以及包含该词条的文档。
 - 若没有扩展匹配，则将没有词条被突出显示，因为文档中包含 `Apfelbaum`，而不是 `Baum`。若使用扩展匹配，则 `Apfelbaum` 组合将被分开，并找到 `Baum` 部分且突出显示。
- 若在行末的字是用连字符连接起来的。

若文字处理器自动插入了连字符，则可以找到并突出显示具有连字符的字。然而，若连字符是由用户输入的，则将找到包含该字的文档，但若不使用扩展匹配，则不会突出显示该字。

示例：文档中包含带连字符的字 `container`，该字在行末被断开，如下所示：

```
Another name for a folder is a con-
tainer.
```

- 在索引期间，该字将被规范成 `container`。
- 当搜索索引来查找词条 `container` 时，将找到该词条以及包含该词条的文档。
- 尝试突出显示与 `container` 相匹配的任何字。若没有扩展匹配，则仅当 `con-tainer` 中的连字符是由文本处理器插入的、而不是由用户输入的时候，才能找到匹配词条。

同义词典是一种受控词汇表，由语义上相关联的词条构成，通常涉及特定学科领域。可将其想象成一个语义网络，每个词条都由一个节点表示。若两个词条互相关，则它们的节点通过一个链路相连，该链路由相关名称标记。以给定词条的节点为起点，沿着它的所有连接，就可以到达与该词条直接相关的所有词条。再以前一步骤中所到达的节点为起点，沿着它的所有连接，又可以到达进一步相关的词条，依此类推。图18显示了一个微型同义词典结构的示例。

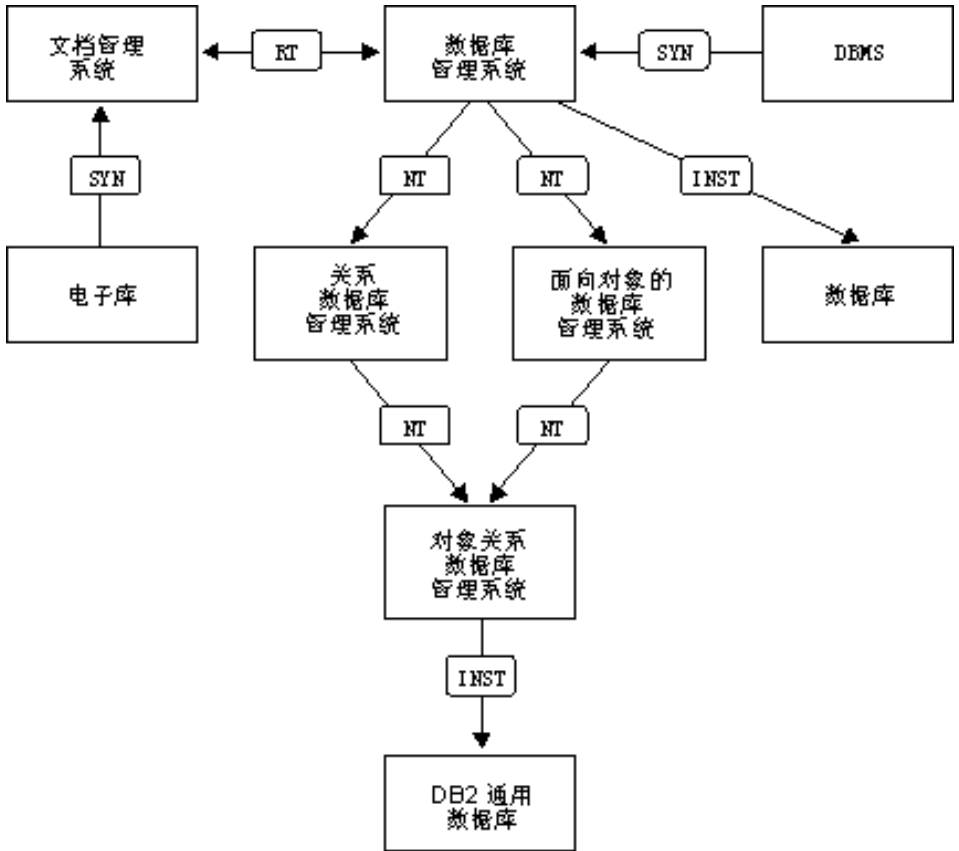


图 18. 显示成网络的同义词典

DB2 Text Extender 允许您通过在先前创建的同义词典中添加其他词条来扩展搜索词条 参考第201页的『第12章 搜索变元的语法』以了解如何在查询中使用同义词典扩充。

要创建同义词典并在搜索应用程序中使用它，要求将同义词典定义文件编译成内部格式，即同义词典。

语言索引和精确索引所使用的词典格式与 Ngram 索引所使用的不同。因而，与产品一起提供了两个不同的同义词典编译器。它们不仅是在所依据的概念上有微小差异，而且需要不同的源格式。因此，在开始为您的搜索应用程序定义同义词之前，您应首先决定将要使用的索引类型。

同义词典的基本组成部分是“词条”和“关系”。

词条

词条是一个词或表达式，用来表示在同义词主题范围内的某种概念。例如，下面可能就是一个或多个同义词典中的词条：

数据处理
直升飞机
国民生产总值

在 DB2 Text Extender 同义词典中，词条被分类为描述符或非描述符。描述符是一类同义词中的词条，在创建索引和搜索时优先选用。将该类中的其他词条称为非描述符。例如，轮廓和形状是同义词，其中形状就是描述符，而轮廓是非描述符。

Ngram 同义词典不区分描述符和非描述符。

关系

关系就是两个词条之间的关联性表达式。关系具有下列特性：

- 关系的深度是指该关系扩充的层次数。在搜索语法中，这是使用 THESDEPTH 关键字指定的。参考第201页的『第12章 搜索变元的语法』以了解如何在查询中使用同义词典扩充。
- 关系的方向性指定从一个词条到另一个词条的关系是真实相等（双向），还是仅在一个方向上相等（单向）。

同义词扩展可以使用在同义词典中定义的每种关系。还可以指定扩展的深度。它是从源词条到目标词条的转换的最大次数。然而，要注意当深度增加时，词条集可能按指数增加。

下列示例显示当深度增加时新添加的那些词条。

health

health service, paramedical, medicine, illness

allergology, virology, veterinary medicine, toxicology, surgery,
stomatology, rheumatology, radiotherapy, psychiatry, preventive

同义词概念

medicine, pathology, odontology, nutrition, nuclear medicine, neurology, nephrology, medical check up, industrial medicine, hematology, general medicine, epidemiology, clinical trial, cardiology, cancerology

DB2 Text Extender 同义词关系

以下是由 DB2 Text Extender 同义词典提供的关系类型:

- 结合
- 同义
- 分层
- 其他

在 DB2 Text Extender 同义词典中, 没有预定义的关系。可以分别为每种关系命名, 例如 BROADER TERM, 它可以是助记缩写, 例如 BT。在同义词典设计中使用的公共关系有:

- BT 或 BROADER TERM
- NT 或 NARROWER TERM
- RT 或 RELATED TERM
- SYN 或 SYNONYM
- USE
- UF 或 USE FOR

结合: 结合关系是扩展到任何深度的、描述符之间的双向关系。它将既不等价也不分层的两个词条结合起来, 然而这两词条在语义上相关联, 并达到了一定程度, 即它们之间的连接可能会提出附加词条以便在索引或搜索中使用。

结合关系通常被指定为 RT (相关词条)。示例有:

狗 RT 安全性
宠物 RT 兽医

同义: 当描述符与非描述符之间存在差异时 (就象在 DB2 Text Extender 同义词典中那样), 同义关系是指具有相同或相近意思的两个词条之间的单向关系。在一组同义词中, 其中一个词条被指定为描述符。其他词条被称为非描述符。参考第249页的『Ngram 同义词典关系』以了解描述符与非描述符之间不存在差异时同义词关系的定义。

通常指定 USE 导致从给定的非描述符到达其描述符。通常指定 USE FOR 导致从描述符到达每个非描述符。例如:

猫科动物 USE 猫
律师 UF 代言人

分层： 分层关系是指描述符之间的单向关系，它说明其中一词条比另一词条更专一化或者更不通用。这种区别导致将词条表达为层次结构，其中，一个词条代表一类，而附属词条指代其成员部分。例如，词条“老鼠”是属于类别“啮齿动物”。

BROADER TERM 和 NARROWER TERM 之间是分层关系。例如：

小型汽车 NT 轿车
马科动物 BT 马

其他： 关系类型其他是最常见的。它表示一种关联性，该关联性不容易属于其他某类别。关系类型其他可以是双向的或单向的，它没有深度限制，并且描述符和非描述符之间可以存在某种关系。

此关系通常用于同义词典中的新词条，直到可以确定它与其他词条之间的正确关系为止。

当然，您可以这样定义您自己的双向同义词关系：对描述符之间的同义词关系使用关系类型结合，甚至对任意词条之间的同义词关系使用关系类型其他。

Ngram 同义词典关系

Ngram 同义词典支持下列两种类型：

- 结合
- 同义

有两种预定义关系，各自依据其中一种类型。您可以根据结合类型定义您自己的关系。有关细节，参见第254页的『创建 Ngram 同义词典』。

结合

结合关系是一种双向关系，存在于两个不表示同一概念但互相关的两个词条之间。预定义关系 *RELATED_TO* 和所有用户定义关系都以此关系类型为基础。

示例有：

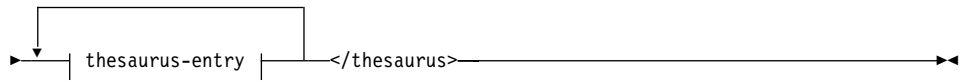
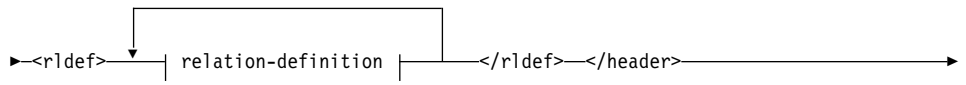
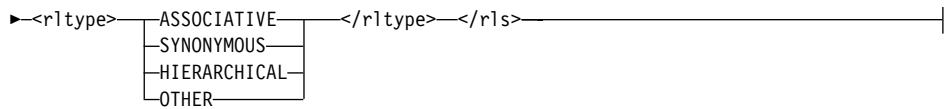
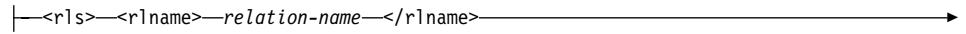
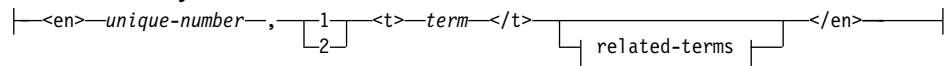
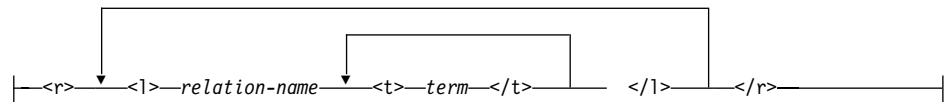
网球 RELATED_TO 球拍
德国 RELATED_TO 香肠

同义

同义关系是一种双向关系，存在于意义相同或类似且可相互替代的两个词条之间。例如，此关系可用于一个词条及其缩写。预定义关系 *SYNONYM_OF* 是唯一基于此类型的关系。

示例有：

斑点 SYNONYM_OF 污点
US SYNONYM_OF 美国

**relation-definition:****thesaurus-entry:****related-terms:**

relation-name 只能包含字符 a-z、A-Z 和 0-9。

第252页的图19显示第246页的图18中显示的同义词典的 SGML 定义。

同义词概念

```
<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>
```

图 19. 简单同义词典的定义 (1/2)

```

<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>

```

图 19. 简单同义词典的定义 (2/2)

创建 Ngram 同义词典

在安装路径的词典目录中，存储了一个英语 Ngram 同义词典编译器的样本输入文件 `desnthes.def`。OS/2 和 Windows 系统上的词典目录是：

```
drive:\dmb\db2tx\dict
```

在 AIX、HP-UX 和 SUN-Solaris 系统上，词典目录是：

```
DB2TX_INSTOWNER /db2tx/dicts
```

此样本同义词典的编译版本也存储在那里。属于此同义词典的文件被称为 `desnthes.<extension>`，带有以下扩展名，其中 `n` 是一个数字：

- 对于目录文件: `wdf`、`wdv`、`grf`、`grv`、`MEY`、`ROS`、`NEY`、`SOS`、`lkn`
- 对于临时文件:
`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` 和 `$21`

要创建 Ngram 同义词典，首先在定义文件中定义其内容。同一目录中可以有数个同义词典，但建议您对每一个同义词典使用不同的目录。不带扩展名的文件名长度一定不能超过 8 个字符。扩展名是可选的，但限于 3 个字符，并且不应与以上列示的任何扩展名不同。

若对其他 DB2 Text Extender 同义词典使用同一目录，则不要使用第250页的『创建同义词典』下面列示的扩展名。

然后通过运行下列命令编译该文件：

```
txthesn -f definition-file-name-ccsid code-page
```

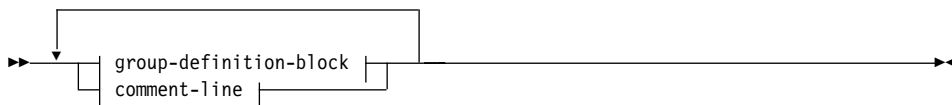
以下是 Ngram 同义词典支持的代码页的列表：

932	AIX, OS/2	日语
942	OS/2	日语
943	OS/2, Windows	日语
949	OS/2	韩语
950	AIX, HP-UX, OS/2, SUN-Solaris, Windows	繁体中文
970	AIX, HP-UX, SUN-Solaris	韩语
1381	OS/2, Windows	简体中文
1363	Windows	韩语
1383	AIX, HP-UX, SUN-Solaris	简体中文
850	AIX, OS/2	拉丁语-1

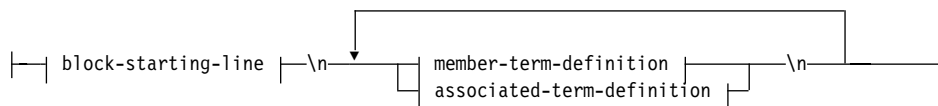
1252	Windows	拉丁语-1
------	---------	-------

`txthesn` 生成名称与 *definition-file-name* 相同，且带有上述扩展名的同义词典文件。将在定义文件所在的目录中创建这些文件。若此目录中已有同名的同义词典，它将被覆盖，且不会给出警告。

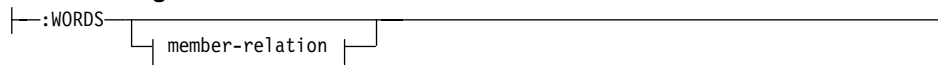
使用下列语法图来指定同义词典的内容。



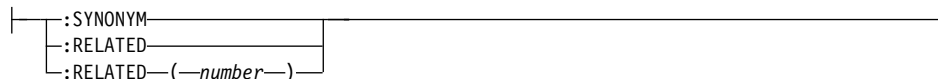
group-definition-block:



block-starting-line:



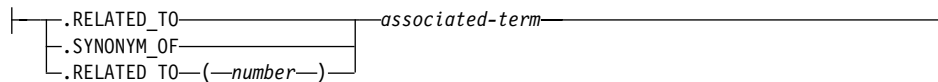
member-relation:



member-term-definition:



associated-term-definition:



comment-line:



同义词概念

每个成员词条都必须写在单独一行上。每个关联词条的前面都必须有一个关系名称。若成员词条互相关，则指定成员关系。

成员词条与关联词条的长度限制为 164 个字符。同一字母的单字节字符和双字节字符被视为是相同的。不区分大写字母和小写字母。一个词条可以包含空白字符，但可使用单字节字符句点 "." 或冒号 ":"。

用户定义关系全都以结合类型为基础。它们由 1 和 128 之间的唯一数字进行标识。

若应用程序要使用符号名（而不是关系名和编号）来表示其同义词典关系，则它必须自行负责管理映射。例如，若将关系 `OPPOSITE_OF` 定义为 `RELATED_TO(1)`，则应用程序必须将此名称映射至内部关系名 `RELATED_TO(1)`。参考第201页的『第12章 搜索变元的语法』以了解如何在查询中使用同义词典扩充。

第16章 配置文件

本章主要描述配置文件。这些文件是在用 `txicrt` 命令来创建 DB2 Text Extender 实例或者用 `descrc1` 命令来创建客户机简要表时自动生成的。这些文件是在代码页 819 中生成的。

可以编辑这些文件以调整您的系统，但是必须确保编辑文件时使用正确的代码页。部分名和选项名不区分大小写。将分号用作注解定界符。

其中，选项是布尔值，认为值 `TRUE`、`YES`、`ON` 和 `1` 为真，而不管它们的大小写；其他所有值都被认为是假。

客户机配置文件

文件名	DESCL.INI
位置	UNIX: DB2TX_INSTOWNERHOMEDIR/db2tx Windows NT 和 Windows 2000: %DMBMPATH%\instance%\DB2INSTANCE%\db2tx

在下一个 `StartSession` 函数中，已更新的选项就变为活动的了。

部分	选项	缺省值	说明
[INSTANCE]	DESWORKCL		指向一个用于临时文件的工作目录。
	DESNLPSCL		指向资源目录。
	DESDTDPATHSRV		指向一个存储了 XML 文档的 DTD 的目录。
	DESDTDPATHCL		指向必须存储了 DTD 文件的路径。用于 XML 文件。
[BUFFER]	BUFFERSEGMENTSIZ	32 000	用于缓冲的块段大小，以字节计。
	BUFFERSEGMENTCOUNT	3	在将缓冲区分配给临时文件之前所使用的最大段数。缓冲区段是由 <code>BUFFERSEGMENTSIZ</code> 来定义的。

客户机配置文件

部分	选项	缺省值	说明
[DOCUMENTFORMAT]	USEREXIT		用来处理不受支持的文档格式的用户出口的名称。指定文件名（若用户出口被存储在是 PATH 语句的一部分的目录中）或者指定全限定文件名。 有关用于格式转换的用户出口的详情，参见第33页的『使用未受支持的文档格式』。
	FORMATRECOGNITION	TRUE	触发文档格式的格式识别。 TRUE: 格式识别为 on FALSE: 格式识别为 off
	UseExitForAllFormats	FALSE	确定在何时调用了用来处理 deslodef.h 中未列示的文档格式的用户出口。必须为 USEREXIT 设置一个值。 TRUE: 一直调用该用户出口。若设置了此值，则忽略 FORMATRECOGNITION。 FALSE: 调用在值 EHW_USER_FORMATS 之上的所有文档格式的用户出口。

服务器配置文件

文件名 DESSRV.INI

位置 UNIX: DB2TX_INSTOWNERHOMEDIR/db2tx/TXINSnnn

Windows NT 和 Windows 2000:
%DMBMMPATH%\instance%\DB2INSTANCE%\db2tx\txinsnnn 对于每个服务器实例，此文件仅存在一次。

当下一次启动服务器实例时，已更新的选项就变为活动的。

表 20. 服务器配置文件选项

部分	选项	缺省值	说明
[INSTANCE]	DESWORKSRV		指向一个用于临时文件的工作目录。

表 20. 服务器配置文件选项 (续)

部分	选项	缺省值	说明
	DESNLPSRV		指向资源目录。
[DAEMON]	MaxMtEntries	30 在 AIX、 Windows、 OS/390 上 10 在 Solaris 上	在任何时刻并行使用的最大索引数。若您缺少资源（例如，信号量或共享内存），则减小此数目。可用的资源是与平台有关的，因此，缺省值也是与平台有关的。
	MaxIndexEntries	1000	已使用的最大索引数。若您缺少资源（例如，共享内存），则减小此数目。
[BUFFER]	BUFFERSEGMENTSIZ	32 000	用于缓冲的块段大小，以字节计。这是供 EhwUpdate 使用的。
	BUFFERSEGMENTCOUNT	3	在将缓冲区分配给临时文件之前，在索引更新过程期间所使用的最大段数。若您的文档集中包含大量文档，则应增大此数目。
	BUFFERSORTSIZE	20 000 000	用来对临时工作文件进行排序的缓冲区大小，以字节计。
[DOCUMENTFORMAT]	USEREXIT		用来处理 deslsdef.h 中未列示的文档格式的用户出口。指定文件名（若用户出口被存储在是 PATH 语句的一部分的目录中）或者指定全限定文件名。 有关用于格式转换的用户出口的详情，参见第33页的『使用未受支持的文档格式』。
	FORMATRECOGNITION	TRUE	触发 deslsdef.h 中所列示的文档格式的格式识别。 TRUE: 格式识别为 on FALSE: 格式识别为 off

服务器配置文件

表 20. 服务器配置文件选项 (续)

部分	选项	缺省值	说明
	UseExitForAllFormats	FALSE	<p>确定在何时调用了用来处理 desldef.h 中未列示的文档格式的用户出口。必须为 USEREXIT 设置一个值。</p> <p>TRUE: 一直调用该用户出口。若设置了此值, 则忽略 FORMATRECOGNITION。</p> <p>FALSE: 调用在值 EHW_USER_FORMATS 之上的所有文档格式的用户出口。</p>
[LINGPREC] 所有索引都将语言索引或精确索引作为它们的基本类型。	UPDATETHRESHOLD	4 000 000	索引更新进程在内部被划分成几个更新和重组运行。此值指定在一个更新步骤中要收集的单词数。
	UPDATESLICE	1	在内部重组进程开始之前所发生的更新运行数。更新运行是由 UPDATETHRESHOLD 定义的。
[NGRAM] 用于具有 DBCS 支持的所有索引。	UPDATETHRESHOLD	10 000 000	在一次更新运行期间添加至索引的文档的总大小, 以字节计。若超过了阈值, 则会启动重组进程。
	UPDATESLICE	10 000	辅助索引中的最大文档数。在每次更新运行之后就会检查此数目。若文档数大于此值, 则会启动重组进程。

文本分析配置文件

文件名	DESTAF.TCF
位置	UNIX: DB2TX_INSTOWNERHOMEDIR/db2tx/TXINSnnn Windows NT 和 Windows 2000: %DMBMPATH%\instance%\DB2INSTANCE%\db2tx\txinsnnn 对于每个服务器实例, 此文件仅存在一次。

对于除了 ngram 索引之外的每种索引类型，文本分析配置文件中都包含更新和查询处理部分。应该仅更改此处所描述的选项。若更改了其他任何项，则可能会导致在更新或查询期间发生错误。

公共选项

下列选项适用于:

- [TAF_Application:UPDATE:PREC] 适用于 PRECISE 索引类型
- [TAF_Application:UPDATE:LING] 适用于 LINGUISTIC 索引类型
- [TAF_Application:UPDATE:NORM] 适用于 PRECISE NORMALIZED 索引类型
- [TAF_Application:QUERY:PREC]
- [TAF_Application:QUERY:LING]
- [TAF_Application:QUERY:NORM]

选项	缺省值	说明
FilterSet	无，必需的	配置文件中包含了停止字过滤规则的部分的名称，例如，StwUseAllUpCs、StwIgnAllUpCs。
规范化	否	指定是否要对索引执行规范化。 若将此选项设置为“是”，则经过规范化之后，所有字符都被转换为小写，并将区别字符替换为基字符。 若将此选项设置为“否”，则会忽略 NormalizeUmlauts 和 ConvertAllUpperToLower 的项目。
NormalizeUmlauts	是	指定是否要替换元音变音，例如，将 ä 替换为 ae。此选项仅适用于用“德语”、“丹麦语”、“瑞典语”或“挪威语”编写的文档。
ConvertAllUpperToLower	否	指定是否要将全大写的单词转换为小写。

[TAF_Filter:STWUSEALLUPCS] [TAF_Filter:STWIGNALLUPCS]

这些部分中的选项描述停止字过滤器集。 [TAF_Filter:STWUSEALLUPCS] 这一部分将全大写的单词作为混合大小写来对待，而 [TAF_Filter:STWIGNALLUPCS] 这一部分定义对全大写单词的特殊处理。

选项	缺省值	说明
FilterType	TAF_FILTER_STW	配置文件中包含了停止字过滤规则的部分。
MinTokenLength	3	记号所需要的最少字符数。若记号的长度比此值小，则将不会对它进行索引。这是一个整数值 ≥ 1 。

文本分析配置文件

选项	缺省值	说明
MinUpperCaseTokenLength	2	全大写记号所需要的最少字符数。若记号的长度比此值小，则将不会对它进行索引。这是一个整数值 ≥ 1 。
UseAlternateTerritories	否	指定若找不到特定于语言的词典的话，是否要使用替代词典来进行停止字处理。 若将此选项设置为“是”，则使用与语言匹配的第一个停止字词典。例如，若当前语言是 en-gb，但是没有 en-gb.tsw 停止字文件，则使用 en-us.tsw 停止字文件。 若将此选项设置为“否”，则不执行停止字过滤。

[TAF_AnnotationType:TAF_ThesaurusEntry]

这一部分用来为需要同义词典扩展的那些查询设置缺省值。

关键字	缺省值	说明
ThesBaseName	desthes	设置同义词典（它用于需要查询扩展的查询）的缺省名称。该名称是一个字符串。
Depth	1	设置浏览同义词典（它用于需要同义词典扩展的查询）的缺省路径。这是一个整数值 ≥ 1 。

[TAF_Plugin:TAF_POE]

要更改在启动 文本搜索引擎 时预装入的语言环境的缺省值，您可以在 PreloadLanguages 选项中定义语言。

选项	缺省值	说明
PreloadLanguages	EN-US	指定在启动时要预装入的一组特定于语言的资源。例如，若应用程序要处理用“意大利语”编写的文档，则会装入用于“意大利语”的资源。该字段的值采用 XX-YY 格式，其中，XX 表示语言，YY 表示地区。

第17章 返回码

本章列示由 DB2 Text Extender API 返回的响应函数调用的代码 按字母顺序列示它们。

所有 DB2 Text Extender API 调用返回数字返回码作为 C 函数值返回码是在随 DB2 Text Extender 一起提供的包括文件 DES_EXT.H 中定义的。

DB2 Text Extender API 中断错误状态并用返回码报告错误情况

调用 DB2 Text Extender API 函数的应用程序应总是在尝试处理任何其他输出参数之前检查返回码 在第213页的『第13章 用于搜索和浏览的 API 函数』中列示每一个调用的可能返回码以及它们的参数。

在某些情况下，不正确的输入（例如，过时的对话指针）可能会导致在不能由 DB2 Text Extender 截取的 API 服务中发生异常结束情况。

RC_ALLOCATION_ERROR

解释： 不能分配存储器用于内部使用。

校正方法： 确保有足够的内存可用。

RC_FILE_IO_PROBLEM

解释： DB2 Text Extender 不能读或写文件

校正方法： 检查服务器上是否有足够的磁盘空间和内存可用。检查环境变量和文本配置设置是否设置正确。

RC_INVALID_BROWSE_INFO

解释： 由 DesGetSearchResultTable 或 DesGetBrowseInfo 返回并用作对 DesStartBrowseSession 的输入的浏览信息无效。

校正方法： 检查程序设计错误是否覆盖了浏览信息。

RC_INVALID_BROWSE_OPTION

解释： DesGetSearchResultTable 中的浏览选项无效。

校正方法： 确保该选项是 BROWSE 或 NO_BROWSE。

RC_INVALID_MATCH_OPTION

解释： 在 DesOpenDocument 中使用的匹配选项无效。

校正方法： 检查该选项是 FAST 还是 EXTENDED。

RC_INVALID_PARAMETER

解释： 下列其中一种输入参数是不正确的。

校正方法： 读取由 DB2 Text Extender 返回的错误信息来确定原因

返回码

RC_INVALID_SEARCH_OPTION

解释: DesGetSearchResultTable 中的搜索选项无效。

校正方法: 确保该选项是 DES_TEXTHANDLEONLY、DES_RANK、DES_MATCH 或 DES_RANKANDMATCH。

RC_INVALID_SESSION

解释: 在当前服务调用中指定的对话指针不正确或过时。

校正方法: 保存任何有助于查找错误原因的信息，然后结束应用程序。

RC_NO_BROWSE_INFO

解释: DB2 Text Extender 没有返回浏览信息 这是因为搜索变元导致空的搜索结果。这不是错误。

校正方法: 没有必要的操作。

RC_PARSER_INVALID_ESCAPE_CHARACTER

解释: 搜索条件包含不正确的换码字符。如果空格用作换码字符，或（对一个字或短语）在搜索条件中指定多个换码字符，则报告此错误。示例：ESCAPE " " 或 ESCAPE "#\$".

校正方法: 检查搜索变元的语法，并再次尝试。

RC_PARSER_INVALID_USE_OF_ESCAPE_CHAR

解释: 不能解释搜索条件中的换码字符语法。

校正方法: 检查换码字符语法。例如，如果 \$ 是指定的换码字符，则字或短语只能包含 \$\$、\$_ 或 \$%，其中，_ 和 % 是两个屏蔽符号。

RC_PARSER_SYNTAX_ERROR

解释: 不能解释搜索条件语法。

校正方法: 通过参考第201页的『第12章 搜索变元的语法』来检查搜索变元的语法。

RC_RESULT_TABLE_NOT_EXIST

解释: 尝试在并不存在的表中存储搜索结果。

校正方法: 创建结果表，如在第227页的图17中显示的那样。

RC_SE_BROWSER_TIME_OUT

解释: 浏览过程开始，但是并未在可接受的时间内响应。DB2 Text Extender 然后取消暂挂过程

在系统没有足够的存储空间或超负荷时，可能发生此错误。

校正方法: 通过调用 DesEndBrowseSession 终止浏览对话，通过调用 DesFreeBrowseInfo 释放已分配的存储器，然后再次尝试。

RC_SE_CAPACITY_LIMIT_EXCEEDED

解释: 不能处理所请求的功能。有足够的内存或磁盘空间。

校正方法: 结束程序并检查系统的资源。

RC_SE_COMMUNICATION_PROBLEM

解释: 与 DB2 Text Extender 服务器的通信失败 可能是因为没有足够的存储空间或因为 DB2 Text Extender 的不正确安装而导致该错误

校正方法: 保存任何有助于查找错误的信息，然后结束应用程序。

RC_SE_CONFLICT_WITH_INDEX_TYPE

解释: 查询的搜索词条的语言说明不符合索引类型。例如, 不能对语言索引使用 PRECISE FORM OF。如在第207页的表8中显示的那样使用缺省语言规范。

校正方法: 改写您的应用程序以防止查询选项说明与索引类型相冲突。

RC_SE_DICTIONARY_NOT_FOUND

解释: DB2 Text Extender 语言服务找不到词典文件在没有语言支持的情况下处理查询。 对应于指定语言码的词典文件不在预期的路径中。

校正方法: 可以继续 API 调用。对于 UNIX, 检查必需的词典是否在路径 {DB2TX_INSTOWNERHOMEDIR}/db2tx/dicts 中。对于 OS/2, 检查必需的词典是否在文本配置设置 DB2TX_DATA 中指定的路径中。必要时, 安装必需的词典。

RC_SE_DOCMOD_READ_PROBLEM

解释: 创建了 DB2 Text Extender 实例时, 一个称为 desmodel.ini 的文档模型文件被放置在实例目录中。创建索引时, 还将一个 desmodel.ini 文件放在索引目录 IXnnnnnn 中。无法读取此文档模型文件。

校正方法: 检查是否存在文档模型文件, 且它是否在正确的目录中。

RC_SE_DOCUMENT_NOT_ACCESSIBLE

解释: 所请求的文本文档已找到, 但当前它是不可存取的。

校正方法: 检查是否由另一个任务或用户独占地存取文档。

RC_SE_DOCUMENT_NOT_FOUND

解释: 未找到请求的文本文档。最可能的原因是已从存储器中删除了文本文档, 但是尚未从 DB2 Text Extender 索引中除去它 这还可能在尝试浏览由损坏的句柄标识的文档时发生。

校正方法: 在大部分情况下, 可以忽略此返回码。在下一次索引更新后, 将不再显示它。

如果它是持续的, 则检查应用程序是否正确地传送到找到的句柄以进行浏览。

RC_SE_EMPTY_INDEX

解释: 由搜索请求寻址的对应于句柄列的 DB2 Text Extender 索引是空的 没有将文本文档添加至此索引, 或者已从索引中去除了所有文本文档。

在启用了文本列, 但是尚未索引该列中的文档时, 可以出现这种情况。也就是说, 在 ENABLE TEXT COLUMN 命令中指定稍后, 在由定期索引设置确定的时间, 创建索引。

在启用了文本表来为所有文本列创建空的公共索引, 但尚未启用一个文本列时, 也会出现这种情况。

校正方法: 如果 ENABLE TEXT TABLE 已用于为所有的文本列创建空的公共索引, 则至少为一个包含要搜索的文本的文本列运行 ENABLE TEXT COLUMN。在此命令中, 可以确定是立即创建索引, 还是在由定期索引设置确定的时间创建它。

运行 GET INDEX STATUS 来检查是否成功构建了索引。

RC_SE_EMPTY_QUERY

解释: 由 DB2 Text Extender 在语言方面分析并且处理指定的搜索条件 程序设计错误导致生成不包含搜索词条的查询, 或者所有的搜索词条都是从查询中除去的停止字 (不由 DB2 Text Extender 索引的字) 结果是没有搜索词条。

返回码

校正方法： 重新进行查询。如果问题仍然存在，检查是否有程序设计错误。

RC_SE_END_OF_INFORMATION

解释： 这不是错误。已到达文档的末尾。无更多信息可用于 DesGetMatches。

校正方法： 使用此返回码来结束用 DesGetMatches 对文档进行的迭代处理。

RC_SE_FUNCTION_DISABLED

解释： 所请求的函数调用了已由管理员阻止的 DB2 Text Extender 函数。

校正方法： 请求管理员进行帮助。可能有必要停止并重新启动 DB2 Text Extender (txstop/txstart)

RC_SE_FUNCTION_IN_ERROR

解释： 由于 DB2 Text Extender 服务器上的错误情况，已锁定所请求的函数不能处理 API 调用。

校正方法： 检查索引状态。检查索引目录中可用的空间。重设索引状态并重试命令。

RC_SE_INCORRECT_HANDLE

解释： 在输入参数中指定的句柄（如浏览对话句柄）无效。它必须是由前一调用返回的句柄，并且不是过时的。

校正方法： 保存任何有助于查找错误原因的信息，然后通过调用 DesEndBrowseSession 终止对话。

检查程序设计错误是否产生了不正确的句柄。

RC_SE_INDEX_DELETED

解释： 删除正在存取的 DB2 Text Extender 索引。

校正方法： 与 DB2 Text Extender 管理员联系以重建索引。

RC_SE_INDEX_NOT_ACCESSIBLE

解释： 不能存取 DB2 Text Extender 索引，并且不能处理当前调用。

校正方法： 请 DB2 Text Extender 管理员检查索引的可存取性。

RC_SE_INDEX_SUSPENDED

解释： DB2 Text Extender 接收到与从另一个对话或从当前对话暂停的索引相关的请求。

校正方法： 请 DB2 Text Extender 管理员检查索引的状态。

RC_SE_INSTALLATION_PROBLEM

解释： DB2 Text Extender 已遇到安装问题

校正方法： 检查环境变量 DB2INSTANCE、DB2TX_INSTOWNER、DB2TXINSTOWNERHOMEDIR 的当前设置。使用 descfgcl -d 和 descfgsv -d -i txinsnnn 来检查搜索服务配置。

RC_SE_IO_PROBLEM

解释： 当服务器试图打开或读取它的索引文件时出现错误。这可能是由于下列其中一种原因：

管理员无意的操作，例如，删除 DB2 Text Extender 索引文件。

文本配置中的不正确设置。

校正方法： 终止应用程序。与管理员一起检查：

- 当前 DB2 Text Extender 索引的所有文件是否存在
- 文本配置设置是否正确。

RC_SE_LS_FUNCTION_FAILED

解释： 存取数据库以检索文本文档来进行浏览的功能失败。对用户来说，数据库不再是可存取的；或者，对文本表来说，用户未被授权。

校正方法： 检查对函数的输入（例如用户 ID）是否正确。检查数据库是否是可存取的，并检查用户是否被授予任务的权限。

RC_SE_LS_NOT_EXECUTABLE

解释： 不能执行尝试存取数据库以检索文本文档来进行浏览的功能。

校正方法： 检查是否正确安装了 DB2 Text Extender。如果问题仍然存在，与 IBM 代表联系。

RC_SE_MAX_OUTPUT_SIZE_EXCEEDED

解释： 已找到异乎寻常的大量的匹配。浏览信息的大小已超过可处理的最大值。不能处理该请求。

校正方法： 使查询更特定，或确保有更多的系统内存可用。

RC_SE_MAX_NUMBER_OF_BUSY_INDEXES

解释： 因为当前活动索引数已达到最大量，所以所请求的功能已被搜索服务阻止。

校正方法： 短时间后，重新发出功能调用。一般来说，问题只是暂时的。

RC_SE_NO_DATA

解释： 这不是错误。没有文本文档与搜索条件相匹配。如果请求浏览信息，则不返回任何浏览信息。对于浏览信息，不分配任何存储器。

校正方法： 不需要任何操作。

RC_SE_NOT_ENOUGH_MEMORY

解释： 在客户机或服务器系统上没有足够的存储空间。不能处理当前请求。

校正方法： 释放存储空间并结束应用程序。

RC_SE_PROCESSING_LIMIT_EXCEEDED

解释： 当前搜索请求超过了为客户机 / 服务器环境所指定的最大结果大小或最大处理时间。取消了该请求。

校正方法： 使搜索请求更特定。考虑增加最大处理时间。

RC_SE_QUERY_TOO_COMPLEX

解释： 指定的查询太复杂。

校正方法： 适应应用程序以防止屏蔽字符和同义词的过度使用。

屏蔽符号或 SYNONYM 选项的过度使用可能使查询扩展至不能由 DB2 Text Extender 管理的大小

RC_SE_REQUEST_IN_PROGRESS

解释： 当另一个浏览 API 请求对同一个对话是活动的时候，调用了 DB2 Text Extender 浏览 API 服务。

校正方法： 通过调用 DesEndBrowseSession 结束对话，并且通过调用 DesFreeBrowseInfo 释放存储器。

返回码

DB2 Text Extender 浏览 API 不支持对同一个浏览对话的并行存取。

所有在同一个过程中并行运行的应用程序应当处理它们自己的浏览对话。

RC_SE_SERVER_BUSY

解释: 当前, DB2 Text Extender 客户机不能创建与所请求的 DB2 Text Extender 服务器的对话, 或者, DB2 Text Extender 服务器通信链路已被中断且不能重建。

已正确启动了 DB2 Text Extender 服务器, 但是达到了最大数目的并行服务器过程。

校正方法: 如果这不是临时问题, 则更改服务器上的通信配置。

RC_SE_SERVER_CONNECTION_LOST

解释: 客户机和服务器之间的通信已被中断, 并且不能重建。

可能已由管理员停止了 DB2 Text Extender 服务器任务, 或者可能已关闭了服务器工作站。

校正方法: 检查这些情况中的任何一种是否已发生, 并且校正它们。

RC_SE_SERVER_NOT_AVAILABLE

解释: DB2 Text Extender API 服务不能创建与所请求的 DB2 Text Extender 服务器的对话。

可能尚未启动 DB2 Text Extender 服务器。

校正方法: 检查是否正确启动了 DB2 Text Extender 服务器 如果错误仍然存在, 则有安装问题。

RC_SE_STOPWORD_IGNORED

解释: 当指定的查询至少包含一个只由停止字组成的搜索词条时, 返回此信息码。处理查询时, 忽略搜索词条。

校正方法: 可以继续发出 API 调用。避免在 DB2 Text Extender 查询中使用停止字。

RC_SE_UNEXPECTED_ERROR

解释: 发生了可能由 DB2 Text Extender 的不正确安装导致的错误。

校正方法: 结束应用程序, 保存任何有助于查找错误原因的信息。

RC_SE_UNKNOWN_INDEX_NAME

解释: 与文本列相关联的文本索引的名称是句柄的一部分。

校正方法: 确保用作 DesGetBrowseInfo 的输入的句柄是正确的。

RC_SE_UNKNOWN_SECTION_NAME

解释: 指定的节名不是文档模型文件中指定的模型的一部分, 或不是使用的缺省模型的一部分。

校正方法: 指定作为指定模型或缺省模型的一部分的节名。

RC_SE_UNKNOWN_SERVER_NAME

解释: DB2 Text Extender 服务器的名称是句柄的一部分。

校正方法: 确保用作 DesGetBrowseInfo 的输入的句柄是正确的。

RC_SE_WRITE_TO_DISK_ERROR

解释: 可能由 DB2 Text Extender 服务器工作站上的磁盘满或 DB2 Text Extender 的不正确安装而导致了发生写入错误。

校正方法: 结束应用程序, 保存任何有助于查找错误原因的信息。检查服务器上是否有足够的磁盘空间可用。

RC_SQL_ERROR_WITH_INFO

解释: 发生了 SQL 错误。返回错误信息。

校正方法: 检查由 DB2 Text Extender 返回的错误信息以了解更多信息, 如, SQL 错误信息、SQLState 和本机 SQL 错误码。

RC_SQL_ERROR_NO_INFO

解释: 发生了 SQL 错误。不返回任何错误信息。

RC_TEXT_COLUMN_NOT_ENABLED

解释: 指定的句柄列不是在指定的表中的列。

校正方法: 检查指定的句柄列名是否正确。确保已启用了该表中的文本列。

返回码

第18章 信息

本章描述下列内容:

- 由 **DB2 Text Extender** 函数返回的 **SQL 状态**: 当使用 DB2 Text Extender 函数时就可以显示这些信息。
- 来自 **DB2TX** 命令行处理器的信息: 当使用命令行处理器 DB2TX 来输入命令时, 可以显示这些信息。每一个信息号的前缀是 DES。

由 DB2 Text Extender 函数返回的 SQL 状态

由 DB2 Text Extender 提供的 SQL 函数可以返回错误状态。示例:

```
SQL0443N User-defined function
"DB2TX.CONTAINS" (specific name "DES5A")
has returned an error SQLSTATE with
diagnostic text "Cannot open message file".
SQLSTATE=38702
```

本节中的信息按 SQLSTATE 号排列。

01H10 不能打开文件 *file-name*。

校正方法: 确保文件存在, 并且 DB2 实例名具有打开它的必要许可权。

01H11 文本句柄是未完成的

解释: 试图使用已经初始化但未完成的句柄。已使用包含文档语言和格式的预置值的 INIT_TEXT_HANDLE 创建部分句柄。但是, 句柄尚未由触发器完成。

校正方法: 只使用已完成的句柄。如果有关的句柄存储在句柄列中, 则启用或重新启用它的相应的文本列。

01H12 搜索变元太长。忽略了第二个变元。

解释: REFINE 函数用于组合两个搜索变元, 但是, 搜索变元的组合长度大于对 LONG VARCHAR 所允许的最大值。REFINE 函数返回第一个搜索变元, 而不是所组合的搜索变元。

校正方法: 减小一个或两个搜索变元的长度, 然后重复查询。

01H13 搜索变元包含停止字。

解释: 指定的查询至少包含一个只由停止字组成的搜索词条。处理查询时, 忽略搜索词条。

校正方法: 避免在 DB2 Text Extender 查询中使用停止字

01H14 用于语言处理的语言词典丢失。

解释: DB2 Text Extender 语言服务找不到词典文件在没有语言支持的情况下处理查询。对应于指定语言码的词典文件不在预期的路径中。

校正方法: 对于 UNIX, 检查必需的词典是否在路径 {DB2TX_INSTOWNERHOMEDIR}/db2tx/dicts 中。对于 OS/2, 检查必需的词典是否在文本配置设置中指定的路径中。必要时, 安装必需的词典。

由 DB2 Text Extender 函数返回的 SQL 状态

01H15 语言搜索词条说明与索引类型不匹配。

解释: 查询的搜索词条的语言说明不符合索引类型。例如, 不对语言索引使用 PRECISE FORM OF。如在第207页的表8中显示的那样使用缺省语言规范。

校正方法: 改写您的应用程序以防止查询选项说明与索引类型相冲突。

38700 Text Extender 库不是当前的。

解释: 试图使用只能由 DB2 Text Extender 的更新版本解释的句柄。

校正方法: 确保正确设置了至当前库版本的路径, 并且拥有存取它的必要许可权。

在 IMPLEMENTATION 列中查看 DB2 目录视图 SYSCAT.FUNCTIONS, 以了解导致问题的函数。

38701 *tracefile* 不能打开此跟踪文件。

解释: 试图使用写入目录 DB2TX_TRACEDIR 中的文件 DB2TX_TRACEFILE 的跟踪功能。文件不存在、找不到或对文件的必要许可权不可用。

38702 不能打开信息文件 *message-file*。

解释: 出现了导致 DB2 Text Extender 试图返回信息的情况 包含信息的文件不存在或找不到, 或者对文件的必要许可权不可用。

校正方法: 确保文件存在, 路径设置正确, 并且拥有打开文件的必要许可权。

38704 文本句柄的格式不正确。

解释: 具有不正确格式的句柄被用作 DB2 Text Extender 函数的自变量。

校正方法: 确保句柄不是由 INIT_TEXT_HANDLE 产生的。

38705 *udfname* 不正确的 UDF 说明。

解释: 已经在说明了函数的脚本中更改了 DB2 Text Extender 函数的特定名称。可以更改 DB2 Text Extender 函数名, 但不能更改它们的特定名称。

校正方法: 检查包含了 DB2 Text Extender 函数说明的脚本 DESCVDF.DDL, 以确保仍在使用正确的名称。对照原始分发媒体中的名称来检查这些名称。

38706 *attribute* 不能识别此属性值。

解释: 试图将 CCSID、格式或语言设置为未知值。

校正方法: 参考第29页的『第4章 计划搜索需要』以了解正确值

38707 尚不支持所请求的函数。

解释: 尚不支持所指定的函数。

校正方法: 检查指定的函数。

38708 *return code*

解释: 处理搜索请求时出现错误。

校正方法: 参考第263页的『第17章 返回码』中的返回码说明。

38709 没有足够的内存可用。

解释: 没有足够的内存可用来运行 DB2 Text Extender 函数。

校正方法: 关闭任何不必要的应用程序来释放内存, 然后再试。

38710 *errornumber* 不能存取搜索结果。

解释: 试图读取由搜索服务返回的找到的文档的列表 (结果列表) 时出错。

校正方法: 尝试重复搜索。如果不成功, 则重新启动搜索服务。如果问题仍然存在, 则将它报告给本地 IBM 代表, 并说明错误号。

38711 严重的内部错误。

解释: 出现了严重的错误。

校正方法: 将错误报告给本地 IBM 代表, 说明它所发生的环境。

38712 *indexname* 此文本索引中的不正确句柄。

解释: 句柄已被损坏。

校正方法: 使用 UPDATE INDEX 来重建索引。

38714 缩短
DB2TX_INSTOWNERHOMEDIR
环境变量。

解释: 实例拥有者的主目录名必须不多于 256 个字符。

校正方法: 使用链路来减少目录名的长度。

38717 找不到所指定的同义词典。

解释: 找不到所指定的同义词典。

校正方法: 检查所指定的同义词典名。

38718 在同义词典中找不到指定的关系名称。

解释: 在指定的同义词典中不存在指定的关系。

校正方法: 确保存在指定的关系。

38719 发生了搜索处理错误。原因码: *rc*。

解释: 由于指定的原因而不能执行搜索。

校正方法: 尝试由原因码报告的问题。若指定的原因没有什么帮助作用, 并且在 *desdiag.log* 文件中找不到进一步的信息, 则创建跟踪, 并将错误信息报告给当地的 IBM 代表。

38720 发生了共享内存连接错误。

解释: 系统不能存取共享内存。

校正方法: 检查系统配置, 并增加共享资源, 或者检查当前共享资源的使用情况 (*ipcs*), 并清除不再需要的资源。

38721 发生了信号创建 / 存取错误。

解释: 系统不能创建或存取信号。

校正方法: 检查系统配置, 并增加共享资源, 或者检查当前共享资源的使用情况 (*ipcs*), 并清除不再需要的资源。

38722 搜索进程未返回。

解释: 处理搜索请求时出现错误。

校正方法: 验证系统配置 *descfgcl*, 并检查是否所有节点都在运行。

38723 索引 **CCSID** 和查询 **CCSID** 不匹配。

解释: 用来查询字符串的数据库 **CCSID** 与文本索引的 **CCSID** 不相同。

校正方法: 禁用文本索引, 并使用数据库的 **CCSID** 来重新创建文本索引。

38724 节名或模型名不正确。

解释: 在查询中指定的节名或模型名不正确。

校正方法: 检查节名或模型名。

38726 发生了模型文件读取错误。

解释: 找不到或打不开模型定义文件。

校正方法: 检查索引目录中是否存在模型定义文件。

来自 DB2 Text Extender 的信息

每一条信息都有一个由前缀 (DES)、信息号和后缀字母组成的信息标识符。后缀字母指示产生信息的事件的严重程度。

DB2 Text Extender 中的信息

- I 资料式信息
- W 警告信息
- N 错误（或“否定的”）信息
- C 严重错误信息。

DES0001N db2txinstance 命令的自变量数不正确。

解释： db2txinstance 命令需要两个自变量。

校正方法： 再次输入带有这些自变量的命令：

```
db2txinstance instanceName db2InstanceName
```

其中, *instanceName* 是指定作为此实例的所有者的现存 UNIX 用户的注册名, *db2InstanceName* 是对应 DB2 实例的所有者的注册名。

DES0002N 无效的 instanceName。

解释： 指定的实例名必须是现存 UNIX 用户的注册名。

校正方法： 校正实例名, 或选择现存 UNIX 用户, 或创建 UNIX 用户来作为实例所有者。

再次输入 db2txinstance 命令, 如下所示:

```
db2txinstance instanceName
```

其中, *instanceName* 是所选择的 UNIX 用户的注册名。

DES0004N 指定的实例已经存在。不能处理该命令。

解释： *instanceName* 指定作为实例所有者的 UNIX 用户的注册名。此实例所有者在主目录中已有 db2tx 目录。

校正方法： 要创建实例, 除去现存的实例, 然后再试该命令。

DES0005N 找不到安装信息目录。

解释： 安装脚本所必需的信息目录已从系统中丢失; 它可能已被删除, 或者可能不正确地装入了数据库产品。

校正方法： 验证是否正确地安装了 db2tx_01_01_0000.client 产品选项。如果有验证错误, 则重新安装选项。

DES0015W 语言搜索词条说明与索引类型不匹配。

解释： 查询的搜索词条的语言说明不符合索引类型。例如, 不应为语言索引使用 PRECISE FORM OF。如在第207页的表8中显示的那样使用缺省语言规范。

校正方法： 改写您的应用程序以防止查询选项说明与索引类型相冲突。

DES0016W 对于当前索引类型, 语言说明不受支持。

解释： 对于指定的索引类型, 所指定的语言不受支持。

校正方法： 参见文档, 以了解对于索引类型受支持的语言的列表。

DES0017W 尚未启用功能部件抽取。

解释： 在查询中使用了功能部件搜索变元, 但未用索引选项 FEATURE_EXTRACTION 构建索引。

校正方法： 将索引选项更改为 FEATURE_EXTRACTION。

DES0018W 对于当前索引类型, *option* 不受支持。

解释: 请求了当前索引类型和索引选项不支持的搜索选项。

校正方法: 检查哪一个索引类型或索引选项支持所请求的搜索选项。 参见第207页的表8。

DES0121N 不能分配内存 (*malloc* 失败)。

解释: 不能为应用程序保留存储器。

校正方法: 增加调页空间。

DES0333N 发生了文本索引文件 I/O 问题。

解释: DB2 Text Extender 客户机不能创建与所请求的服务器的对话

校正方法: 检查 DB2 Text Extender 服务器是否已启动 如果未启动, 则运行 TXSTART。

DES0709W 没有安装指定语言的词典。

解释: DB2 Text Extender 找不到词典文件

校正方法: 安装或重新安装指定语言的词典。

DES0377N 发生了文本索引文件 I/O 问题。

解释: DB2 Text Extender 不能存取文本索引 如果文本配置中的 DIRECTORY 设置指向无效目录, 就可能发生这种情况。

校正方法: 检查文本配置设置。

DES0700N 节点号值 '*node*' 未包含在节点组定义中。

解释: 指定的节点号无效。

校正方法: 检查 DB2 节点号。

DES0701N 节点号值 '*node*' 超出范围。

解释: 指定的节点号无效。

校正方法: 检查 DB2 节点号。

DES0704N 格式 '*format*' 需要索引特性的说明。

解释: 文档格式与索引类型信息不兼容。

校正方法: 指定与文档格式相兼容的索引特性。

DES0705N 在模型定义文件中找不到指定的文档模型名 '*model*'。

解释: 在模型定义文件中找不到文档模型名。注意, 模型名是区分大小写的。

校正方法: 使用模型定义文件中指定的模型名。

DES0706N 不能在 DB2 Text Extender 服务器上存取模型定义文件。

解释: 找不到或打不开模型定义文件。

校正方法: 检查模型定义文件是否存在。

DES0707N 格式 '*format*' 不支持指定的索引特性。

解释: 文档格式不支持索引特性。

校正方法: 指定与文档格式相兼容的索引特性。

DES0710N 参数 '*parameter*' 不允许空指针。

解释: 没有为 *parameter* 指定任何值。

校正方法: 为该参数指定一个值。

DES0711N 发生了 Text Extender 内部错误。
诊断信息: *message*。

解释: 发生了内部处理错误。

校正方法: 检查诊断信息以解决该问题。若此内部错误不是由安装问题引起的, 则 *desdiag.log* 文件或创建的跟踪文件中可能还有其他信息。如果这没什么帮助, 则收集可获得的信息, 并与 IBM 服务代表联系。

DB2 Text Extender 中的信息

DES0712N 参数 '*parameter*' 太长。

解释: 指定的参数超出了范围。

校正方法: 指定具有有效长度的参数。

DES0713N 参数 '*parameter*' 的长度不正确:
%d1。

解释: 指定的参数超出了范围。

校正方法: 指定具有有效长度的参数。

DES0714N 直接指定参数 *parameter*, 或在配置表中指定。

解释: 未指定 CCSID、格式或语言, 且没有任何此值的文本配置设置。

校正方法: 直接在 ENABLE TEXT COLUMN 命令中指定丢失的参数, 或者在文本配置设置中设置一个值。

DES0715N 对于文本数据, 数据类型 *schema.type* 不受支持。

解释: *schema.type* 是文本列的模式名和类型名, 或者是某个存取函数的结果。文本列的数据类型不受 DB2 Text Extender 支持。它必须是 CHAR、GRAPHIC、VARGRAPHIC、LONG VARGRAPHIC、DBCLOB、VARCHAR、LONG VARCHAR 或 CLOB。如果不是这种情况, 则必须提供存取函数, 它的输入是文本列的数据类型, 输出是 VARCHAR、LONG VARCHAR 或 LOB。

校正方法: 如果 *schema.type* 是文本列类型, 则必须注册具有类型 VARCHAR、LONG VARCHAR 或 LOB 的结果的存取函数。如果 *schema.type* 是存取函数的结果, 则不能使用它。提供带有必需类型的结果的存取函数。

DES0716N 格式 *format* 不受支持。

解释: *format* 是不受 Text Extender 支持的格式。

校正方法: 检查第32页的『受支持的文档格式』中受支持格式的列表。

DES0717N 语言 *language* 不受支持。

解释: *language* 是一种 DB2 Text Extender 不支持的语言, 或者指定的索引类型不支持所选择的语言。

校正方法: 检查第53页的表5中受支持语言的列表

DES0718N CCSID *ccsid_value* 是不受支持的。

解释: 指定了无效的 CCSID 值。

校正方法: 参见文档以了解受支持的 CCSID 的列表。

DES0719N 对 DB2 Text Extender 程序 *program* 的调用失败, 返回码为 *rc*。

解释: 安装期间可能出现了错误。在文件 DES_EXT.H 中列示返回码。

校正方法: 检查安装是否成功。检查环境变量(例如, DB2TX_INSTOWNER 和 DB2TX_INSTOWNERHOMEDIR)是否设置正确。

DES0720N 在数据库中没有注册存取函数 *schema.function*。

解释: 功能的名称是不正确的, 或者未向数据库注册功能。

校正方法: 检查存取函数的名称。如果它是正确的, 则检查数据库系统是否知道该功能。使用 CREATE FUNCTION 来向数据库注册存取函数。

DES0721N 数据库是不一致的; DB2 Text Extender 目录视图丢失

解释: 一种 DB2 Text Extender 目录视图不在数据库中

校正方法: 使用 DISABLE DATABASE 命令来除去剩余的目录视图, 然后再次输入 ENABLE DATABASE。索引数据丢失; 重新创建文本文档的索引。

DES0722N 表 *schema.table* 不是数据库中的基本表。

解释: 表在数据库中不存在, 或者它是结果表或视图。在可以为 DB2 Text Extender 启用文本列之前, 文本列必须在基本表中

校正方法: 确保表名是正确的, 并且它是一个基本表。

DES0723N 为表 *schema.table* 中的句柄列 *handlecolumnname* 创建索引失败。

解释: 不能为该句柄列创建文本索引。

校正方法: 使用 *txstatus* 来检查服务器的状态。如果服务器上的服务运行正确, 则使用 `DISABLE TEXT COLUMN` 或 `DISABLE TEXT TABLE` 来再次获得一致的状态。然后, 使用 `ENABLE TEXT COLUMN` 或 `ENABLE TEXT TABLE` 来再次启用文本列。

DES0724N 表 *schema.table* 中的句柄列 *handlecolumn* 的 **TextIndices** 目录视图图中的一项丢失。

解释: `TextIndices` 目录视图受到损坏。

校正方法: 使用 `DISABLE TEXT COLUMN` 或 `DISABLE TEXT TABLE` 来再次获得一致的状态。然后使用 `ENABLE TEXT COLUMN` 或 `ENABLE TEXT TABLE` 来启用文本列。

DES0727N 表 *schema.table* 中的列 *column* 已被启用。

解释: 如果已卸下表并使用同一文本列重建该表, 而没有首先禁用列, 则可能出现此信息。

校正方法: 禁用列, 然后再试。

DES0728N 列 *column* 在表 *schema.table* 中不存在。

解释: 正在尝试启用不存在的文本列。

校正方法: 更改表名或列名, 然后再试。

DES0729N 句柄列 *handlecolumn* 在表 *schema.table* 中不存在。

解释: 正在尝试使用不存在的句柄列。

校正方法: 使用 `GET STATUS` 命令来检查句柄列是否存在, 并且是否已正确指定了它的名称。

DES0730N 表 *schema.table* 已作为公共索引表而启用。

解释: 正在尝试启用已启用作公共索引表的表。

校正方法: 无需启用表而继续, 或运行 `DISABLE TEXT TABLE` 命令来在再次启用该表之前禁用它。

DES0731N 表 *schema.table* 未为 **Text Extender** 而启用; 不能禁用它。

解释: 正在尝试禁用未被启用的表。

校正方法: 检查表名。

DES0732N 在位置 *location* 附近的更新频率是不正确的; 所预期的更新频率是 *parameter*。

解释: “更新频率”的 *parameter* 规范是不正确的。

校正方法: 检查更新频率参数并重新输入命令。

DES0733N 表 *schema.table* 包含已启用的列; 不能启用它作为公共索引表。

解释: 此表包含已具有自己索引的文本列。在此个别索引存在的情况下, 不能为所有的文本列创建公共索引。

校正方法: 使用 `DISABLE TEXT COLUMN` 来禁用已启用的列, 然后再次输入 `ENABLE TEXT TABLE` 命令。

DB2 Text Extender 中的信息

DES0734N 句柄列 *handlecolumn* 属于部分文本表 *schema.table*; 不能单独禁用它。

解释: 不能禁用已在启用作为部分文本表的表中的单个文本列。

校正方法: 禁用完整的部分文本表。

DES0736N *handlecolumn* 已是表 *schema.table* 中的句柄列。

解释: 正在尝试使用现存的句柄列名。

校正方法: 使用句柄列的不同名称, 重新输入命令。

DES0737N 表 *schema.tablename* 启用作为公共索引表, 带有 **STORAGE** 选项 *storage_option*。

解释: 不可能启用外部文件的公共索引表。

校正方法: 如果想启用外部文件的表, 则使用多索引表。

DES0738N 存取函数 *schema.function* 具有不正确的参数。

解释: *schema.function* 的输入或输出参数是不正确的。

- 只能有一个输入参数, 并且它必须是要启用的文本列的数据类型。
 - 输入参数必须是类型 CHAR、GRAPHIC、VARGRAPHIC、LONG VARGRAPHIC、DBCLOB、VARCHAR、LONG VARCHAR 或 CLOB 的。
-

DES0739W 不能启动表 *schema.table*、句柄列名 *handlecolumn* 的索引更新程序。

解释: 不能启动更新索引的程序。安装期间可能出现了错误。

校正方法: 检查安装是否成功。检查环境变量 (例如, DB2TX_INSTOWNER 和 DB2TX_INSTOWNERHOMEDIR) 是否设置正确。

DES0740N 只有类型为 *columntype* 的文本列才能重新使用句柄列 '*column*'。

解释: 该句柄列已用于另一种句柄列类型。

校正方法: 指定新的句柄列名。

DES0741N 未找到或不能启动程序或文件 *parameter*。

解释: ENABLE DATABASE 或 DISABLE DATABASE 命令不能打开文件 *parameter*。安装期间可能出现了错误。

校正方法: 检查安装是否成功。

DES0742N 此表使用列索引。在命令中指定句柄列。

解释: 指定的表是作为多索引表启用的。要使用特定列, 指定相关的句柄列。

校正方法: 指定句柄列。

DES0745N DB2TX 实例所有者 *instance-owner* 不是有效用户 ID。

解释: 环境变量 DB2TX_INSTOWNER 不包含有效用户 ID。

校正方法: 校正环境变量。

DES0747N 对于索引类型 *index_type*, 当前 CCSID 不受支持。

解释: 对于所请求的索引类型, 指定了不受支持的 CCSID。

校正方法: 参见文档以了解受支持的 CCSID 的列表。

DES0748N DB2 Text Extender 不支持落实计数值 '*commitcount*'。

解释: 不支持指定的落实计数值。

校正方法: 指定有效的落实计数值。

DES0749N DB2 Text Extender 不知道更新索引值 'indexvalue'.

解释: 指定的 'update index' 值无效。

校正方法: 指定有效的更新索引值。

DES0750N DB2 Text Extender 不知道索引类型值 'indextype'.

解释: 指定的索引类型值无效。

校正方法: 指定有效的索引类型值。

DES0751N 您没有权限执行所指定的操作。

解释: 没有必需的数据库管理员权限来执行此项操作。

校正方法: 让数据库管理员执行此项操作。

DES0756N 未为 Text Extender 启用数据库。

解释: 在可以运行此命令前, 必须启用数据库。

校正方法: 运行 ENABLE DATABASE, 然后重新提交命令。

DES0763N 更新频率不正确。

解释: 指定的 'update frequency' 值无效。

校正方法: 指定有效的 'update frequency'。

DES0765N 已为 Text Extender 启用数据库。

解释: 正在尝试启用已启用的数据库。

校正方法: 无需启用数据库而继续进行, 或使用 DISABLE DATABASE 来在再次启用数据库之前禁用它。

DES0766N 某个操作导致超出了该表的最大行大小或临时表的最大行大小。

解释: ENABLE TEXT COLUMN 命令将句柄列添加至表。如果表已经很大, 这可能导致表的行大小超出最大值 4005。

ENABLE TEXT COLUMN 命令还创建临时表, 它的大小与已启用的文本列的数目成比例。如果已启用了许多文本列, 临时表的大小可能超出最大值。

校正方法: 只在不会导致超出此限制的表上使用 ENABLE TEXT COLUMN。

DES0769W 警告: 索引特性已指定, 但将被忽略。表 'tablename' 是公共索引表。

解释: 指定的表是公共索引表, 因此不可以指定索引特性。

校正方法: 不需要任何操作。

DES0770N 未定义环境变量 env-variable。

解释: 未指定命令参数, 且系统尝试从环境变量 env-variable 读取缺省值, 但是未定义此环境变量。

校正方法: 定义必需的环境变量。

DES0774N 变量 'variable' 的值长度 length 超出范围。

解释: 参数的值长度超出了范围。

校正方法: 指定具有有效长度的参数。

DES0775N 索引目录值 'directory' 不正确。

解释: 索引目录值不正确, 可能是目录长度不正确。

校正方法: 指定有效的索引目录值。

DES0776N 表空间名 'tablespace' 不正确。

解释: 指定的表空间名不正确, 可能是表空间值长度不正确。

校正方法: 指定有效的表空间值。

DES0777N 数据库管理系统不知道表空间 tablespace。

解释: 数据库系统不知道指定的表空间。

校正方法: 检查数据库中是否存在指定的表空间。

DB2 Text Extender 中的信息

DES0778N *'tablespace'* 不是“常规”表空间。它是用关键字 *'keyword'* 创建的。

解释： 不支持指定的表空间的数据类型。

校正方法： 指定常规表空间。

DES0779I 已成功启动索引。要检查索引状态，使用 **'GET INDEX STATUS'**。

解释： 索引程序已启动。可以使用 **'GET INDEX STATUS'** 命令来检查索引创建过程的状态。

校正方法： 检查 **GET INDEX STATUS** 命令的输出。

DES0780I 已成功启动索引重组。要检查索引状态，使用 **'GET INDEX STATUS'**。

解释： 重组程序已启动。可以使用 **'GET INDEX STATUS'** 命令来检查重组进程的状态。

校正方法： 检查 **GET INDEX STATUS** 命令的输出。

DES0789W **警告：** 当前节点组的分区映象已更新，请调用 **TXNCHECK** 实用程序。

解释： 当前节点组的分区映象已更新。

校正方法： 使用 **TXNCHECK** 命令。

DES0800I 成功地完成了 *command* 命令。

解释： 指定的命令成功完成。

校正方法： 不需要任何操作。

DES0810N 丢失右引号。

解释： 引号已找到，但是第二个引号丢失。

校正方法： 检查命令的语法，并再试。

DES0811N *"token"* 是出乎意料的。检查索引特性或文本信息。

解释： 索引特性或文本信息是不正确的。

校正方法： 检查语法并再试。

DES0812N 表 *schema.table* 不存在，或者未为 **DB2 Text Extender** 启用。

解释： 运行 **GET** 命令时，数据库表的名称不正确，或者该表不存在，或者尚未启用它。

校正方法： 如果表名是正确的，使用 **GET STATUS** 来检查它是否已被启用。启用该表并重试。

DES0813N 表 *schema.table* 不存在，或未为 **DB2 Text Extender** 启用，或不包含句柄列 *column*。

解释： 运行 **GET** 命令时，在表中未找到句柄列的任何项。如果表存在，则未启用它或它不包含句柄列。

校正方法： 如果表名是正确的，使用 **GET STATUS** 来检查它是否已被启用。启用该表并重试。

DES0814N 表 *tablename* 不存在，或者对 **DB2 Text Extender** 未启用，或者未启用此表中的任何文本列。

解释： 未对 **DB2 Text Extender** 启用指定的表。

校正方法： 对 **DB2 Text Extender** 启用该表。

DES0815N 找到空引号 *""*。在引号内期望有一个名称。

解释： 找到两个连续的引号，在它们之间没有文本。

校正方法： 检查语法并再试。

DES0816N 字 *"token"* 是意外的。使用关键字 *keyword* 或 *keyword* 的其中之一。

解释: 发现了意外的令牌。

校正方法: 在命令中, 使用在信息中给出的关键字之一。

DES0817N *"token"* 是出乎意料的。使用关键字 *keyword*。

解释: 发现了意外的令牌。

校正方法: 在命令中, 使用在信息中给出的关键字。

DES0818N 命令意外结束。预期关键字 *keyword*。

解释: 关键字丢失。

校正方法: 在命令中, 使用在信息中给出的关键字。

DES0819N 命令意外结束。下列其中一个关键字是预期的: *keyword* 或 *keyword*。

解释: 关键字丢失。

校正方法: 在命令中, 使用在信息中给出的关键字之一。

DES0820N 对于索引类型 *index_type*, 索引选项 *index_option* 不受支持。

解释: 对于给出的索引类型, 指定了不受支持的索引选项。

校正方法: 参见文档, 以了解对于给出的索引类型受支持的索引选项。

DES0821N 名称 *"token"* 太长。只有 *nn* 字符允许用于 *variable* 名称。

解释: 名称太长。

校正方法: 指定具有可接受的长度的名称。

DES0822N 命令包含不识别的令牌 *"token"*。期望命令结束。

解释: 找到了命令的结束, 但是预期有关键字。

校正方法: 检查命令的语法, 并再试。

DES0823N 预期在 *"schema."*。

解释: "." 后面丢失表名或函数名。

校正方法: 检查命令的语法, 并再试。

DES0824N 命令意外结束; *keyword* 是必需的。

解释: 信息中的关键字从语法中丢失。

校正方法: 检查命令的语法, 并再试。

DES0825N *'alias'* 是一个未知的数据库别名。仅允许 *nn* 个字符。

解释: 数据库系统不知道指定的数据库别名。

校正方法: 检查指定的别名是否有效。

DES0826N 数据库别名 *alias* 一定不能在引号中。

解释: 信息中的名称已被解释为数据库别名。它一定不能在引号中。

校正方法: 检查命令的语法, 并再试。

DES0827N *CCSID "ccsid"* 是不受支持的。

解释: *CCSID* 不是受 DB2 Text Extender 支持的那些之一。

校正方法: 参考文档以了解受支持的 *CCSID* 的列表。

DES0829N 用户名 *userid* 一定不能在引号中。

解释: 在引号中输入了用户名。

校正方法: 除去引号。

DB2 Text Extender 中的信息

DES0830N 不识别在 *enable/disable DATABASE* 命令中的参数 *"parameter"*。期望命令结束。

解释: 命令 ENABLE DATABASE 和 DISABLE DATABASE 不使用参数。

校正方法: 再次输入命令, 不带参数。

DES0831N 命令意外结束。丢失表名。

解释: 命令需要表名。

校正方法: 输入适当的表名。

DES0832N 命令意外结束。丢失数据库名。

解释: 命令需要数据库名。

校正方法: 输入适当的数据库名。

DES0833N 命令意外结束。丢失列名。

解释: 命令需要列名。

校正方法: 输入适当的列名。

DES0899N 未知的 DB2TX 命令: *command*。

解释: 指定的命令不受 DB2 Text Extender 支持

校正方法: 输入 **db2tx ?** 来获得命令的列表。

DES0971N 在不带节点说明的情况下, 只能指定一次索引目录, 若带有节点说明, 可指定多次。

解释: 索引目录的说明不正确。

校正方法: 检查索引目录的说明。可以指定一个不带节点说明的索引目录, 也可以指定多个带有节点说明的索引目录。

DES0972N 节点说明不正确。期望是无符号数值。

解释: 对节点号指定了非数值。

校正方法: 对节点号指定无符号数值。

DES0973N 节点说明不正确。在右括号前面找到了一个或多个意外的字符。

解释: 节点说明句法不正确。

校正方法: 检查节点说明的语法, 并再试。

DES0974N TO 子句中的节点号顺序不正确 (第二个节点小于第一个节点)。

解释: 节点说明句法不正确。

校正方法: 检查节点说明的语法, 并再试。

DES0975N 节点信息说明的语法不正确。

解释: 节点说明句法不正确。

校正方法: 检查节点说明的语法, 并再试。

DES0976N 节点信息不完整。

解释: 节点说明不完整 - 某些信息丢失。

校正方法: 检查节点说明的语法, 并再试。

DES0977N 节点信息不完整。左括号丢失。

解释: 节点说明不完整 - 左括号丢失。

校正方法: 校正节点说明并再试。

DES0998N 文档模型名的长度不正确。

解释: 'document model name' 值的长度不正确。

校正方法: 检查模型名值并再试。

DES0999N 文档模型说明的语法不正确。

解释: 模型名说明的语法不正确。

校正方法: 检查模型说明的语法并再试。

DES9994N 发生了“文本搜索引擎”错误。原因码: **reason-code**

解释: DB2 Text Extender 使用的文本搜索引擎发生了错误。

校正方法： 检查 第285页的『第19章 搜索引擎原因码』中的搜索引擎原因码。若报告的原因无法帮助解决问题，则 `desdiag.log` 文件或创建的跟踪文件中可能还有其他信息。如果这没什么帮助，则收集可获得的信息，并与 IBM 服务代表联系。

联系。

DES9995N 发生了 Text Extender 错误。 信息文本: `message-text`

解释： 发生了 DB2 Text Extender 错误。

校正方法： 使用 DB2 Text Extender 提供的信息来解决问题。若报告的信息无法帮助解决问题，则 `desdiag.log` 文件或创建的跟踪文件中可能还有其他信息。如果这没什么帮助，则收集可获得的信息，并与 IBM 服务代表联系。

DES9996N 发生了内部 DB2 Text Extender 错误 原因码: `reason_code`

解释： 发生了内部处理错误。

校正方法： 检查 DB2 Text Extender 安装是否已成功完成 如果是，注意原因码并与 IBM 服务代表联系。

DES9997N 发生了 SQL 错误。SqlState: `state`
QL 错误码: `rc`; SqlErrorMessage: `message`

解释： 发生了 SQL 错误。

校正方法： 对与此信息一起显示的 SQL 错误信息采取操作。

DES9998N 发生了 SQL 错误。没有进一步的信息可用。

DES9999N 没有对应的错误信息。

解释： 发生了内部处理错误。

校正方法： 检查诊断信息以解决该问题。若此内部错误不是由安装问题引起的，则 `desdiag.log` 文件或创建的跟踪文件中可能还有其他信息。如果这没什么帮助，则收集可获得的信息，并与 IBM 服务代表

第19章 搜索引擎原因码

本章列示由 DB2 Text Extender 的搜索引擎返回的原因码。

表 21. 搜索引擎原因码

原因码	值
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

搜索引擎原因码

表 21. 搜索引擎原因码 (续)

原因码	值
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. 搜索引擎原因码 (续)

原因码	值
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

第20章 错误事件原因码

本章列示在 DB2 Text Extender 对文档创建索引时可能发生的错误事件。这可能发生，例如，当：

- 不能对文档创建索引时
- 对文档进行了索引，但是出现了问题时
- 找不到语言词典时。

提示

如果原因码未归档：

1. 检查是否有足够的磁盘空间。
2. 收集所有可用的错误信息：
 - desdiag.log 文件
 - 事件信息
3. 与 IBM 服务代表联系。

1 存储器不够。服务器的内存用完。应减少工作负荷。

116

数据流语法错误

280

尚未对文档创建索引。不能打开其中一个索引文件。

281

尚未对文档创建索引。不能读取其中一个索引文件。

441

尚未对文档创建索引。只有 Ngram 索引才会发生此信息。文档的代码页与创建索引时所使用的代码页是不同的。若不是在 UTF8 中创建索引，HTML 和 XML 文档就可能会发生这种情况。

500

尚未对文档创建索引。不能装入“库服务”。检查 DLL 是否可用以及资源路径是否有效。

错误事件原因码

501

尚未对文档创建索引。 “库服务” 中的 Lib_Init 失败在 “平面文件” 系统上: 未找到 DIT 文件或它不在有效的目录上, 或 DIT 内容不正确。

502

尚未对文档创建索引。 在库服务 LIB_read_doc_content 中读取文档内容时出现了错误。

503

尚未对文档创建索引。 在库服务 LIB_access_doc 中出现了错误。

504

尚未对文档创建索引。 库服务 LIB_doc_index_status 返回了错误。

505

关闭文档失败。 库服务 LIB_close_doc 返回了错误。

506

结束 “库服务” 失败库服务 LIB_end 返回了错误。

507

库服务调用 LIB_read_doc_content 失败, 并带有意常的返回码。

508

库服务调用 LIB_close_doc 返回了 RC_TERMINATION 错误。

545

尚未对文档创建索引。 不能打开其中一个临时索引文件。

546

尚未对文档创建索引。 不能关闭其中一个临时索引文件。

548

内部错误。 将诊断日志中的信息发送给 IBM 代表。

549, 550

存储器不够 (alloc 失败)。 服务器的内存用完。 应减少工作负荷。

551-564

尚未对文档创建索引。 不能打开、读取、写入或关闭其中一个索引文件。 检查是否有足够的磁盘空间用于索引, 以及存取权是否正确。

565

存储器不够 (alloc 失败)。 服务器的内存用完。 应减少工作负荷。

566-587

尚未对文档创建索引。 不能打开、读取、写入或关闭其中一个索引文件。

588-590

内部错误。将诊断日志中的信息发送给 IBM 代表。

591-604

尚未对文档创建索引。不能打开、读取、写入或关闭其中一个索引文件。

605

存储器不够 (alloc 失败)。服务器的内存用完。应减少工作负荷。

606-623

尚未对文档创建索引。不能打开、读取、写入或关闭其中一个索引文件。检查是否有足够的磁盘空间用于索引，以及存取权是否正确。

624

存储器不够 (alloc 失败)。服务器的内存用完。应减少工作负荷。

625-631

尚未对文档创建索引。不能打开、读取、写入或关闭其中一个索引文件。检查是否有足够的磁盘空间用于索引，以及存取权是否正确。

632

用写存取权不能打开在索引期间创建的其中一个临时文件。检查存取权。

633

不能关闭在索引期间创建的其中一个临时文件。

634

不能写入索引期间创建的临时文件之一。检查索引工作目录是否具有足够的磁盘空间。

635

不能读取在索引期间创建的其中一个临时文件。

636

用读存取权不能打开在索引期间创建的其中一个临时文件。检查存取权。

659

不能打开索引期间创建的其中一个临时文件。

660

不能写入索引期间创建的其中一个临时文件。

661

不能关闭在索引期间创建的其中一个临时文件。

662

不能打开索引期间创建的其中一个临时文件。

错误事件原因码

663

不能写入索引期间创建的临时文件之一。

664

不能关闭在索引期间创建的其中一个临时文件。

665

不能打开索引期间创建的其中一个临时文件。

667

不能写入索引期间创建的其中一个临时文件。

668-669

未对文档创建索引。在文档中遇到的段标记与在文档模型文件中定义的那些段标记之间存在匹配问题。

670-672

尚未对文档创建索引。不能打开、读取、写入或关闭其中一个索引文件。检查是否有足够的磁盘空间用于索引，以及存取权是否正确。

673

存储器不够（alloc 失败）。服务器的内存用完。应减少工作负荷。

674

内部错误，将诊断日志中的信息发送给 IBM 代表。

675-687

尚未对文档创建索引。不能打开、读取、写入或关闭其中一个索引文件。检查是否有足够的磁盘空间用于索引，以及存取权是否正确。

688, 690

存储器不够（alloc 失败）。服务器的内存用完。应减少工作负荷。尝试配置文件中的更小值。

689

内部错误。将诊断日志中的信息发送给 IBM 代表。

691-695

尚未对文档创建索引。不能打开、读取、写入或关闭其中一个索引文件。检查是否有足够的磁盘空间用于索引，以及存取权是否正确。

696-707

内部错误，将诊断日志中的信息发送给 IBM 代表。

708

存储器不够（alloc 失败）。服务器的内存用完。应减少工作负荷。尝试配置文件中的更小值。

709-718

尚未对文档创建索引。不能打开、读取、写入或关闭其中一个索引文件。检查是否有足够的磁盘空间用于索引，以及存取权是否正确。

719-721

内部错误，将诊断日志中的信息发送给 IBM 代表。

722, 729

存储器不够 (alloc 失败)。服务器的内存用完。应减少工作负荷。尝试配置文件中的更小值。

730, 732, 733, 735-738

尚未对文档创建索引。引不能打开、读取、写入或关闭其中一个索引?? ?文件。检查是否有足够的磁盘空间用于索引，以及存取权是否正确。

731, 739-742, 744-746, 749, 755-758, 760-761, 767

内部错误，将诊断日志中的信息发送给 IBM 代表。

743, 748, 750-754, 759, 765-766, 768-770

尚未对文档创建索引。不能打开、读取、写入或关闭其中一个索引文件。检查是否有足够的磁盘空间用于索引，以及存取权是否正确。

747, 763, 764

存储器不够 (alloc 失败)。服务器的内存用完。应减少工作负荷。尝试配置文件中的更小值。

815

有两种可能的原因:

- 找不到需要支持用于导致故障的语言的其中一个资源文件。
- DB2 Text Extender 不支持该文档所请求的语言

831

尚未对文档创建索引。未找到任何文本。文档长度是 0 字节。

860

打开文件时发生错误。找不到某些词典或同义词典文件。检查词典文件的资源路径。若在搜索期间已经指定了同义词典文件的路径信息，则检查位置和文件名。

954-956

内部错误。将诊断日志中的信息发送给 IBM 代表。

957-967

尚未对文档创建索引。不能打开、读取、写入或关闭其中一个索引文件。检查是否有足够的磁盘空间用于索引，以及存取权是否正确。

错误事件原因码

1000

文件打开期间出现错误。检查存取权。

1001

文件追加期间出现错误。检查存取权。

1002

文件读取期间出现错误。文件可能是毁坏的。

1003

文件写入期间出现错误。检查磁盘空间和存取权。

1005

在文件读取期间发生错误（在文件内定位）。文件可能是毁坏的。

1006

在重命名临时文件期间发生错误。检查存取权。

1007

文件创建期间出现错误。检查存取权。

1008

在文件压缩期间发生错误。检查存取权。

1009

在关闭文件期间发生错误。文件可能是毁坏的。

1010

指定的索引名已在使用中。使用另一个索引名。

1011

指定的路径已在使用中。使用另一个位置。

1012

相同路径用于数据和工作目录。使用另一个位置。

1013

指定的索引名无效。索引名必须是大写字母或数字，并且不多于 8 个字符。

1014

在复制文件期间发生错误。检查存取权和磁盘空间。

1017

索引名是未知的。检查正确的拼写。

1019

在删除文件期间发生错误。检查存取权。此错误信息可以作为“次要错误”发生 - 查看诊断文件，以了解先前的错误项是否给出了更多信息。

1020

一般文件错误。检查存取权。

1070-1074

尚未对文档创建索引。指定的代码页无效，或者对正在存取的索引无效。

1085

尚未对文档创建索引。读取索引队列时出现错误。

1086

尚未对文档创建索引。索引队列是空的。

1116-1117

尚未对文档创建索引。不能处理服务器实例初始化文件中的信息。确保初始化文件中的各项是有效的，并且应用程序可以存取该文件。

1129

尚未对任何文档创建索引。启动后台处理失败。

1158

在重命名文件时发生错误。检查存取权和磁盘空间。

1162

Ngram 索引的索引文件可能被毁坏了。

1163, 1164

由于发生了意外的错误，因此尚未对文档进行索引。

1165

由于意外的文件结束条件，因此尚未对文档进行索引。

1176

对于 Ngram 索引，不能对更多文档进行索引。文档号发生了溢出（因为太长而溢出）。若对同一文档进行了多次删除或重复更新，则尝试调用 EhwReorg 以解决该问题。若不是以上情况，则对新文档使用第二个索引。

1177

尚未对文档创建索引。Ngram 位标认为它太大了。

1189

尚未对文档创建索引。边界序列有问题（特定于韩国语）。

1198-1200

尚未对文档创建索引。索引存取有问题。索引可能被毁坏了。

1201

尚未对文档创建索引。不能将文档代码页转换为特定于索引的代码页。只有 UTF8 代码页中的 Ngram 索引才会发生此错误。

错误事件原因码

1202

尚未对文档创建索引。由于文档中存在无效数据，所以不能将文档代码页转换为特定于索引的代码页。只有 UTF8 代码页中的 Ngram 索引才会发生此错误。

1500-1505

文档分析部件有问题。它可能未初始化（检查 LIBPATH 和配置文件的内容），或者是由于内部问题而失败。有关详情，参见诊断文件。

1904

尚未对文档创建索引。在存取启用了部分的索引的文档模型时有问题。检查存取权，以及文件是否存在。

2000

尚未对文档创建索引。文档类型不受支持。库服务 Lib_access_doc 返回无效的文档类型。

2001

尚未对文档创建索引。已在文档的数据流中检测到不正确的字段顺序。

2002

尚未对文档创建索引。已在文档的数据流中检测到不正确结构化的字段。

2003

尚未对文档创建索引。对 DB2 Text Extender 文本格式的文档只允许一个文本部分

2005

尚未对文档创建索引。在文档的数据流中指定的语言不受支持。

2006

尚未对文档创建索引。在文档的数据流中指定的 CCSID 不受支持。

2007

由库或缺省规则给出的预期文档格式不正确。对于该格式来说，文档标题不正确。检查缺省规则是否是带有特殊文档标题的文档，并且在规则不正确时更改它。

2008

因为不能存取文档，所以未对文档创建索引。

2009

因为文档在使用中并且不能存取，所以未对文档创建索引。

2010

尚未对文档创建索引。指定的 CCSID 不正确。

2011

因为文档不是有效的 IBM DCA RFT 或 FFT 文档，所以尚未对它创建索引。
页结束必须是文档的正文中的最后控制。

2012

因为文档不是有效的 IBM DCA RFT 或 FFT 文档，所以尚未对它创建索引。
结构化字段包含不正确的长度说明。

2013

因为文档不是有效的 IBM DCA RFT 或 FFT 文档，所以尚未对它创建索引。
已在文档中检测到不正确的控制。

2014

因为文档不是有效的 IBM DCA RFT 或 FFT 文档，所以尚未对它创建索引。
已在文档中检测到不正确的多字节控制或结构化字段。

2015

因为文档不是有效的 IBM DCA RFT 或 FFT 文档，所以尚未对它创建索引。
已找到复制的文档参数。

2016

因为文档不是有效的 IBM DCA RFT 或 FFT 文档，所以尚未对它创建索引。
已找到空的文本部件。

2018

文档具有不受支持的格式，或在文档扩展的 DIT 中有“排除”项。检查文档是否具有允许对它创建索引的扩展。

2020

尚未对文档创建索引。它既不是 WordPerfect 文档，也不是 WordPerfect 文件。

2021

尚未对文档创建索引。它是 WordPerfect 文件但不是 WordPerfect 文档。

2022

尚未对文档创建索引。此版本的 WordPerfect 不受支持。

2023

尚未对文档创建索引。它是加密的 WordPerfect 文件。在不加密的情况下存储文件。

2026

END_TXT 出现在脚注或尾注中。检查 WordPerfect 文件，它可能受到损坏。

错误事件原因码

2028

语法分析程序返回了非文档文本。检查文件内容，特别是那些特定于格式的单词。检查文档格式是否受支持。若自动格式识别失败，则应确保调用了正确的语法分析程序。

2030

尚未对文档创建索引。它不是 Microsoft Word 文件，或它是不受支持的 Word 版本。

2031

尚未对文档创建索引。已在 Microsoft Word 文档中检测到意外的文件结束。

2032

尚未对文档创建索引。已在 Microsoft Word 文档中检测到不正确的控制。

2033

尚未对文档创建索引。已用 *fastsave* 选项将它保存为复杂格式在关闭 *fastsave* 选项的情况下保存它

2034

尚未对文档创建索引。在 Microsoft Word 文档中，必需的字段结束标记丢失。

2035

文档是加密的。在不加密的情况下将文档存储在 Microsoft Word 中。

2036

这是 Word for Macintosh 文档；不能处理它。用 Word for Windows 格式存储文档。

2037

此 Word 文档包含嵌入的 OLE 对象。

2040

因为文档不是有效的 ECTF 文件，所以尚未对它创建索引。

2041

尚未对文档创建索引。它包含没有后接数字的 .SO LEN 控件。

2042

尚未对文档创建索引。它包含了后面跟有不正确数字的 .SO LEN 控件。该数必须在 1 和 79 之间。

2043

尚未对文档创建索引。只允许一个 .SO DOC 控件。将每一个 ECTF 文档保存在独立的文件中。

2044

尚未对文档创建索引。 .SO HDE 控件后面必须有开始和结束标记。

2046

尚未对文档创建索引。 文档包含 .SO DOC 控件之前的文本。

2047

尚未对文档创建索引。 文档包含 .SO PID 控件之前的文本。

2048

尚未对文档创建索引。 在开始标记后，结束标记丢失。

2050

尚未对文档创建索引。 已经在 .SO HDE 控件之后检测到不正确的标记。

2051

尚未对文档创建索引。 已经在 .SO 控件之后检测到行结束。

2052

尚未对文档创建索引。 已检测到意外的文本结束。

2060

尚未对文档创建索引。 它不是 AmiPro 文档，或它是不受支持的 AmiPro 版本。

2061

尚未对文档创建索引。 AmiPro 文档中的控制的长度太长。

2062

尚未对文档创建索引。 此版本的 AmiPro 不受支持。 只有 AmiPro Architecture 版本 4 受支持。

2063

尚未对“AmiPro 样式表”创建索引。

2064

尚未对文档创建索引。 已经检测到不正确的字符集。 只有“Lotus 字符集 82” (Windows ANSI) 受支持。

2065

尚未对文档创建索引。 已在 AmiPro 文档中检测到意外的文件结束。

2072

因为文档是加密的，所以不能扫描该文档。

2073

文档格式是不一致的。

错误事件原因码

2074

文档设置了“坏文件”标记位。

2080

尚未对文档创建索引。 它不是 RTF 文档，或它是不受支持的 RTF 的版本。

2081

尚未对文档创建索引。 已经检测到太长的 RTF 控制字。

2083

尚未对文档创建索引。 Macintosh 代码页是不受支持的。

2084

尚未对文档创建索引。 它是 RTF 文档，但是此 RTF 版本不受支持。 只有 RTF 版本 1 受支持。

2090

尚未对文档创建索引。 它是 HTML 文档，它包含一个语法分析程序认为太长的标记。

2093

尚未对文档创建索引。 它是 XML 文档，它被 XML 语法分析程序拒绝。

2100

文档受损坏，或因为某种其他原因不可读。新的公共语法分析器可校正问题。

2101

因为文档是空的或者不包含文本，所以不能对该文档创建索引。 检查文档是否只包含图形。

2102

因为文档是用口令保护的或者是加密的，所以不能对该文档创建索引。

2105

文档类型是已知的，但是过滤器不可用。

2106

因为文档是空的，所以不能对它创建索引。

2107

因为不能打开文档，所以不能对该文档创建索引。检查文档是否可以存取。

2112

因为文档是可执行文件，所以不能对该文档创建索引。

2113

因为文档是压缩的，所以不能对该文档创建索引。

2114

因为文档是图形的，所以不能对该文档创建索引。如果图形文档格式返回可接受的文本段，那么请求在索引过程中包括此文档格式。

2120

用户出口的输出文件不存在或是不可存取的。新的公共语法分析器版本可校正问题。

2121

不能打开输出文件进行读取或者该文件是空的。新的公共语法分析器版本可校正问题。

2122

试图使用用户出口输出文件，但是未在对象中给出或设置文件名。

2130

不能运行用户出口程序。检查在 `PATH` 环境变量中是否能够找到可执行程序。创建跟踪和转储来获取有关环境（错误）返回码的附加信息。

2131

用户出口程序失败，带有坏的返回码。创建跟踪和转储来获取有关环境（错误）返回码的附加信息。

第3部分 附录及附属资料

注意事项

IBM 可能未在所有国家中提供本文中讨论的产品、服务或功能部件。关于您所在区域目前可用的产品及服务的信息，请向当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并不说明或暗示只能使用 IBM 的产品、程序或服务。凡是同等功能的产品、程序或服务，只要不侵犯 IBM 的知识产权，都可以用来替代 IBM 产品、程序或服务。当然，评估和验证非 IBM 产品、程序或服务均由用户自行负责。

本文档的议题可能涉及 IBM 的某些专利或正在申请中的专利的应用。提供本文档，并不表示允许您使用这些专利。您可以将许可证查询以书面形式寄往：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

关于双字节 (DBCS) 许可证查询的信息，请与您所在国家的 IBM 知识产权部门联系，将查询以书面形式寄往：

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

以下段落不适用于英国与其它当地法律不允许这种供应方式的国家：国际商用机器公司『按原样』出版此书，不做任何明确或暗示的担保，包括但不限于不侵权、可销售或适用于特殊目的暗示性担保。一些地区在某些事务中不允许否认拒绝明确或暗示的担保，因此本条款可能不适合您。

本信息中可能有技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些信息将包含在本书新的版本中。IBM 可以随时对本书中说明的产品和/或程序进行改进和/或改动，而不必通知您。

此信息中对非 IBM Web 站点的任何引用仅是为了方便起见，而不以任何方式为那些 Web 站点作保证。那些 Web 站点的资料并非此 IBM 产品资料的一部分，使用那些 Web 站点的风险由您自己承担。

对于您所提供的任何信息，IBM 有权利以任何她认为适当的方式使用或散发，而不必对您负任何责任。

为了以下目的：(1) 允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换 (2) 允许对已经交换的信息进行相互使用，而希望获取本程序有关信息的合法用户请与下列地址联系：

IBM Canada Limited
Office of the Lab Director
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

只要遵守适当的条款和条件，包括某些情形下的一定数量的付款，都可获取这方面的信息。

这些信息中描述的特许程序及其所有可用的特许资料，按 IBM 客户协议、IBM 国际程序许可证协议或任何等价的协议中的条款，由 IBM 提供。

此处包含的所有性能数据都是在受控环境中确定的。因此，在其他操作环境中获得的结果可能与之相差很大。某些测量可能是在开发级的系统上进行的，不能保证这些测量方法在通用系统上同样可用。此外，某些测量方法可能是通过外推法归纳来估计的。实际结果可能会有所不同。此文档的用户应针对他们的特定环境验证数据是否适用。

涉及非 IBM 产品的信息可从这些产品的供应商、其发行公告或其它公众可用源得到。IBM 未测试这些产品，因此不能确认性能的精确度、兼容性或其它对非 IBM 产品的索赔赔偿要求等。有关非 IBM 产品功能方面的问题可向它们的供应商提出。

所有关于 IBM 未来方向或意向的声明都可能随时更改或撤消，而不作任何通知，并且仅代表发展目标。

此信息包含了用于日常商业处理的数据和报表的示例。为了尽可能完整地说明问题，这些示例中包含了个人、公司、品牌和产品的名称。所有这些名称都是虚构的，如与实际商业企业所使用的名称和地址相似，纯属巧合。

版权许可证：

本信息中可能包含用源语言编写的示例应用程序，它们说明了各种不同的操作平台上的程序设计技术。您可以为了开发、使用、市场营销或分发应用程序(这些应用程序遵守编写这些示例程序的操作平台的应用程序接口)的目的，以任何形式复

制、修改和分发这些示例程序，不用向 IBM 付费。这些例子未经所有条件下的完整测试。因此，IBM 不能保证或暗示其可靠性、可用性或这些程序的功能。

这些样本程序或任何派生产品的每个副本或任何部分必须包含如下的版权公告：

©（您的公司名称）（年度）。此代码各部分派生自“IBM 公司样本程序”。© Copyright IBM Corp. _输入年份_。All rights reserved.

注册商标

下列术语是 IBM 公司在美国和 / 或其他国家的商标：

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 Extender	SQL/400
DB2 OLAP 服务器	System/370
DB2 通用数据库	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

下列各项是其他公司的商标或注册商标：

Microsoft、Windows、Windows NT 和 Windows logo 是 Microsoft 公司在美国和 / 或其他国家的商标。

Java 和所有基于 Java 的商标都是 Sun Microsystems 公司在美国和 / 或其他国家的商标。

UNIX 是 Open Group 公司在美国和其他国家的注册商标。

其他公司、产品或服务名可能是其他公司的商标或服务标志。

词汇表

此词汇表定义许多在本手册中使用的术语和缩写。如果找不到要查找的术语，则参考索引或参考 *Dictionary of Computing* , New York: McGraw-Hill, 1994。

A

API. 应用程序设计接口。

B

边界搜索 (bound search). 在韩文文档中，考虑字边界的搜索。

布尔搜索 (Boolean search). 使用布尔运算符将一个或多个搜索条件组合在一起的搜索。

C

CCSID. 编码字符集标识符。

初始化句柄 (initialized handle). 事先准备好的句柄，只包含文本格式和 / 或文本语言。

出现 (occurrence). 是匹配的同义词。

触发器 (trigger). 无论何时从文本列中添加、更改或删除文档，自动将有关需要索引的文档的信息添加至日志表的机制。

词典 (dictionary). 在对特定语言的文档进行文本分析、创建索引、检索和突出显示期间，DB2 Text Extender 使用的与语言相关的语言信息集合

存取函数 (access function). 用户提供的函数，可将存储在列中的文本的数据类型转换为 DB2 Text Extender 可处理的类型

D

DBCS. 双字节字符支持。

DB2 Extender. 一组程序中的其中一个程序，除了传统的数字和字符数据外，它允许您存储和检索诸如图像、音频和视频数据以及复杂文档等数据类型。

代码页 (code page). 对所有代码点指定的图形字符和控制功能含义。例如，对 256 个 8 位代码的代码点指定字符和含义。

单值类型 (distinct type). 参见用户定义的单值类型 (*user-defined distinct type*)。

等级 (rank). 类型为双精度的绝对值，介于 0 和 1 之间，它指示与其他已找到的文档相比，文档满足搜索条件的程度。该值指示在文档中找到的、与文档的大小相关的匹配数。

定期创建索引 (periodic indexing). 按预定的时间间隔创建索引，以天、小时和分钟以及在创建索引前必须列示在 日志表中的最少文档名数表示。

多索引表 (multi-index table). 一个 DB2 表，它的文本列具有个别的文本索引。另见公共索引表 (*common-index table*)。

E

Extender. 参见 *DB2 Extender*。

F

分区数据库 (partitioned database). 由几部分组成的数据库，每一部分都由单独的数据库分区服务器维护。

G

格式 (format). 文档类型, 诸如 ASCII 或 WordPerfect。

跟踪 (tracing). 将信息存储在稍后可用来查找错误的原因的文件中的操作。

更新频率 (update frequency). 更新文本索引的频率, 以天、小时和分钟以及在创建索引前必须列示在日志表中的最少文档名数来表示。

公共索引表 (common-index table). 一个 DB2 表, 其文本列共享公共文本索引。另见 *多索引表 (multi-index table)*。

H

函数 (function). 参见 *存取函数 (access function)*。

环境变量 (environment variable). 用于为 DB2 Text Extender 环境的值提供缺省值的变量

环境简要表 (environment profile). 与 DB2 Text Extender 一起提供的脚本, 包含环境变量的设置

混合搜索 (hybrid search). 组合的布尔搜索和自由文本搜索。

J

计数 (count). 用于指定同义词典中词条的级别数 (深度) 的关键字, 这些级别用于扩展给定关系的搜索词条。

检索 (retrieve). 在 DB2 Text Extender 的其中一个搜索函数中使用搜索变元来查找文本文档。

简要表 (profile). 参见 *环境简要表 (environment profile)*。

节点 (node). 在分区数据库环境中的服务器。另见 *逻辑节点 (logical node)*、*物理节点 (physical node)* 和 *节点组 (nodegroup)*。

节点组 (nodegroup). 一个或多个数据库分区服务器的命名子集。节点分配给物理上分开的机器。另见 *逻辑节点 (logical node)*。

禁用 (disable). 通过除去启用过程中创建的项, 将数据库、文本表或文本列复原至为 DB2 Text Extender 启用它之前的状态

精确索引 (precise index). 一种文本索引, 包含与抽取词条的文本文档中出现的词条完全一样的词条。另见 *语言索引* 和 *Ngram 索引*。

句柄 (handle). 标识文本文档的二进制值。它包括:

文档 ID

相关索引的名称和位置

文档的文本信息

如果文档位于不在 DB2 控制下的外部文件中, 则是文件的路径和名称。

当启用了文本列供 DB2 Text Extender 使用时, 在该列中为每一个文本文档创建句柄

K

扩充 (expand). 将从同义词典中派生的附加词条添加至搜索词条的操作。

扩展匹配 (extended matching). 一个过程, 涉及使用 *词典* 来突出显示不是搜索词条的明显匹配的词条。

L

浏览 (browse). 查看在计算机监视器上显示的文本。

浏览器 (browser). 使您能够在计算机监视器上显示文本的 DB2 Text Extender 功能

逻辑节点 (logical node). 与其他节点一起分配给同一物理机器的节点。另见 *物理节点 (physical node)*。

M

命令行处理器 (command line processor). 即 DB2TX 程序, 它:

- 允许输入 DB2 Text Extender 命令
- 处理命令
- 显示结果。

模糊搜索 (fuzzy search). 一种搜索, 使用它可以找到其拼写与搜索词条的拼写相似的字。

目录视图 (catalog view). 由 DB2 Text Extender 创建的用于管理的系统表视图目录视图包含有关已允许 DB2 Text Extender 使用的表和列的信息

N

Ngram 索引. 支持 DBCS 文档和 SBCS 文档的模糊搜索的文本索引。另见语言索引和精确索引。

P

匹配 (match). 搜索词条在文本文档中的出现。

屏蔽字符 (masking character). 一个字符, 用来表示搜索词条的前部、中部和尾部的可选字符。屏蔽字符通常用于在精确索引中查找词条的各种变体。

Q

启用 (enable). 准备数据库、文本表或文本列以供 DB2 Text Extender 使用

R

日志表 (log table). 由 DB2 Text Extender 创建的一个表, 包含有关索引哪些文本文档的信息无论何时添加、更改或删除已启用的文本列中的文档, 都使用触发器将此信息存储在日志表中。

S

SBCS. 单字节字符支持。

shell 简要表 (shell profile). 参见环境简要表 (*environment profile*)。

实例 (instance). 一种逻辑 DB2 Text Extender 环境在同一个工作站上可以有几个 DB2 Text Extender 的实例, 但是对于每一个 DB2 实例来说, 只有一个实例可以使用这些实例来:

- 将开发环境与生产环境分开
- 限制敏感信息只对特定一组人员开放。

实例变量 (instance variable). 一个变量, 用来提供实例拥有者名称的缺省值, 或实例拥有者的主目录名称的缺省值。

数据流 (data stream). API 函数返回的信息, 由包含搜索词条的文本 (至少一个段落) 以及用于突出显示该文本中找到的词条的信息组成。

搜索变元 (search argument). 在创建索引时指定的条件, 由一个或几个搜索词条以及搜索参数组成。

索引 (index). 从文本中抽取重要的词条, 并将它们存储在文本索引中。

索引类型 (index type). 文本索引的一种特性, 它确定文本索引是否包含精确或语言格式的文档词条。参见精确索引、语言索引和 Ngram 索引。

索引特性 (index characteristics). 确定以下内容的文本索引特性:

- 存储索引的目录
- 索引类型
- 更新索引的频率
- 发生第一次索引更新的时间。

T

特性搜索 (feature search). 对诸如人名、地名或机构名这些词条的搜索，在使用 FEATURE_EXTRACTION 创建索引选项创建的语言索引中进行。

停止字 (stop word). 一种公共字，例如“before”，在文本文档中，将它从文本索引中排除，如果它包括在搜索变元中，则忽略它。

通配符 (wildcard character). 参见屏蔽字符 (*masking character*)。

突出显示信息 (highlighting information). 参见数据流 (*data stream*)。

U

UDF. 用户定义函数。

UDT. 用户定义的单值类型。

W

外部文件 (external file). 一个文本文档，以文件的格式存储在操作系统的文件系统中，而不使用 DB2 控制之下的表单元的格式。

文本表 (text table). 包含文本列的一个 DB2 表。

文本列 (text column). 包含文本文档的列。

文本配置 (text configuration). 创建索引、文本和处理这些值的缺省设置。

文本索引 (text index). 从文本文档中抽取的重要词条的集合。每一个词条都与从中抽取它的文档相关。通过在索引中进行搜索而不是在文档自身中进行搜索，可大大节省搜索时间。另见精确索引和语言索引。

文本文档 (text document). 类型为 CHAR、GRAPHIC、VARGRAPHIC、LONG

VARGRAPHIC、DBCLOB、VARCHAR、LONG VARCHAR 或 CLOB 的文本，存储在 DB2 表中。

文本信息 (text information). 描述下列内容的文本文档特性：

CCSID

格式

语言。

文档 (document). 参见文本文档 (*text document*)。

文档句柄 (document handle). 参见句柄 (*handle*)。

文档模型 (document model). 基于所包含的节而建立的文档结构定义。文档模型使 DB2 Text Extender 在创建索引时就知道文档内的节文档模型列示标识节的标记。对于每个标记，您可以指定描述性节名，以便在对该节的查询中使用。可以在文档模型文件中指定一个或多个文档模型。

文件句柄 (file handle). 参见句柄 (*handle*)。

物理节点 (physical node). 分配给物理上分开的机器的节点。另见逻辑节点 (*logical node*)。

X

细化 (refine). 将前一搜索中的搜索条件添加至其他的搜索条件，来减少匹配的数目。

Y

应用程序设计接口 (application programming interface, API). 应用程序和 DB2 Text Extender 信息检索服务之间的通用接口

用户定义的单值类型 (UDT) (user-defined distinct type (UDT)). 由 DB2 用户创建的数据类型，与由 DB2 提供的数据类型（如 LONG VARCHAR）形成对照。

用户定义函数 (UDF) (user-defined function (UDF)). 由 DB2 用户创建的 SQL 函数, 与由 DB2 提供的 SQL 函数形成对照。DB2 Text Extender 以 UDF 的形式提供管理和搜索函数, 例如 CONTAINS。

语言 (language). 在创建索引、搜索和浏览时使用的词典的名称。

语言索引 (linguistic index). 一种文本索引, 包含已通过语言处理缩小为基本格式的词条。例如, 为 “Mice” 建立的索引为 “mouse”。另见**精确索引**和 *Ngram* 索引。

Z

转义字符 (escape character). 一个字符, 指示不将后续的字符解释为屏蔽字符。

自由文本搜索 (free-text search). 一种搜索, 使用这种搜索时, 搜索词条被表示为自由格式文本 – 用自然语言描述要搜索的主题的短语或句子。

索引

[A]

安装验证 16

[B]

包含文件 des_ext.h 111

备份和复原 21

边界搜索, 示例 87

编译同义词典定义文件 174

编译 Ngram 同义词典定义文件 176

变量

 环境变量的说明 17

 显示环境变量 101

 GET ENVIRONMENT 命令 149

表空间 65

表空间和节点组 65

并行概念 7

布尔搜索变元 206

布尔运算符

 NOT 86

 & (AND) 和 | (OR) 81

[C]

查找变元

 边界搜索 87

 规则和限制摘要 210

 混合搜索 88

 模糊搜索 87, 207

 使用屏蔽字符 82

 属性值 204

 说明 201

 搜索词条变体 82

 搜索词条的各部分 82

 搜索发音类似的字 87

 搜索各种语言的词条 85

 搜索任何顺序的词条 83

 搜索若干词条 81

 搜索同义词 84

 同义词典搜索 87

 用 NOT 进行搜索 86

查找变元 (续)

 用 & 和 | 进行搜索 81

 语法 202

 在同一个段落中搜索词条 84

 在同一个句子中搜索词条 84

 在文档节中搜索词条 84

 指定 81

 自由文本搜索 88

 AND 关键字 206

 BOUND 关键字 207

 COUNT 关键字 205

 EXPAND 关键字 205

 FUZZY FORM OF 关键字 207

 IN SAME PARAGRAPH AS 206

 IN SAME SENTENCE AS 206

 MODEL 关键字 204

 NOT 关键字 206

 PRECISE FORM OF 关键字 207

 RESULT LIMIT 关键字 205

 SECTION 关键字 204

 STEMMED FORM OF 关键字
 207

 SYNONYM FORM OF 关键字
 207

 TERM OF 关键字 205

 THESAURUS 关键字 205

 & (AND) 运算符 206

 | (OR) 运算符 206

查找变元中的 & (AND) 运算符

 如何使用 81

 搜索变元语法 206

查找变元中的 | (OR) 运算符

 如何使用 81

 搜索变元语法 206

初始化句柄

 如何 91

 INIT_TEXT_HANDLE 函数 193

处理特性

 文本配置设置中的缺省值 19

触发器

 创建 65

触发器 (续)

 说明 30

 创建索引, 语言处理 235

 创建样本表

 TXSAMPLE 命令 170

 创建样本数据库 183

 创建 DB2 Text Extender 实例

 TXICRT 命令 162

 词典文件名 52

 词条的规范化 237

 词条扩展以便突出显示 244

 存取函数

 说明 68

 在 ENABLE TEXT COLUMN 中
 136

 错误事件

 记录 65

 删除 98

 显示 103

 原因码 289

 DELETE INDEX EVENTS 129

 GET INDEX STATUS 命令 152

[D]

打开文档, API 函数

 使用 115

 说明 229

大型表, 启用 67

单词分开 43

单值类型 185

等级

 从 DesGetSearchResultTable 114

 在搜索结果中 80

多分区节点组 7

多个索引, 使用 44

多节点环境 46

[F]

返回码 263

分区数据库支持 7

服务器

- 跟踪故障 22
- 启动 171
- 设置和维护 20
- 停止 173
- 显示状态 172
- TXICRT 命令 162
- TXIDROP 命令 164
- TXILIST 命令 165
- TXIPCLEAR 命令 166
- TXNADD 命令 167
- TXNCHECK 命令 168
- TXNDROP 命令 169
- TXSAMPLE 命令 170
- TXSTART 命令 171
- TXSTATUS 命令 172
- TXSTOP 命令 173
- TXTRACE 命令 178
- TXVERIFY 命令 183
- TXWIZARD 命令 184

复合词的字符分解 241

复合词, 划分 241

[G]

跟踪故障

- 设置 22
- TXTRACE 命令 178

更新频率

- 更改 96
- 说明 45
- 文本配置设置中的缺省值 19
- 语法 45
- 在 CHANGE INDEX SETTINGS 中的 UPDATEFREQ 124
- 在 CHANGE TEXT CONFIGURATION 中的 UPDATEFREQ 126
- GET INDEX SETTINGS 命令 150

更新文本索引

- 定期 45
- 更改频率 96
- 在 CHANGE INDEX SETTINGS 中的 UPDATEFREQ 124

更新文本索引 (续)

- 在 CHANGE TEXT CONFIGURATION 中的 UPDATEFREQ 126

更新状态, 重设

- 示例 97
- 语法 158

更新状态, 显示

- 示例和输出 102
- 语法 152

公共索引表

- 创建 60
- 说明 44
- ENABLE TEXT TABLE 命令 145

故障查找 22

关闭文档, API 函数

- 使用 116
- 说明 214

管理

- 备份和复原 21
- 编译同义词典定义文件 174
- 编译 Ngram 同义词典定义文件 176
- 重设索引状态 97
- 重组索引 98
- 创建样本数据库 183
- 创建 DB2 Text Extender 实例 20
- 跟踪故障 22
- 更改索引设置 96
- 更改文本配置 19
- 更新外部文件的索引 96
- 环境变量 17
- 禁用数据库 109
- 禁用文本表 109
- 禁用文本列 108
- 禁用文本文件 108
- 立即更新索引 95
- 命令行处理器 121
- 命令摘要, 服务器 161
- 命令摘要, 客户机 121
- 启动 DB2 Text Extender 服务器 171
- 删除索引事件 98
- 缩写文件, 编辑 53
- 停止字文件, 修改 53

管理 (续)

- 停止 DB2 Text Extender 服务器 173
- 维护文本索引 95
- 显示服务器状态 21
- 显示数据库、表和列的状态 100
- 显示索引设置 104
- 显示索引状态 102
- 显示文本信息设置 105
- 卸下 DB2 Text Extender 实例 20
- 修改停止字和缩写文件 53
- 样本数据库, 使用实用程序来创建 16
- 状态信息, 获取 100
- CHANGE INDEX SETTINGS 命令 123
- CHANGE TEXT CONFIGURATION 命令 125
- CONNECT 命令 128
- DB2TX 命令 121
- DELETE INDEX EVENTS 129
- DELETE INDEX EVENTS 命令 129
- DISABLE DATABASE 命令 130
- DISABLE TEXT COLUMN 命令 131
- DISABLE TEXT FILES 命令 132
- DISABLE TEXT TABLE 命令 133
- GET ENVIRONMENT 命令 149
- GET INDEX SETTINGS 命令 150
- GET INDEX STATUS 命令 152
- GET STATUS 命令 153
- GET TEXT CONFIGURATION 命令 154
- GET TEXT INFO 命令 155
- QUIT 命令 156
- REORGANIZE INDEX 命令 157
- RESET INDEX STATUS 命令 158
- TXICRT 命令 162
- TXIDROP 命令 164
- TXILIST 命令 165
- TXIPCLEAR 命令 166

管理 (续)

- TXNADD 命令 167
- TXNCHECK 命令 168
- TXNDROP 命令 169
- TXSAMPLE 命令 170
- TXSTART 命令 171
- TXSTATUS 命令 172
- TXSTOP 命令 173
- TXTHESC 命令 174
- TXTHESN 命令 176
- TXTRACE命令 178
- TXVERIFY 命令 183
- TXWIZARD 命令 184
- UPDATE INDEX 命令 159

国家语言支持 38

[H]

函数

- 参考 185
- 返回的 SQL 状态 271
- 概述 186
- 函数路径 78
- 说明 73
- 搜索函数 73
- 搜索文本 78
- 提高搜索性能 92
- 为 DB2 Text Extender 函数设置路径 78
- 细化先前搜索 89
- 用于转换数据类型 68
- 在句柄中设置并抽取信息 90
- 指定搜索变元 81
- API 函数 111
- CCSID 187
- CONTAINS 188
- FILE 189
- FORMAT 190
- HANDLE 191
- HANDLE_LIST 192
- INIT_TEXT_HANDLE 193
- LANGUAGE 194
- NO_OF_DOCUMENTS 195
- NUMBER_OF_MATCHES 196
- RANK 197
- REFINE 198
- SEARCH_RESULT 199

函数 (续)

- SET CURRENT FUNCTION
PATH 语句 78
- 环境变量 17
- 说明 17
- 显示 101
- GET ENVIRONMENT 命令 149
- 环境, 客户机 / 服务器 4
- 换码字符
- 使用 83
- 语法 210
- 混合搜索, 示例 88
- 获得浏览信息, API 函数
- 使用 115
- 说明 217
- 获得匹配, API 函数
- 使用 116
- 说明 220
- 获得搜索结果表, API 函数
- 使用 114
- 说明 225

[J]

- 基本格式, 将词条缩减至 240
- 基本文本分析
- 包含非字母数字字符的词条的 236
- 对于索引词条 236
- 规范化 237
- 句子识别 237
- 以便突出显示 244
- 检索, 语言处理用于 242
- 将词条缩减至其基本格式 240
- 节点
- 多节点环境 46
- 检查节点命令 TXNCHECK 168
- 节点组和表空间 65
- 添加节点命令 TXNADD 167
- 卸下节点命令 TXNDROP 169
- 指定 7
- 结果表 226
- 接口样本 27
- 结束浏览对话, API 函数
- 使用 117
- 说明 215

精确搜索 42

精确索引

- 说明 42
- 搜索选项缺省值 207

句柄

- 抽取 CCSID、格式和语言 91
- 初始化 91
- 从句柄列表中获得 191
- 单值类型 DB2TEXTFH 185
- 单值类型 DB2TEXTH 185
- 更改格式和语言 92
- 设置并抽取信息 90
- 使用列表来提高性能 92
- 说明 77
- 外部文件 78
- CCSID 函数 187
- FORMAT 函数 190
- LANGUAGE 函数 194
- 句柄列表指针 (单值类型 DB2TEXTFHLISTP) 186
- 句柄列表指针 (单值类型 DB2TEXTHLISTP) 185
- 句子分开 43
- 句子识别 237

[K]

- 客户机 / 服务器环境 4
- 扩展匹配 244

[L]

- 来自搜索引擎的原因码 285
- 连接至数据库
- 如何 58
- CONNECT 命令 128
- 列
- 对各种索引类型启用 66
- 禁用 108
- 启用 63
- 在大型表中启用 67
- DISABLE TEXT COLUMN 命令 131
- ENABLE TEXT COLUMN 命令 135
- 浏览
- 程序示例 233

浏览 (续)

- 使用自己的浏览器 114
- 语言处理用于 244

浏览函数 111

逻辑节点, 分配 7

[M]

命令

- 摘要, 服务器命令 161
- 摘要, 客户机命令 121
- CHANGE INDEX SETTINGS 123
- CHANGE TEXT CONFIGURATION 125
- CONNECT 128
- DB2TX 121
- DELETE INDEX EVENTS 129
- DISABLE DATABASE 130
- DISABLE TEXT COLUMN 131
- DISABLE TEXT FILES 132
- DISABLE TEXT TABLE 133
- ENABLE DATABASE 134
- ENABLE TEXT COLUMN 135
- ENABLE TEXT FILES 142
- ENABLE TEXT TABLE 145
- GET ENVIRONMENT 149
- GET INDEX SETTINGS 150
- GET INDEX STATUS 152
- GET STATUS 153
- GET TEXT CONFIGURATION 154
- GET TEXT INFO 155
- QUIT 156
- REORGANIZE INDEX 157
- RESET INDEX STATUS 158
- TXICRT 162
- TXIDROP 164
- TXILIST 165
- TXIPCLEAN 166
- TXNADD 167
- TXNCHECK 168
- TXNDROP 169
- TXSAMPLE 170
- TXSTART 171
- TXSTATUS 172
- TXSTOP 173

命令 (续)

- TXTHESC 174
- TXTHESN 176
- TXTRACE 178
- TXVERIFY 183
- TXWIZARD 184
- UPDATE INDEX 159

命令的帮助 58

命令行处理器

- 帮助 58
- 启动 56
- DB2TX 命令 121
- QUIT 命令 156

模糊搜索 41

模糊搜索, 示例 87

目录视图

- 创建 59
- 删除 109
- content 106

[P]

配置 16

配置表

- 创建 59
- 显示 101
- CHANGE TEXT CONFIGURATION 命令 125
- GET TEXT CONFIGURATION 命令 154

配置文件 257

匹配

- 从 DesGetSearchResultTable 114
- 在搜索结果中 80
- DesGetMatches 函数 220
- NUMBER_OF_MATCHES 函数 196

匹配, 扩展 244

屏蔽

- 语言处理 243
- 在搜索词条中 82

平面文件文档, 部分支持 50

平面 ASCII, 文档格式 32

[Q]

启动浏览对话, API 函数

使用 115

说明 231

启动 DB2 Text Extender 服务器 171

[R]

日期格式, 部分支持 48

日志表

抽取错误事件 103

创建 65

说明 30

指定给表空间 65

日志空间, 用完 67

入门 25

[S]

声音扩展

示例 87

说明 243

识别句子 237

时间格式, 部分支持 49

实例

创建 20

环境变量 17

卸下 20

用于列示的命令, TXILIST 165

事件原因码 289

首标文件 des_ext.h 111

受支持的浮点格式 49

数据库

备份和复原 21

禁用 109

连接至 58

启用 59

状态信息, 显示 100

CONNECT 命令 128

DISABLE DATABASE 命令 130

ENABLE DATABASE 命令 134

GET STATUS 命令 153

数据流语法 220

搜索变元规则 210

搜索变元限制 210

搜索词条的出现 196

- 搜索函数 111
- 搜索文本
 - 程序示例 233
 - 创建找到的文档的列表 192
 - 概述 78
 - 获取所找到的匹配数 80
 - 获取所找到的文档的等级 80
 - 使用 API 113
 - 提高性能 92
 - 细化先前搜索 89
 - 语法 202
 - 执行查询 79
 - REFINE 函数 198
 - SEARCH_RESULT 函数 199
- 搜索引擎原因码 285
- 搜索状态, 重设
 - 示例 97
 - 语法 158
- 搜索状态, 显示
 - 示例和输出 102
 - 语法 152
- 缩写
 - 编辑缩写文件 53
 - 列表 52
- 索引
 - 备份和复原 21
 - 重组 98
 - 大小计算 44
 - 定期索引更新 45
 - 多个, 使用 44
 - 概述 29
 - 更改当前的设置 96
 - 更改更新频率 96
 - 更改索引类型 44
 - 更改文本配置 19
 - 更新频率 45
 - 更新外部文件 96
 - 公共索引表 44
 - 计划 29
 - 精确 42
 - 类型 40
 - 立即索引更新 95
 - 维护 95
 - 为文本列创建各种类型 66
 - 文本配置设置中的缺省类型 19
 - 显示当前设置 104

- 索引 (续)
 - 语言 41
 - 在 CHANGE TEXT CONFIGURATION 中的 INDEXTYPE 126
 - 在 CHANGE TEXT CONFIGURATION 中的 TABLESPACE 126
 - 在 ENABLE TEXT COLUMN 中的 INDEXOPTION 138
 - 在 ENABLE TEXT COLUMN 中的 INDEXTYPE 138
 - 在 ENABLE TEXT TABLE 中的 INDEXOPTION 146
 - 在 ENABLE TEXT TABLE 中的 INDEXTYPE 146
 - CASE_ENABLED 选项 43
 - CHANGE INDEX SETTINGS 命令 123
 - CHANGE TEXT CONFIGURATION 命令 125
 - GET INDEX SETTINGS 命令 150
 - GET INDEX STATUS 命令 152
 - GET TEXT CONFIGURATION 命令 154
 - Ngram 43
 - UPDATE INDEX 命令 159
- 索引的空间需求 44
- 索引的目录
 - GET INDEX SETTINGS 命令 150
- 索引的状态
 - 重设 97
 - 出错之后重设 97
 - 显示 102
 - 显示当前状态 152
- 索引类型, 更改
 - 更改 44
 - 为文本列创建各种类型 66
- 索引事件
 - 原因码 289
- 索引事件, 删除
 - 示例 98
 - 语法 129

- 索引特性
 - 文本配置设置中的缺省值 18
 - 显示 104
 - 在 ENABLE TEXT COLUMN 中 135
 - 在 ENABLE TEXT FILES 中 142
 - 在 ENABLE TEXT TABLE 中 145
- 索引状态, 重设
 - 示例 97
 - 语法 158
- 索引状态, 显示
 - 示例和输出 102
 - 语法 152
- [T]
- 停止字
 - 编辑停止字文件 53
 - 列表 52
 - 说明 29
 - 作为基本文本分析的一部分 241
- 停止 DB2 Text Extender 服务器 173
- 通配符
 - 在搜索词条中 82
 - 字屏蔽 243
- 同义词
 - 说明 242
 - 在搜索变元中 84
 - SYNONYM FORM OF 关键字 207
- 同义词典搜索
 - 编译同义词典定义文件 174
 - 编译 Ngram 同义词典定义文件 176
 - 创建同义词典 250
 - 创建 Ngram 同义词典 254
 - 概念 246
 - 示例 87
 - 语法 205
 - THESAURUS 关键字 205
 - TXTHESC 命令 174
 - TXTHESN 命令 176
- 同义词典中词条的深度, 指定 205

突出显示信息

- 数据流 116
- 数据流语法 220

[W]

外部文件

- 初始化句柄 193
- 从句柄中抽取路径 / 名称 91
- 获得或更改句柄中的文件名 189
- 禁用 108
- 句柄 78
- 启用 69
- 索引更新考虑事项 96
- 在句柄中更改路径 / 名称 92
- DISABLE TEXT FILES 命令 132
- ENABLE TEXT FILES 命令 142
- FILE 函数 189

文本表

- 备份和复原 21
- 启用大型表中的列 67
- DISABLE TEXT TABLE 命令 133
- ENABLE TEXT TABLE 命令 145

文本分析

- 对于索引 236
- 用于浏览 244

文本配置设置

- 安装缺省值 18
- 更改 19
- 显示 101

文本索引的类型

- 精确 42
- 搜索选项缺省值 207
- 文本配置设置中的缺省值 19
- 语言 41
- 在 CHANGE TEXT CONFIGURATION 中的 INDEXTYPE 126
- 在 ENABLE TEXT COLUMN 中的 INDEXTYPE 138
- 在 ENABLE TEXT TABLE 中的 INDEXTYPE 146
- CASE_ENABLED 选项 43

文本索引的类型 (续)

- CHANGE TEXT CONFIGURATION 命令 126
- GET INDEX SETTINGS 命令 150
- Ngram 43

文本特性

- 格式 32
- 文本配置中的缺省值 18
- 语言 35
- 在 ENABLE TEXT COLUMN 中 135
- 在 ENABLE TEXT FILES 中 142
- CCSID 35

文本文档的格式

- 从句柄中抽取 91
- 受支持的列表 32
- 说明 32
- 文本配置中的缺省值 18
- 在句柄中初始化 193
- 在句柄中更改 92
- 在 CHANGE TEXT CONFIGURATION 中 127
- 在 ENABLE TEXT COLUMN 中 137
- 转换不受支持的格式 33
- FORMAT 关键字 127, 137
- FORMAT 函数 190
- GET TEXT INFO 命令 155

文本文档的数据类型

- 启用包含 DATALINK 数据类型的列 68
- 用于转换的函数 68
- 支持的 136

文本文档的语言

- 从句柄中抽取 91
- 列表 35
- 说明 35
- 文本配置中的缺省值 18
- 在句柄中初始化 193
- 在句柄中更改 92
- 在搜索变元中 85
- GET TEXT INFO 命令 155
- LANGUAGE 函数 194

文档

- 格式, 说明 32
- 结构 47
- 受支持的格式 32
- 索引 29
- 显示设置 105
- 信息, 关于 105
- 语言 35
- 在 CHANGE TEXT CONFIGURATION 中的格式 127
- 在 ENABLE TEXT COLUMN 中的格式 137
- 支持的数据类型 136
- 转换格式 33
- 转换数据类型 68
- CCSID 35
- GET TEXT INFO 命令 155

文档的结构

- 启用部分支持 47
- 搜索示例 84
- 搜索语法中的属性值 204
- 搜索语法中的 MODEL 关键字 204
- 搜索语法中的 SECTION 关键字 204

文档模型

- 说明 47
- 搜索语法中的属性值 204
- 搜索语法中的 MODEL 关键字 204
- 搜索语法中的 SECTION 关键字 204
- 修改文档模型文件 47

文档模型文件, 内容

文档中的节

- 平面文件文档 50
- 启用部分支持 47
- 受支持的浮点格式 49
- 受支持的日期格式 48
- 受支持的时间格式 49
- 受支持的整数格式 49
- 属性部分 47
- 搜索示例 84
- 搜索语法中的属性值 204

文档中的节 (续)

搜索语法中的 MODEL 关键字
204

搜索语法中的 SECTION 关键字
204

文档模型文件, 内容 50

DESMODEL.INI 47

HTML 文档 50

XML 文档 51

物理节点, 分配 7

[X]

细化先前搜索 89

向导样本

说明 27

TXWIZARD 命令 184

卸下实例

如何 20

TXIDROP 命令 164

信息 271

性能, 提高 92

[Y]

样本表

删除 109

说明 74

TXSAMPLE 命令 170

样本数据库

创建 (TXVERIFY) 16

样本向导 27

样本 API 程序 233

样本 DB2 Text Extender 函数

运行 73

应用程序设计接口 (API)

参考 213

程序示例 233

打开文档进行浏览 115

返回码 263

概述 112

关闭文档 116

获取浏览信息 115

获取匹配 116

获取搜索结果表 114

获取指向突出显示信息的指针
220

应用程序设计接口 (API) (续)

结束浏览对话 117

浏览函数 111

启动浏览对话 115

释放浏览信息存储器 117

搜索函数 111

搜索文本 113

突出显示信息 220

信息 271

摘要 213

DesCloseDocument 函数 214

DesEndBrowseSession 函数 215

DesFreeBrowseInfo 函数 216

DesGetBrowseInfo 函数 217

DesGetMatches 函数 220

DesGetSearchResultTable 函数
225

DesOpenDocument 函数 229

DesStartBrowseSession 函数 231

用户出口, 文档格式转换 33

用于浏览信息的自由存储器, API 函
数

使用 117

说明 216

用于索引的磁盘空间 44

由 DB2 Text Extender 函数返回的

SQL 状态 271

有关文本文档的信息

格式 32

类型 32

显示当前设置 105

语言 35

CCSID 35

GET TEXT INFO 命令 155

语言参数, 列表 52

语言处理

创建索引时 235

词条扩展 244

对于检索 242

划分复合词 241

基本文本分析 236

将词条缩减至其基本格式 240

扩展匹配 244

屏蔽 243

声音扩展 243

说明 235

语言处理 (续)

停止字过滤 241

同义词 242

用于浏览 244

字符屏蔽 243

字屏蔽 243

语言索引

说明 41

搜索选项缺省值 207

[Z]

整数格式, 部分支持 49

字符屏蔽 243

自由文本搜索

示例 88

A

AmiPro, 文档格式 32

AND

布尔运算符 81

搜索变元中的关键字 206

ASCII, 文档格式 32

B

BOUND 关键字 207

C

CASE_ENABLED 关键字

在 ENABLE TEXT COLUMN 中
138

在 ENABLE TEXT TABLE 中
146

CCSID

避免代码页问题 38

从句柄中抽取 91

函数 187

列表 35

说明 35

文本配置中的缺省值 18

在句柄中初始化 193

在 CHANGE TEXT

CONFIGURATION 中 127

CCSID (续)
 在 ENABLE TEXT COLUMN 中
 137
 GET TEXT INFO 命令 155
 CHANGE INDEX SETTINGS 命令
 使用 96
 语法 123
 CHANGE TEXT CONFIGURATION
 命令
 使用 19
 语法 125
 COMMITCOUNT 配置参数
 保留日志空间 67
 说明 67
 文本配置设置中的缺省值 19
 在 CHANGE TEXT
 CONFIGURATION 中 127
 在 ENABLE TEXT COLUMN 中
 141
 在 ENABLE TEXT TABLE 中
 159
 CONNECT 命令
 使用 58
 语法 128
 CONTAINS 函数
 示例 79
 语法 188
 COUNT 关键字 205

D

DATALINK 数据类型 68
 DB2 Extender
 用法示例 3
 DB2 Text Extender 服务器的状态
 172
 DB2 Text Extender 概述 3
 DB2 UDB 中的 LOGPRIMARY、
 LOGSECOND 和 LOGFILSIZ 参数
 67
 DB2DBDFT, 环境变量 17
 DB2INSTANCE, 环境变量 17
 DB2TEXTFH 单值类型 185
 DB2TEXTFHLISTP 单值类型 186
 DB2TEXTX 单值类型 185
 DB2TEXTHLISTP 单值类型 185

DB2TX, 命令行处理器
 使用 56
 语法 121
 DB2TX.SAMPLE 表
 删除 109
 说明 74
 用于创建的实用程序 16
 DB2TX_ 环境变量
 说明 17
 显示 101
 DB2TX_INSTOWNER, 环境变量
 17
 DB2TX_INSTOWNERHOMEDIR, 环
 境变量 17
 DBCS 文档, 搜索 43
 DELETE INDEX EVENTS 命令
 示例 98
 语法 129
 DesCloseDocument 函数
 使用 116
 说明 214
 DESCL.INI 257
 DesEndBrowseSession 函数
 使用 117
 说明 215
 DesFreeBrowseInfo 函数
 使用 117
 说明 216
 DesGetBrowseInfo 函数
 使用 115
 说明 217
 DesGetMatches 函数
 使用 116
 说明 220
 DesGetSearchResultTable 函数
 使用 114
 说明 225
 DESMODEL.INI 47
 DesOpenDocument 函数
 使用 115
 说明 229
 DESRESTB, 用于创建结果表 226
 DESSAMP1, 样本程序 233
 DESSRV.INI 258
 DesStartBrowseSession 函数
 使用 115

DesStartBrowseSession 函数 (续)
 说明 231
 DES_BROWSE, DesGetSearchResultTable
 中的选项 226
 DES_EXTENDED, DesOpenDocument
 中的选项 229
 DES_EXT.H 首标文件 111
 DES_FAST, DesOpenDocument 中的
 选项 229
 DES_MATCH, DesGetSearchResultTable
 中的选项 226
 DES_NOBROWSE, DesGetSearchResultTable
 中的选项 226
 DES_RANK, DesGetSearchResultTable
 中的选项 226
 DES_RANKANDMATCH, DesGetSearchResultTable
 中的选项 226
 DES_TEXTHANDLEONLY, DesGetSearchResultTable
 中的选项 226
 DIRECTORY 关键字
 文本配置设置中的缺省值 18
 显示当前设置 104
 在 CHANGE TEXT
 CONFIGURATION 中 126
 在 ENABLE TEXT COLUMN 中
 140
 在 ENABLE TEXT TABLE 中
 147
 DISABLE DATABASE 命令
 使用 109
 语法 130
 DISABLE TEXT COLUMN 命令
 使用 108
 语法 131
 DISABLE TEXT FILES 命令
 使用 108
 语法 132
 DISABLE TEXT TABLE 命令
 使用 109
 语法 133
E
 ENABLE DATABASE 命令
 使用 59
 语法 134

ENABLE TEXT COLUMN 命令
使用 63
语法 135

ENABLE TEXT FILES 命令
使用 69
语法 142

ENABLE TEXT TABLE 命令
使用 60
语法 145

EXPAND 关键字 205
Extender
用法示例 3

F

FFT, 文档格式 32

FILE 函数
示例 91
语法 189

FORMAT 函数
示例 91
语法 190

FUNCTION 关键字
说明 68
在 ENABLE TEXT COLUMN 中
136

FUZZY FORM OF 关键字 207

G

GET ENVIRONMENT 命令
示例和输出 101
语法 149

GET INDEX SETTINGS 命令
示例和输出 104
语法 150

GET INDEX STATUS 命令
示例和输出 102
语法 152

GET STATUS 命令
示例和输出 100
语法 153

GET TEXT CONFIGURATION 命令
示例和输出 101
语法 154

GET TEXT INFO 命令
示例和输出 105

GET TEXT INFO 命令 (续)
语法 155
GUI 样本 27

H

HANDLE 函数
使用 92
语法 191

HANDLE_LIST 函数
使用 92
语法 192

HTML 结构文档 47
HTML 文档, 部分支持 50
HTML, 文档格式 32

I

IN SAME PARAGRAPH AS 关键字
206

IN SAME SENTENCE AS 关键字
206

INDEXOPTION 关键字
在 CHANGE TEXT
CONFIGURATION 中 126
在 ENABLE TEXT COLUMN 中
138

在 ENABLE TEXT TABLE 中
146

INDEXPROPERTY 关键字
在 ENABLE TEXT COLUMN 中
139
在 ENABLE TEXT TABLE 中
146

INDEXTYPE 关键字
在 CHANGE TEXT
CONFIGURATION 中 126
在 ENABLE TEXT COLUMN 中
138
在 ENABLE TEXT TABLE 中
146

INIT_TEXT_HANDLE 函数
示例 91
语法 193

L

LANGUAGE 关键字 127, 137

LANGUAGE 函数
示例 91
语法 194

M

Microsoft, 文档格式 32

N

Ngram 索引
说明 43
搜索选项缺省值 207
CASE_ENABLED 选项 43

NODE 关键字
在 ENABLE TEXT COLUMN 中
140
在 ENABLE TEXT TABLE 中
148
NORMALIZED 关键字
在 ENABLE TEXT COLUMN 中
139

NOT
布尔运算符 86
搜索变元中的关键字 206
NO_OF_DOCUMENTS 函数
语法 195
NUMBER_OF_MATCHES 函数, 语法
196

O

ON NODE 关键字
在 ENABLE TEXT COLUMN 中
140
在 ENABLE TEXT TABLE 中
148
OR 布尔运算符 81

P

PRECISE FORM OF 关键字 207

Q

QUIT 命令
使用 71

QUIT 命令 (续)

语法 156

R

RANK 函数

示例 80

语法 197

REFINE 函数

示例 89

语法 198

REORGANIZE INDEX 命令

示例 98

语法 157

RESET INDEX STATUS 命令

示例 97

语法 158

RESULT LIMIT 关键字 205

S

SEARCH_RESULT 函数

示例 92

语法 199

SET CURRENT FUNCTION PATH

语句 78

shell 简要表 17

STEMMED FORM OF 关键字 207

T

TERM OF 关键字 205

TEXTINDEXES 目录视图

创建 59

删除 109

content 106

TXICRT 命令

创建 DB2 Text Extender 实例 20

语法 162

TXIDROP 命令

语法 164

TXILIST 命令

语法 165

TXIPCLEAN 命令

语法 166

TXNADD 命令

语法 167

TXNCHECK 命令

语法 168

TXNDROP 命令

语法 169

TXSAMPLE 命令

使用 16

语法 170

TXSAMPLE.UDF

运行 73

TXSTART 命令

使用 21

语法 171

TXSTATUS 命令

使用 21

语法 172

TXSTOP 命令

使用 21

语法 173

TXTHESC 命令

语法 174

TXTHESN 命令

语法 176

TXTRACE 命令

使用 22

语法 178

TXVERIFY

创建样本数据库 16

TXVERIFY 命令

语法 183

TXWIZARD 命令

语法 184

U

UDT 185

UPDATE INDEX 命令

示例 95

语法 159

UPDATEFREQ 关键字

在 CHANGE INDEX SETTINGS
中 124

在 CHANGE TEXT
CONFIGURATION 中 126

在 ENABLE TEXT COLUMN 中
139

UPDATEFREQ 关键字 (续)

在 ENABLE TEXT TABLE 中
147

UPDATEINDEX 关键字

文本配置设置中的缺省值 19

显示当前设置 104

在 CHANGE TEXT

CONFIGURATION 中 126

在 ENABLE TEXT COLUMN 中
140

GET INDEX SETTINGS 命令

150

W

WordPerfect, 文档格式 32

X

XML 结构文档 47

XML 文档, 部分支持 51

XML, 文档格式 32

与 IBM 联系

如果有技术问题，请在与“DB2 客户支持中心”联系之前复查并执行 *Troubleshooting Guide* 所建议的操作。本指南对您可以收集哪些信息以使“DB2 客户支持中心”更好地为您服务提出了建议。

要获取信息或订购任何“DB2 通用数据库”产品，与当地分支机构的 IBM 代表联系，或与任何特许 IBM 软件经销商联系。

您如果住在美国，请致电下列其中一个号码：

- 1-800-237-5511，可获得客户支持
- 1-888-426-4343，可了解所提供的服务项目

产品信息

您如果住在美国，请致电下列其中一个号码：

- 1-800-IBM-CALL (1-800-426-2255) 或 1-800-3IBM-OS2 (1-800-342-6672)，可订购产品或获取一般信息。
- 1-800-879-2755，可订购出版物。

<http://www.ibm.com/software/data/>

DB2 万维网网页提供关于新闻、产品说明、培训计划等等的当前 DB2 信息。

<http://www.ibm.com/software/data/db2/library/>

“DB2 产品和服务技术库”可供您访问常见问题、修订、书籍以及最新的 DB2 技术资料。

注：此资料可能只有英文版。

<http://www.elink.ibm.com/pbl/pbl/>

International Publications Ordering Web 站点提供关于如何订购书籍的信息。

<http://www.ibm.com/education/certify/>

IBM Web 站点中的“专业认证程序”提供各种 IBM 产品（包括 DB2）的认证测试信息。

<ftp://software.ibm.com>

以匿名形式注册。可在目录 /ps/products/db2 中找到有关 DB2 和许多其他产品的演示程序、修订、信息和工具。

comp.databases.ibm-db2, bit.listserv.db2-l

这些 Internet 新闻组可供用户来讨论使用 DB2 产品的经验。

On Compuserve: GO IBMDB2

输入此命令来访问 IBM DB2 系列论坛。这些论坛支持所有的 DB2 产品。

有关如何在美国以外的地区与 IBM 联系的信息，参见 *IBM Software Support Handbook* 的附录 A。要存取此文档，访问以下 Web 页面：<http://www.ibm.com/support/>，然后选择该页面底部附近的 IBM Software Support Handbook 链接。

注：在某些国家，IBM 特许经销商应与他们的经销商支持机构联系，而不是与“IBM 支持中心”联系。

读者意见表

DB2® 通用数据库
DB2 Text Extender
管理和程序设计

SB84-0248-00

姓名

地址

单位及部门

电话号码

读者意见表
SB84-0248-00



请沿此线
撕下或折起

折起并封口

请勿使用钉书机

折起并封口

在此
贴上
邮票

IBM Deutschland Entwicklung GmbH
Information Development, Dept 0446
Schoenaicher Strasse 220
71032 Boeblingen
Germany

折起并封口

请勿使用钉书机

折起并封口

SB84-0248-00

请沿此线
撕下或折起



Printed in U.S.A.

SB84-0248-00



Spine information:



DB2[®] 通用数据库

DB2 Text Extender 管理和程序设计

版本 7